

MULTILEVEL NETWORK MODELING TO ACHIEVE CROSS LAYER MECHANISMS

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

*mahamadou.issoufou.tiado@enseeiht.fr, Riadh.Dhaou@enseeiht.fr, beylot@enseeiht.fr
ENSEEIHT – IRIT, 2 rue Camichel BP 7122, 31071 Toulouse Cedex France +33 56158 8306*

Abstract: A new way to improve the performance of ad hoc networks consists in using cross layer mechanisms. Currently, several protocols have demonstrated some reachable performance gain. Global integration on each level of the protocol stack has to be ensured. We present some efficient methods that may either produce or update cross-layer models. Those models, developed on different levels, allow an efficient organisation of the wireless systems and could take several forms. A cross-layer conceptual model is composed of: cross-layer interaction models and interactions description arrays. In this paper, we propose a method which has been applied to a chosen protocol stack.

Key words: Ad hoc Network, Cross-layer Method.

1. INTRODUCTION

Ad hoc Networks are wireless networks characterised by a dynamic topology, limited bandwidth, and energy consumption constraints. The link quality quickly changes and causes bursts of errors. Moreover, it has a large-scale variation: the average state of the channel depends on user position and on interferences¹. Wireless networks are generally less efficient than wired ones and classical protocols were not optimized in such a context. Some innovating techniques have to be developed to improve their performance. Protocols are generally designed independently. In an opposite operating mode, the cross-layer concept adapts them by sharing information between layers and by an overall optimisation instead of multiple optimisations at different levels. Several significant experiments were performed^{2,5,6,19}. The Cross-layer technique can be used by all the protocols if there are

interactions whose execution improves the performance of the global system. Specific protocols such as those improving TCP throughput have been proposed in ². Other large models implementing cross-layer interactions have been designed ^{3,4}. Nevertheless, because of the diversity of protocols, of their behaviour (even at the same layer), and of the possible interactions, it is important to design a generic method, 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, 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 integrating 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 ECN 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 of messages or signals exchanged between layers when particular events occur. In ⁴, different cross-layer methods such as “Packet Header” or “ICMP Messages” are presented. 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. The method may be applied to a given protocol stack or to integrate interactions in an existent cross-layer model. By considering CLIM as a conception and protocols and interactions as an implementation, we propose a reverse method, “reverse” materialises 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.

2. CROSS-LAYER DESIGN METHOD

2.1 Cross-Layer Atomic Action (CLAA) Concept

A Cross-Layer Atomic Action (CLAA) may be the setting or the utilisation of a layer parameter, the utilisation of other layers services, and layer events that have to be exposed to other layers. "Atomic" means that the action can not be divided into actions that do not impact the same protocols.

Actions 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, ...

Three kinds of CLAA may be distinguished:

Exported States CLAA (ES-CLAA) correspond to CLAA that export parameters to other layers. They may be used for admission control, QoS ... MobileMan system, distributed WCI servers ⁴ are Exported States models.

Notified Events CLAA (NE-CLAA): they 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".

Available Services CLAA (AS-CLAA): when specific mechanisms are developed to give interesting parameters or services to other layers.

2.2 Modeling of Interactions

ES-CLAA and AS-CLAA are local interactions within a node. They can be characterised by variables/environment parameters. For example, the activation of a service such as VMAC^{14,19} depends on a Boolean variable. Using this service, environment variables such as estimation of local delay, jitter and collisions, will be regularly updated. NE-CLAA includes both local interactions (significant energy drop notification) ⁵ and distant interactions (ECN) ⁶. Thus the cross-layer interaction model is divided into an environment subsystem including environment variables/parameters, an interface subsystem allowing communication between non adjacent layers and a distant subsystem allowing communication between layers of separate nodes. To fit the necessary standardisation of communication mechanisms, we suggest for the environment system input/output functions (read or write values). For the interface system, we can choose input/output functions or a standard protocol. The choice will be refined when simulation will be produced. Information conveyed by each interaction will allow defining a protocol or input/output functions. For the distant subsystem, standardised protocols will be used according to each CLAA.

