UCL : A new Method for Cross–Layer Network Modelling

M. Issoufou Tiado, R. Dhaou and A.-L. Beylot

ENSEEIHT – IRIT Lab.

2 rue C. Camichel, BP7122 – 31072 Toulouse Cedex - France

***Abstract*— The evolution from wired network based systems to wireless environments enables the emerging of cross–layer systems to improve the performance of systems such as Ad-hoc networks. Several particular protocols have demonstrated some reachable performance gain by using cross–layer mechanisms. Global integration on each level of the protocol stack has to be ensured. Conversion of wired systems to wireless environments has to be considered through efficient methods that may either produce or update cross–layer conceptual models. Those models allow an efficient organisation of the translation from the wired to the wireless systems and highlight the work of translation to be realised. These models could take several forms. In our approach, a cross–layer conceptual model is composed of :** cross–layer interaction models **and** interactions description arrays**, all of them produced by an Upward Cross-Layer (UCL) method. In this paper, we propose a method with several steps which has been applied to a chosen protocol stack. A set of atomic actions and cross–layer interactions arrays are produced. Interactions based models and interactions description arrays are generated.**

***Index Terms*— Ad hoc Network, Cross–layer Method.**

A. Introduction

Ad hoc Networks are wireless networks characterised by a dynamic topology, a limited bandwidth, and energy consumption constraints. The wireless link quality changes through time and space, with small–scale memory because of multipath that may cause bursts of errors during which it is not possible to transmit packets correctly. Moreover, it has a large–scale variation: the average state of the channel depends on the user position and on possible interferences [1]. Wireless networks are generally less efficient than wired ones. Thus, classical protocols can not be used directly in a wireless environment. Some innovating techniques have to be developed to improve the performance of those networks.

If each protocol of the layered model is designed independently, the end of the execution of a low level protocol, consuming data at the destination node, should not influence the behaviour of high level protocols. In an opposite operating mode, the cross–layer concept adapts the protocols to the wireless context by sharing information between layers and by an overall optimisation instead of multiple optimisations at different levels. Several significant experiments were already performed [2][5][6][19]. The Cross–layer technique can be used by all the protocols at several levels if there are interactions whose execution improves the performances of the global system. Specific protocols such as those improving the TCP throughput have been proposed in [2]. Other large models implementing cross–layer interactions have been designed such as MobileMan [3] or CLASS [4]. Nevertheless, because of the plurality of protocols, the diversity of their behaviour (even at the same layer), and the possible interactions between themselves, it is important to design a method adapted to any interaction, ensuring a continuous evolution of cross–layer models and allowing the integration of new protocols and interactions. An interaction may be defined as an information exchange between protocols of different layers, not necessarily adjacent, that may be located in one or several nodes. Their architecture may be complex and may lead to a partial model design or produce apparent antagonist models when taken separately. A conceptual method allows to integrate in the same model different aspects of the cross–layer Interaction Model (CLIM). For example, in [3], the MobileMan system based on “full cross–layer design” has been proposed in opposite to “layer triggering signals”. We will show that triggering signals such as Explicit Congestion Notification or L2 triggers are a kind of cross–layer interactions gathered in cross–layer Atomic Action of Notification. In fact, the two concepts are different aspects of a global cross–layer model. A part of this model consists of cross–layer information collection and their exposition to other layers, the other part consists of messages or signals exchanged between layers when particular events occur.

In [4], different methods implementing cross–layer such as « Packet Header » or « ICMP Messages » are presented. They are complementary if taken for particular interactions.

The designed method has the advantage to highlight the impact of each cross–layer interaction on each protocol in order to update its source code and adapt it to this context. These modifications will not affect the behaviour of the protocol if the interaction is disabled (upward compatibility principle). The method may be applied to a given protocol stack or to an existent cross–layer model to integrate other interactions.

By considering cross–layer Interactions Models (CLIM) as a conception and protocols and interactions as an implementation, we propose an upward method that shows the evolution from concrete models to conceptual models. This method aims at an efficient organisation and uses potentialities that may improve the performance of the designed system.

B. Cross–Layer design Method

1st. Cross–Layer Atomic Action (CLAA) Concept

A Cross–Layer Atomic Action (CLAA) may be the setting or the utilisation of a layer parameter or service that concerns other layers, a behaviour or layer arriving events that have to be exposed to other layers. The term "Atomic" means that the action can not be divided into actions that do not impact the same protocols. An action such as "the coordination of the point-to-point link layer communication with the end-to-end transport layer communication" [4] or "the utilisation of channel state" are not atomic. The first one is imprecise and the second one refers to the use of parameters such as BER, SNR, carrier power, existence of carrier signal, retransmission/acknowledgement management policy, …

Three kinds of CLAA may be distinguished :

- **Exported States CLAA (ES-CLAA)** correspond to CLAA that export layer parameters to other layers. They may be used for admission control or QoS. MobileMan system, distributed WCI servers [4] are designed as Exported States models.
- **Notified Events CLAA (NE-CLAA)** : those CLAA report events to other layers. Examples of those interactions are error control coordination, delay jitter notification when transmitting a packet during a temporally "bad" channel state (avoiding sending new data), retransmission avoidance notification. CLASS system is consequently a model of "Notified events".
- **Activable Services CLAA (AS-CLAA)** : when specific layer mechanisms are developed to give interesting parameters or specific services to other layers.

2nd. Modelling of Interactions

"Exported States" and "Activable Services" CLAA are local interactions within a node. They can be characterised by variables/environment parameters having specific significant values. For example, the activation of a service such as VMAC is described through a simple Boolean variable. Using this service, environment variables, such as estimation of local delay, jitter and collisions, will then be regularly updated.

"Notified Events" CLAA includes both local interactions (significant energy drop notification) [5] and distant interactions (Explicit Congestion Notification) [6].

Thus the cross–layer interaction model is subdivided into:

- an environment subsystem which includes environment variables and parameters;
- an interface subsystem which allows communication between non adjacent layers;
- a distant subsystem which allows communication between layers of separate nodes.

To fit the necessary standardisation of the communication mechanisms of the global model, we suggest for the environment subsystem input/output functions. For the interface subsystem, a choice will be made between input/output functions and a standard protocol when performance evaluation results through simulation of cross–layer models will be produced. Information that are conveyed by each interaction will allow to define a protocol or input/output functions. For the distant subsystem, standardised protocols will be used according to each CLAA.

3rd. Method steps

We propose the following seven steps method :

1. Select a layered protocol stack to produce the cross–layer Interaction Model;
2. Cross–layer Atomic Action census : it could be either a set or a simple CLAA for which a performance evaluation has to be performed;
3. Production of Protocols interaction array : this array represents the interactions between CLAA (array lines) and protocols (array column). Each array cell can take those values : S (local or distant) if the protocol is source of the interaction, D if the protocol is the destination of the CLAA, U if the protocol uses the CLAA data, X if the protocol exchanges signals for the setting or exploitation of the CLAA;
4. Production of the Protocol functions interaction array : protocols are divided into functions. For each protocol, the previous array is modified. The column representing the protocol is divided into columns, each one corresponding to a protocol function. This array has the advantage to show the functions to be modified for this CLAA implementation and for each protocol;
5. Deduction of an interaction model for each kind of CLAA: the array produced in step 3 shows the cross–layer interaction model for each kind of CLAA. This model shows the layered protocol stack chosen in step 1 with an additional subsystem and interactions arrows. Interaction model helps understanding the internal cross–layer mechanisms of the global model;
6. Production of an interaction description array for each protocol. For a given CLAA and a given protocol, it indicates the origin of the CLAA, the source or the destination function, the kind of communication to use (direct, via subsystem, upward/downward/normal) and the possible exploitation of the CLAA by the influenced protocol's function;
7. Deduction of the implementation mode of each interaction model : every CLAA belongs to an upper predefined subsystem. Each subsystem has a standardised communication method.

Note than, even if some information is often repeated in different format, from one step to another, the method uses that redundancy to clarify the design.