2.3 Method steps

We propose the following seven steps method:

1. Select a protocol stack to produce the cross-layer Interaction Model;

2. Cross-layer Atomic Action census : it could be either a set of CLAA for which a performance evaluation has to be performed;
3. Production of Protocols interaction array that represents the interactions between CLAA (array lines) and protocols (array column). Each array cell can take those values: S (local to a node - by default, 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 of the CLAA. The number following "S/D" shows the chronology of conveying CLAA informations;
4. Production of the Protocol functions interaction array: protocols are divided into functions. The previous array is modified: each column corresponds now to a protocol function. This new array shows the functions to be modified for this CLAA implementation;
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;
6. Production of an interaction description array for each protocol. For each CLAA and each protocol, it indicates the origin of the CLAA, the source or the destination function, the kind of communication to use (direct, via subsystem) and its possible exploitation by the 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.

3. APPLICATION OF THE METHOD

3.1 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 contributes to the definition of the functions of the protocol in the layered protocol stack. These functions are influenced by the listed CLAA. This step aims to fix one or more wired protocol that may be adjusted to wireless environment within Cross – Layer mechanisms and concepts.

3.2 CLAA census

At this CLAA census step, all potentially wireless environment interesting events and variables are considered.

3.2.1 Available services: AS-CLAA

VMAC (Virtual MAC) ^{14,19} introduced at link layer is an example of AS-CLAA. 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. A virtual source adjusts its application parameters and determines the accepted service level. Additional services can be considered as AS-CLAA such as IntServ ^{4, 15}, RSVP ¹⁵, DiffServ ^{4,15} at network layer, FEC ^{4,16}, ARQ ^{4,16,17,18} at link layer.

3.2.2 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 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 become an interesting optimisation problem that requires co-operation between all the layers. Physical layer 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 allows to evaluate the distance between two nodes or to establish their direct access, for the needs of routing protocols. The link layer PLR ⁴, Physical layer SNR ⁴, The Physical layer BER ⁴ are other examples of ES-CLAA.

3.2.3 Notified events CLAA – NE-CLAA

ECN ^{1,6} and ELN ^{4,19} are NE-CLAA. When routers detect congestions, they set the ECN bit in the header. The receiver node reports the congestion to the sender by turning “on” the ECN bit in TCP header. The sender invokes then the congestion avoidance mechanism. In the case of wireless networks with infrastructure, a “snoop agent” can be introduced at the base station. It keeps the trace of non acknowledged segments lost on the wireless link. It sets the ELN bit in the duplicate ACK if it corresponds to a segment of the list. Then the sender will retransmit the next segment and do not take any congestion control action. The use of snoop agent on a mobile node is not appropriate because the sender can not 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. There are many other examples of NE-CLAA such as jitter of sent packets ⁴, retransmission avoidance ^{4, 19} (for instance in IP layer handoff ¹⁸, retransmission are needed and new traffic admission freezed), link layer acknowledgements (used for example by DSR and based on link layer grouping of acknowledged frames containing complete IP datagram ⁷⁻⁸ by using SIFS intervals of 802.11 ¹¹), significant energy lowering event ⁵, DSR packet salvaging ⁷⁻⁸, ...

3.3 Protocols Interactions Array

The CLAA will now be classified in a protocol interaction array which includes the protocols using them, the source and the destination of the interactions. To read this array, for example, by the physical layer SNR ES-CLAA, the 802.11 updates environment subsystem parameters that give SNR value. This parameter is used by 802.11 link layer, TCP and application layer. The source of received signal power NE-CLAA is physical 802.11. It informs 802.11 link layer which at sends a notification to DSR protocol.