C. APPLICATION OF THE METHOD

1st. Protocols stack choice

To experiment our method for the design of cross–layer models, we choose some specific protocols : TCP, DSR (Dynamic Source Routing) [7-10], IP, IEEE 802.11 (link and physical layer) [11]. Each protocol contribute to the definition of the functions of the protocol in the layered protocol stack. These functions are influenced by the listed CLAA.

TCP allows a reliable transfer over a connection between two nodes. It ensures data transfer control function (determine if the segment is damaged, lost or duplicated, reorder in case of an out of order delivery), error correction function (by retransmission), flow control function, congestion control function, priority management function.

IP protocol ensures data transfers through routing function (verify IP header, time to live, node identification, routing algorithm structure), and fragmentation function.

DSR protocol ensures functions such as routing (by using route message, IP data, routing structure), route discovery (route requests/reply messages), transmission control (route maintenance – using of acknowledgement), route error management (route maintenance – route error messages), packet salvaging (route maintenance – modifying source route, original sender notification), fragmentation (to adjust packets to the path size).

IEEE 802.11 protocol ensures control functions at the LLC sub-layer (802.2 protocol), security and integrity functions at the MAC under layer (802.10 protocol), wireless medium access control function (802.11) at the MAC sub-layer.

2nd. CLAA census

1) Activable services : AS-CLAA

The following services can be considered as AS-CLAA:

- IntServ [4, 12] is a network layer service. Its architecture defines a set of extensions to provide QoS.
- RSVP [12] is another network layer service. It is a signalling protocol used by applications (such as multimedia applications) to ask for network resources.
- DiffServ [4, 12] is also another network layer service. It classifies packets in a small number of aggregated or class flows : behaviour aggregate (BA) classification.
- FEC [4, 13] is introduced at link layer. It is used for Bit Error Rate (BER) control and overcomes the packet loss and bits corruption.
- ARQ [4, 13-15] is introduced at link layer. ARQ mechanism checks the integrity of each frame (example of strong link layer CRC) to detect channel errors and uses retransmission process on reverse channel to send back lost frames (data blocks).
- VMAC (Virtual MAC) [11, 16] is introduced at link layer. It monitors the radio channel to establish delay, jitter, collisions and packets loss estimations using DIFS free time measure, virtual packets, simulation of transmissions and virtual packets stamps. When using VMAC, a virtual source (VS) adjusts its application parameters and determines the accepted service level.

2) Exported states CLAA (ES-CLAA)

Let us examine some exported states CLAA. The Energy level [5, 17] is an ES-CLAA of the system energy manager, it implements the interaction that updates environment subsystem variable indicating the battery level so that protocols adapt their behaviour. Gallager pioneer works [17] define a reliable communication through energy constraints. Nodes have a finite energy and thus a finite number of bits before energy exhausts. Bit allocations according to network needs become an interesting optimisation problem that requires co-operation between all the layers.

The Packet loss ratio is also an ES-CLAA [4] produced by the Link layer control function. It corresponds to the calculation of environment subsystem variables :

- Non acknowledged packet number / total number of sent packets during a period;
- Damaged received packets number / total number of received packets during a period.

Each parameter threshold defines the state of the channel.

Physical layer SNR (Signal to Noise Ratio) [4] is also an ES-CLAA which represents the update of environment subsystem variable by the physical layer. This parameter gives signal to noise ratio value valid during a period of time and defines the good or bad state of the channel.

Physical layer RSS (Received Signal Strength) ES-CLAA [4, 18] materialises the update of environment subsystem variable that gives a signal intensity received from a node. Its value by threshold allows to evaluate the distance between two nodes or to establish their direct access, for the needs of routing protocols. [20] illustrates the importance of this parameter in simultaneous broadcasting context on a wireless link: the author refers to a catch phenomenon model based on received power subdivision by the base station, in p levels indexed from 1 to p. It is assumed that the reception is always successful for simultaneous broadcasting in two or more power levels and the reception is never successful in the same level because of collision.

The Physical layer BER [4] is an ES-CLAA which corresponds to the calculation of the environment subsystem variables that allows to define the “good” or “bad” state of the channel : Number of received damaged bits / total number of received bits in a period of time.