Table 1. Cross-Layer Atomic Action Array

Cross – Layer Atomic Actions	Protocols					
	Appli.	TCP	DSR	IP	Link 802.11	Phys. 802.11
Jitter of sent packets NE		D			S	
Retransmission avoidance NE		D	D		S	
Acknowledgement NE		D3	D2, S3		D1, S2	S1
Explicit congestion NE		D		S distant		
Significant energy decrease NE	D	D	D	D	D	S
Salvaging packet NE			D		S	
Sending jitter due to Route error NE		D	S			
Sending jitter due to Route modification NE		D	S			
Packet loss ratio ES	U	U			S	
SNR, RSS, BER ES	U	U			U	S
Energy level ES	U	U	U	U	U	S
Delay constraint RSVP AS	X			X	X	
VMAC AS	U				U	S
IntServ, DiffServ AS	U			S		
FEC, ARQ AS		U			S	

3.4 Functions Interaction array

DSR protocol ensures functions such as routing, route discovery, transmission control, ... Let us now produce a DSR function interaction array which presents the CLAA used by DSR, DSR functions and the others protocols. For example, the transmission control function is the effective

network layer destination of the Retransmission avoidance NE-CLAA originated by 802.11 link layer. This function may be modified at the 6th step (Interaction description array). For an efficient presentation of the method, we will limit protocol function interaction array to one example.

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

NE – CLAA	DSR Functions						Other Protocols			
	Routing	Route Discov	Trans. Ctrl	Route error	Salvaging	Segment	App.	TCP	Link-802.11	Phys. 802.11
Retransmission avoidance			D					D	S	
Acknowledgements			D2, S3					D3	D1, S2	S1
Significant energy decrease		D	D		D	D	D	D	D	S
Salvaging packet					D				S	
RSS	D2								D1, S2	S1
Sending jitter due to R. error				S				D		
Sending jitter due to R. modification					S			D		
Energy level		U	U		U	U	U	U	U	S

3.5 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.

3.5.1 Notified events CLAA case

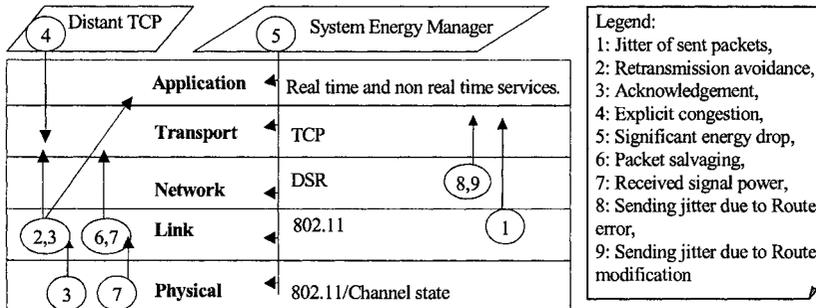


Figure 1: interaction model of Notified events CLAA

The system energy manager and a distant TCP can then be represented as follows. Note that the number represents the CLAA given in legend, the circle symbol the CLAA source layer and the arrow the CLAA destination layer. Here, the number 5 shows that if the energy level reaches a crucial threshold, the system energy manager sends this information to all the layers.

3.5.2 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. In this model, according to our CLAA census step, the DSR protocol uses only the battery level variable regularly updates by the system energy manager. Here, the number represents the CLAA given in legend, the circle symbol contains CLAA list concerned by the arrow, the arrow indicates the use or the update of variables from or towards environment subsystem. For example, in number 1, the 802.11 link layer updates the packet loss ratio variable that is use by TCP and application independently of other CLAA in the list.

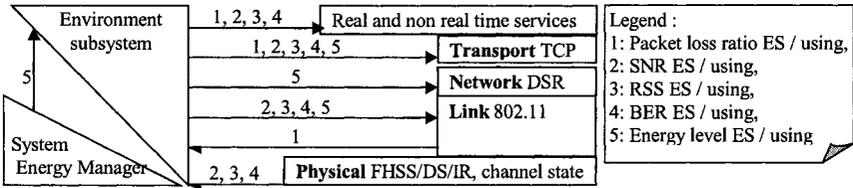


Figure 2: interaction model of Exported state CLAA

3.5.3 Available services CLAA case

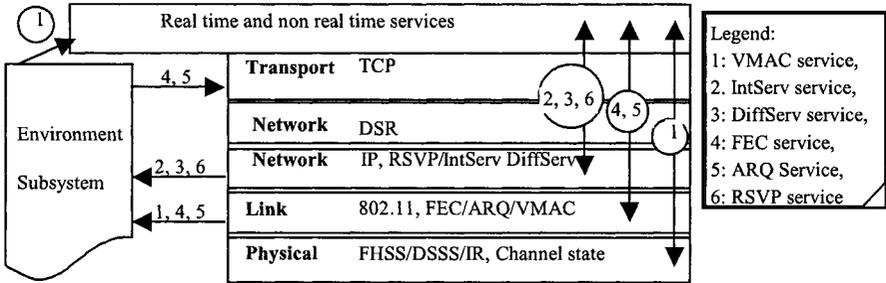


Figure 3: interaction model of Available services CLAA

By the same deducing mechanism of the interaction model, the model given by Available Services CLAA on the protocol stack and the environment subsystem is as follows. From or towards environment subsystem, the number, the arrows and the circle symbol have the same Exported states CLAA case signification. At difference, in the 3 other cases, the double arrows show information exchange between layers (through Interface Subsystem that do not appear for readability reasons). Theses exchanges aim at activating services in appropriate layer. For example, 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.

3.6 Interactions description arrays

Table 3 : Cross-Layer Atomic Actions of TCP

CLAA	TCP Function	TCP using of CLAA
Sending jitter due to Route error NE	Transferred data control	If a possible expiration of the ACK waiting timeout, then cancel and reset packet ACK waiting timeout. Do not retransmit the packet during the new timeout. Do not invoke congestion control mechanism.
Sending jitter due to R. modification NE		
Retransmis. avoidance NE		
ECN NE	Congestion ctrl	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, BER ES		Use the link layer ACK if the threshold of this parameter indicates that the destination node is directly accessible.
RSS ES		Modify the retransmission frequency and the transmission throughputs according to high value of this parameter that is established by threshold.
Energy level ES		Cancel the data checksum control mechanism.
FEC AS		Cancel the data error correction function if the DSR protocol ensures that the destination is directly accessible.
ARQ AS	error correction	

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. Only the TCP case is described in the present paper. Its interaction description array indicates the use of each CLAA by TCP functions and the modification of TCP source code proposed.

4. CONCLUSION

Cross-layer design is required for mobile ad hoc network to improve their performance. 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 at creating useful formal steps that produces conceptual interaction models and efficient interaction description arrays, as it has been shown in a significant example.

On-going work consists of implementing that cross-layer interaction models in NS (Network Simulator). 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. A simulation will be run to quantify the obtained gain of each CLAA.

5. REFERENCES

1. S. Shakkottai, T. S. Rappaport, P. C. Karlsson, "Cross-Layer Design for Wireless Network", IEEE Comm. Mag., Oct. 2003.
2. V. T. Raisinghani, et al., "Improving TCP performance over Mobile Wireless Environments using Cross – Layer Feedback", IEEE ICPWC , 15-17 Dec. 2002, p81–85.
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 Mar 2003 p1084-1089 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 WG, 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", May2001. <http://netlab.boun.edu.tr/papers/lscis2001-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.
12. Y. Bernet et al., "A framework for Integrated Services Operation over DiffServ Networks", RFC 2998, Nov. 2000.
13. M. Luby et al., "The use of FEC in Reliable Multicast", RFC 3453, December 2002.
14. G. Fairhurst, L. Wood, "Advice to link designers on link ARQ", RFC 3366, August 2002
15. S. Dawkins et al., "E.-to-E. Perf. Implications of Links with Errors", RFC 3155, 08/2001
16. Veres, A.T. et al., "Supporting service differentiation 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 98
18. Y. Min-hua, L. Yu, Z. Hui-min, "The IP Handoff between Hybrid Networks", IEEE PIMRC, Sept. 2002, Vol.1, pp265-269
19. H. Balakrishna, R. Katz, "Explicit Loss Notification and Wireless Web Performance", IEEE Globecom, Sydney, Nov 98.