3) Notified events CLAA – NE-CLAA

The jitter of sent packets [4] is a NE-CLAA established at link layer because of retransmissions or because of the persistence of parameters indicating the “bad” state of the channel (packet loss ratio, SNR, BER) and can be used by TCP and application layer.

Retransmission avoidance [4, 19] for saturation reasons, IP layer handoff [18] or other reasons that need retransmission and new traffic admission freezing is a NE-CLAA which implies direct interaction from the link layer. It may be sent to TCP or to the application.

Acknowledgement is a NE-CLAA. The link layer uses one or more frames to transmit IP datagram over physical link. DSR uses those acknowledgements based on link layer grouping of acknowledged frames containing complete IP datagram [7-8] by using SIFS intervals of 802.11 [11]. The use of those acknowledgements is extended to transport layer.

ECN [1,6] and ELN [4,19] are NE-CLAA. When routers detect a congestion before the overflow of their internal buffers, they set the ECN bit in the TCP header of the packets. The receiver node reports the congestion to the sender by turning “on” the ECN bit in TCP header. When the TCP sender receives that indication, it invokes the congestion avoidance mechanism. In the case of wireless networks with infrastructure, a module called snoop agent can be introduced at the base station. It records all packets that pass through TCP connections and keeps the trace of wholes (non acknowledged segments by the receptor that are lost on the wireless link). The snoop agent sets the ELN bit in the duplicate ACK if it corresponds to a segment of the whole list before conveying it to the sender. When receiving such an ACK, the sender retransmits the next segment and do not take any congestion control action. The use of snoop agent on a mobile node is not appropriate because there is no possibility for the sender to know if the loss occurs on the wireless link or elsewhere in the network because of congestion. That is why in our ad hoc network, this CLAA will not be used.

Another example is the Significant energy lowering event [5]. When the energy level reaches a crucial threshold, the system energy manager sends this information to all the layers. Packet salvaging [7-8] is also a NE-CLAA. DSR is designed to detect that a link of a path is out of use. Thus, when the intermediate node has another route for the IP destination address of the conveying packet, it salvages the packet using the new route. This salvage may also be done according to the inaccessibility of the next node indicated by state channel parameters as BER, RSS.

The “Received node signal power” [4, 18] is a NE-CLAA that represents the sending of packet RSS value to DSR by 802.11 link layer. It can be used in routing table (establishing direct access or not to a facing node).

Finally, “Sending jitter due to route error” and “Sending jitter due to route modification” [7-8] are NE-CLAA. They allow to continue TCP adaptation in an ad hoc context. The first one materialises the interaction by which DSR reports to TCP that a route error occurs during a packet conveying to avoid the congestion control mechanism of TCP. The second one allows to inform TCP about a route modification that occurs during a packet conveying and may extend reception delays, to avoid the congestion control mechanism of TCP.

3rd. Protocols Interactions Array

The CLAA which have been described will now be classified in a protocol interaction array which includes the protocols that use those CLAA, the source and the destination of the interactions.

Cross – Layer Atomic Actions (CLAA)	Protocols					
	Application	**TCP**	**DSR**	**IP**	**Link 802.11**	**Phys. 802.11**
Jitter of sent packets Notified Events		D			S	
Retransmission avoidance Notified Events		D	D		S	
Acknowledgement Notified Events		/ /D	/D/S		D/S/	S//
Explicit congestion Notified Events		D		S distant		
Significant energy decrease Notified Events	D	D	D	D	D	S
Salvaging packet Notified events			D		S	
Received signal power at a node Notified Events			/D		D/S	S/
Sending jitter due to Route error Notified Events		D	S			
Sending jitter due to Route modification Notified Events		D	S			
Packet loss ratio Exported States	U	U			S	
SNR Exported States	U	U			U	S
RSS Exported States	U	U			U	S
BER Exported States	U	U			U	S
Energy level Exported States	U	U	U	U	U	S
Delay constraint RSVP Activable Service	X			X	X	
VMAC Activable Service	U/X				U/X	S
IntServ Activable Service	U/X			S/X		
DiffServ Activable Service	U/X			S/X		
FEC Activable Service		U			S	
ARQ Activable Service		U			S	

Table 1. Cross-Layer Atomic Action Array

Legend :

X : bidirectionnel Exchange S : source of the interaction D : destination of the interaction U : using CLAA data

4th. Functions Interaction array : DSR protocol case

Let us now produce a DSR function interaction array which presents the CLAA used by DSR, DSR functions and the others protocols. We will then explicit DSR functions that use each CLAA.

NE – CLAA	DSR Functions						Other Protocols			
	Routing	**Route Discov.**	**Trans. Ctrl**	**Route error**	**Salvaging**	**Segmen-tation**	**Applica-tion**	**TCP**	**Link 802.11**	**Physical 802.11**
Retransmission avoidance			D					D	S	
Acknowledgements			/D/S					//D	D/S/	S//
Significant energy decrease		D	D		D	D	D	D	D	S
Salvaging packet					D				S	
Received signal power at a node	/D								D/S	S/
Sending jitter due to Route error				S				D		
Sending jitter due to Route modif					S			D		
Energy level		U	U		U	U	U	U	U	S

Table 2 : Cross-Layer Atomic Action Notified Events (NE-CLAA) of DSR

For an efficient presentation of the method, we will limit protocol function interaction array to this example.

5th. Deducing of CLAA interaction models

A model of each kind of interaction can now be deduced. The model aims to show additional subsystems due to interactions and to explicit internal cross–layer mechanisms. For model readability, interface subsystem and distant subsystem are not represented in lower models, they are implicit.

1) Notified events CLAA case

The model given by Notified Events CLAA on the protocol stack, the system energy manager and a distant TCP can be represented as follows :

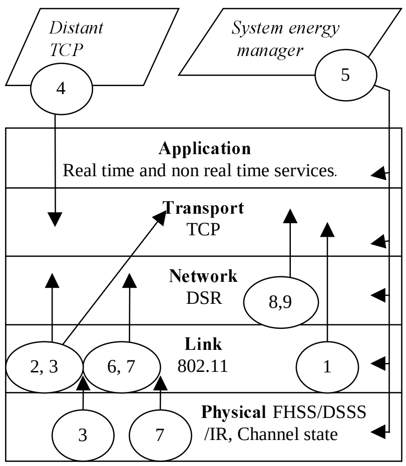

Figure 1 : interaction model of Notified events CLAA

Legend :

1. Jitter of sent packets Notified Events
2. Retransmission avoidance Notified Events
3. Acknowledgement Notified Events
4. Explicit congestion Notified Events
5. Significant energy drop Notified Events
6. Packet salvaging Notified Events
7. Received signal power Notified Events
8. Sending jitter due to Route error Notified Events
9. Sending jitter due to Route modification Notified Events

2) Exported states CLAA case

The model given by Exported States CLAA on the protocol stack, the environment subsystem and the system energy manager is as follows :

Legend :
1. Packet loss ratio Exported states /using
2. SNR (Signal to Noise Ratio) Exported states/using
3. RSS (Received Signal Strength) Exported states/using
4. BER (Bit Rate Error) Exported states/using
5. Energy level Exported states/using

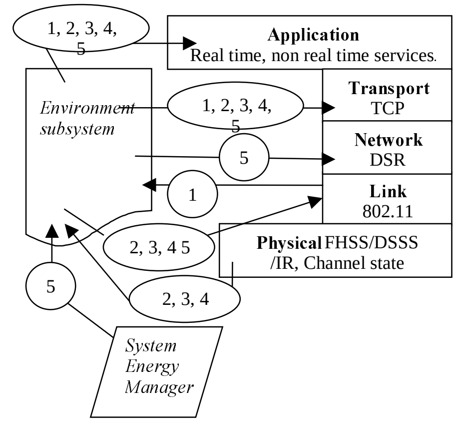

Figure 2 : interaction model of Exported states CLAA

3) Activable services CLAA case

By the same deducing mechanism of the interaction model, the model given by Activable Services CLAA on the protocol stack and the environment subsystem is as follows :

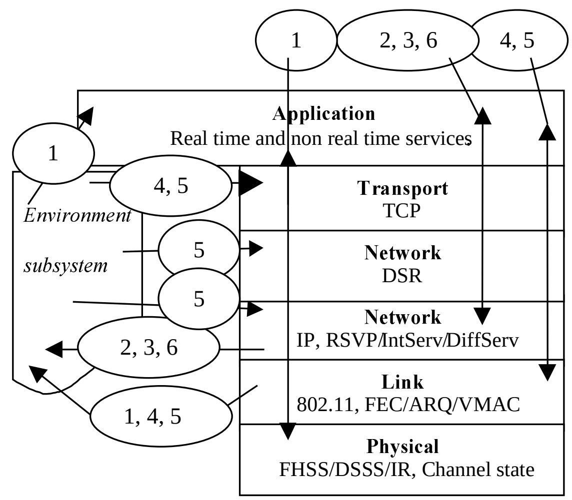

Figure 3 : interaction model of Activable services CLAA

Applications ask the concerned layer for services activation. When services are activated, environment subsystem is setup. The other layers use activation indicators or active services parameters to adapt their behaviour.

6th. Interactions description arrays

At this step, we are able to deduce the interaction description array of each protocol. Each array aims to explicit the possible exploitation of the CLAA by the influenced protocol's function.

For document readability, we limit the interaction description presentation to : the CLAA, the protocol function and the use of the CLAA by the function.

1) TCP protocol case

The TCP interaction description array that follows indicates the use of each CLAA by the TCP function and the modification of TCP source code that is proposed.

CLAA	TCP Function	TCP using of CLAA
Jitter of sent packets NE	Transferred data control	If a possible expiration of the ACK waiting timeout, then cancel and reset packet ACK waiting timeout (user timeout). Do not retransmit the packet during the new timeout. Do not invoke congestion control mechanism.
Sending jitter due to Route error NE		
Sending jitter due to Route modification NE		
Retransmission avoidance NE		Freeze the transmissions and retransmissions for the time specified in the message ⇒ reset all timeouts.
Acknowledgement NE		Anticipate the new data transmission if DSR protocol ensures that the destination is directly accessible.
Explicit congestion NE	Congestion control	Invoke the congestion control mechanism.
Significant energy decrease NE	Transferred data control	Modify the retransmission frequency and/or transmission output.
Packet loss ratio ES		Adjust the retransmission frequency and transmission output according to the high value of this parameter that is established by threshold (indicate channel state).
SNR ES		
BER ES		
RSS ES		Use the link layer ACK if the threshold of this parameter indicates that the destination node is directly accessible.
Energy level ES		Modify the retransmission frequency and the transmission throughputs according to high value of this parameter that is established by threshold.
FEC AS		Cancel the data checksum control mechanism.
ARQ AS	error correction	Cancel the data error correction function if the DSR protocol ensures that the destination is directly accessible.

Table 3 : Cross-Layer Atomic Actions of TCP

2) DSR protocol case

The following DSR interaction description array indicates the use of each CLAA by the DSR function and the modification of DSR source code that is envisaged.

CLAA	DSR Function	DSR using of CLAA
Retransmission avoidance NE	Transmission control	Freeze the transmissions and retransmissions for the time specified in the message ⇒ reset all timeouts.
Acknowledgement NE	Transmission control	Consider the packet link layer acknowledgement notification (SIFS interval using).
Significant energy decrease NE	Route discovery + transmission control + packet salvaging + fragmentation	Modify the retransmission frequency in the transmission control function, Modify the route discovery frequency, Cancel the packet salvaging function, Cancel the fragmentation function.
Salvaging packet NE	Packet salvaging	Activate the packet salvaging function (it will verify new route existence).
Received signal power at a node NE	Routing	Record the received signal power from a node as a route selection metric in routing table (it indicates nodes directly accessible).
Sending jitter due to Route error NE	Route error management	When receiving a route error message from a node that salvage a packet, or running route discovery again, notify the prorogation delay to TCP
Sending jitter due to Route modif NE	Packet salvaging	When receiving a packet salvaging message, notify the prorogation delay to TCP.
Energy level ES	Route discovery + transmission control + packet salvaging + fragmentation	According to the values of this parameter established by threshold : Modify the retransmission frequency in the transmission control function, Modify the route discovery frequency, Cancel the packet salvaging function, Cancel the fragmentation function.

Table 4 : Cross-Layer Atomic Actions of DSR

3) IP protocol case

By the same deducing mechanism, the IP interaction description array that follows indicates the use of each CLAA by IP function and the modification of IP source code that is proposed.

CLAA	IP Function	IP using of CLAA
Explicit congestion Notified Events	Routing	When congestion occurs and before queues overflowing, setup ECN bit in TCP header and convey packet to its destination.
Significant energy decrease Notified Events	Routing, fragmentation	Cancel the routing and fragmentation function, rely on DSR routing.
Delay constraint RSVP Activable Service	Routing	When receiving RSVP delay constraint parameters of an application, use RSVP request – answer scheme to send back explicit answer to the application.
IntServ Activable Service	Routing	When receiving IntServ delay constraint parameters of an application, use IntServ request – answer scheme to send back explicit answer to the application.
DiffServ Activable Service	Routing	When receiving DiffServ parameters form an application, use DiffServ traffic aggregate control and send back explicit answer to the application.
Energy level Exported States	Routing, fragmentation	According to the values of this parameter established by Threshold : cancel routing and fragmentation function, rely on DSR routing.

Table 4 : Cross-Layer Atomic Actions of IP

4) Link layer 802.11 protocol case

Using the previous mechanism, the link layer 802.11 interaction description array that follows indicates the use of each CLAA by the link layer 802.11 function and the modification of source code that is projected.

CLAA	802.11 Function	802.11 using of CLAA
Jitter of sent packets Notified Events	Control	Keep a trace of frames packet fragmentation, keep trace of each sending frame acknowledgement, establish global packet acknowledgement or sending packet shifting, If shift is established, notify the event to other layers.
Retransmission avoidance Notified Events	Control	uring "bad" channel state, very busy state of the system, or other restricting event, notify suspension of sending packet for required period of time to other layers.
Acknowledgement Notified Events	Control	Keep trace of frames packet fragmentation, keep trace of each sending frame acknowledgement, establish global packet acknowledgement, notify the event to other layers.
Significant energy decrease Notified Events	Control	Adjust the retransmission frequency.
Salvaging packet Notified Events	Control	If next hop is unreachable by RSS interpretation or acknowledgements missing, ask for DSR salvaging function.
Received signal power at a node Notified Events	Control	Keep a trace of frames packet fragmentation, keep RSS value for each acknowledged frame, calculate node RSS average and notify it to DSR.
Packet loss ratio Exported States	Control	Establish and setup those parameters values regularly : Service of non acknowledged packet / total number of sent packets Number of damaged received packets / total number of received packets during the period.
SNR Exported States	Control	Use this physical layer parameter to establish the packet loss or retransmission suspension.
RSS Exported States	Control	Keep trace of frames packet fragmentation, keep RSS value of each sent frame acknowledgement, establish node RSS average, expose it for other layers.
BER Exported States	Control	Use this physical layer parameter to establish the packet loss or retransmission suspension.
Energy level Exported States	Control	According to values of this parameter established by threshold : Modify the retransmission frequency.
VMAC Activable Service	Control	When activating VMAC service, setup activation indicator, setup regularly local delay estimations, jitters, collisions and packets loss ratio.
EC Activable Service	Control	When using FEC service for overcoming packet loss and bits corruption, setup the activation indicator.
ARQ Activable Service	Control	When using ARQ service for reliable transmission, setup activation indicator.

Table 6 : Cross-Layer Atomic Actions of IEEE 802.11

D. Conclusion

Cross–layer design is required for mobile ad hoc network to improve their performance contrary to wired networks that are not disabled by the same kind of failures. It is important to do that design in a standard framework to promote the evolution of protocols interaction models by taking into account new interactions or building new models for other protocols. This work aims to create useful formal steps within the seven steps upward design method that produces conceptual interaction models and efficient interaction description arrays, as it has been shown by the application of the method.

Our on-going work consists of implementing that cross–layer interaction models in NS (Network Simulator) environment to estimate the performance gains. The three subsystems (environment, interface, distant), have to be implemented first as additional objects in NS. Then every CLAA of each subsystem have to be placed into the ns source code. It is necessary to identify the existing or the additional fields or instructions or methods of the source protocol, the destination protocol and the subsystem of the CLAA. After each implementation of a CLAA, a simulation will be run to quantify the obtained gain.

E. References

[1] S. Shakkottai, T. S. Rappaport, P. C. Karlsson, "Cross-Layer Design for Wireless Network", IEEE Comm. Mag., October 2003.

[2] V. T. Raisinghani, A. K. Singh, S. Iyer, "Improving TCP performance over Mobile Wireless Environments using Cross – Layer Feedback", Personal Wireless Communications, 2002 IEEE International Conference on , 15-17 Dec. 2002 Pages:81 – 85.

[3] M. Conti, G. Maselli, G. Turi, S. Giordano, "Cross – Layering in Mobile Ad hoc Network Design", IEEE Computer society Magazine, February 2004.

[4] Q. Wang, M. A. Abu-Rgheff,, M. A., "Cross – Layer Signalling for Next – Generation Wireless Systems", IEEE WCNC 2003, Volume: 2 , 16-20 March 2003 Pages:1084 - 1089 vol.2.

[5] W. Li, Z. Bao – yu, "Study on Cross – Layer Design and Power Conservation in Ad hoc Network", IEEE PDCAT'2003, 27-29 Aug. 2003 Pages:324 – 328.

[6] K. Ramakrishnan, S. Floyd, D. Black. « The Addition of Explicit Congestion Notification (ECN) to IP. » RFC3168 September 2001.

[7] J. G. Jetcheva, Y. Hu, D. Johnson, D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)", Internet Draft, IETF MANET working group, Nov 2001

[8] D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks", Mars 1998 http://www.monarch.cs.cmu..edu/monarch -papers/

[9] T. Demir, "Simulation of Ad hoc Networks with DSR Protocol", May 2001. http://netlab.boun.edu.tr/papers/Iscis2001-DSR-TamerDEMIR+.pdf

[10] " The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)", 21 february 2002, www.ietf.org/proceedings/02mar/I-D/draft-ietf-manet-dsr-07.txt

[11] P. Almquist « Type of Service in the Internet Protocol Suite. ». RFC 1349, July 1992, www.ietf.org.

[12] Y. Bernet et al., "A framework for Integrated Services Operation over DiffServ Networks", RFC 2998, November 2000.

[13] M. Luby et al., "The use of Forward Error Correction (FEC) in Reliable Multicast", RFC 3453, December 2002.

[14] G. Fairhurst, L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)", RFC 3366, August 2002

[15] S. Dawkins et al., "End-to-End Performance Implications of Links with Errors", RFC 3155, August 2001

[16] A. Veres, A.T. Campbell, M. Barry, and L.H. Sun, "Supporting service differenciation in wireless packet networks using distributed control", IEEE JSAC, Vol. 19, No 10, pp. 2094-2104, October 2001.

[17] R. G. Gallager, "Energy limited channels : coding, Multi – access, and Spread Spectrum", 1998 Conf. Info. Sci Sys. Mar 1998

[18] Y. Min-hua, L. Yu, Z. Hui-min, "The IP Handoff between Hybrid Networks", IEEE PIMRC, Sept. 2002, Vol. 1, pp.:265 - 269

[19] H. Balakrishna, R. Katz, "Explicit Loss Notification and Wireless Web Performance", IEEE Globecom, Sydney, Nov 1998.

[20] C. Mazel, "évaluation des performances par simulations – application aux canaux de signalisation de systèmes radio téléphoniques", PhD. INP Grenoble, June 1988 (in French).