A Secure Medium Access Control Protocol: Security versus Performances

Pierre SIRON, Bruno d'AUSBOURG

CERT-ONERA
Département d'Etudes et de Recherches en Informatique
2 avenue E.Belin
B.P. 4025
31055 Toulouse cedex FRANCE
email: {siron,ausbourg}@tls-cs.cert.fr

Abstract. Many systems were built in order to protect confidentiality of data and processes. This can be done by using multilevel architectures of machines and networks. But these architectures tolerate the existence of covert channels.We designed an architecture of a distributed security subsystem in order to avoid them, basing it on the use of secure dependencies. Controls exerted on dependencies can control exhaustively elementary flows of information. These controls are achieved by means of some hardware mechanisms which govern the access of hosts to the medium according to secure medium access control protocol (or SMAC). This approach implements in a straightforward manner some multilevel security conditions that ensure a very high degree of protection. We wanted to measure the real cost of introducing security inside a MAC protocol, by comparing under simulation the performances of the SMAC protocol with some other standard but insecure MAC protocols.

1 Introduction

This paper is the second in the series documents describing how one can build a security subsystem in order to implement secure dependencies over a distributed architecture over a LAN. This document builds on the more fundamental issues outlined in the first paper (see [4]) and assumes that the reader is familiar with the basic concepts of the approach. We strongly urge the uninitiated to refer to the above paper before proceeding to read any further.

In brief, we followed the approach[1] described in [4] basing it on using secure dependencies as defined by [1]. The idea is that controls which are exerted on dependencies can control exhaustively elementary flows of information. These controls are achieved by means of some hardware mechanisms which regulate the access of hosts to the medium according to a secure medium access control protocol (or SMAC).

This approach implements, in a straightforward manner, some multilevel security

[1] This project was supported by DGA in France

conditions that ensure a very high degree of protection. But it may be argued· that introducing any security controls inside a communication protocol is an unacceptable approach because performances may be degraded. So, we wanted to measure the real cost of introducing security inside a MAC protocol, by comparing performances of the SMAC protocol with some other standard but insecure MAC protocols.

The first part of the paper reminds of the approach and explains the choices of implementation that we made with respect to the required security. The second part describes the SMAC protocol and the third part shows some simulation results that seem very interesting because they thwart some too commonly accepted ideas.

2 Overview of the security approach

The enforced security is based on controlling dependencies that are involved in elementary operations over a LAN architecture. This control aims at authorizing only *secure* dependencies in the system with respect to the security policy.

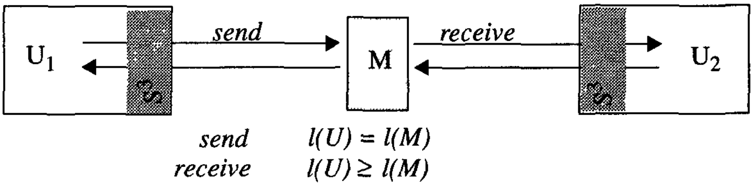

Fig. 1 Rules to access the medium in a network interface unit

In our case, the policy in question is the multilevel security policy, and then levels are assigned to the communication medium and to the network interface units (NIU). Dependencies that are involved in sending or receiving operations are secure if the two conditions on levels as illustrated on Fig. 1 are always satisfied.

The enforcement of these rules is performed by local security subsystems (or S^3) in every NIU. Send and receive operations on the medium can be performed by NIUs only when security conditions are verified. It follows that bidirectional exchanges can occur only between NIUs at the same level l, when this level l is assigned to the medium. Globally, levels are managed and assigned to the medium and to NIUs by a Centralized Security Station or CSS. Each local S^3 must be able to know the level value of the medium and of its own NIU.

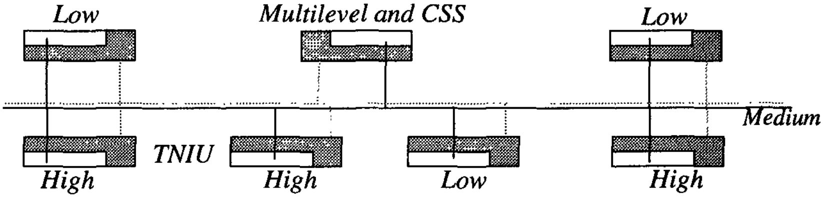

Fig. 2 Multilevel LAN

The enforcement of the access rules to the medium and the need to exchange security data between CSS and local S^3 in NIUs have an impact on the Medium Access Control entities in each NIU. The MAC protocol must be adapted in order to take these constraints into account.

This adapting process has been performed in the framework of a distributed architecture. In fact this protocol operates over a LAN architecture as described by Fig. 2 . Stations are connected to the medium through Trusted NIUs (or TNIUs) that are composed by the NIUs and their own local S^3. CSS is the security manager of the LAN. A multilevel station can be used to enforce the sharing of data between processes at different running levels. This secure multilevel station[3] can be used to implement the CSS functions.

Secure MAC or SMAC is the adapted MAC protocol we devised to regulate accesses of NIUs to the medium in accordance with the security rules.

3 The SMAC protocol

3.1 Global functioning: user and security modes

This protocol is in charge of regulating the access of network interface unit to the medium. But this access must be done in a secure way.

In particular, SMAC takes account of the time slicing that is exerted on the level of the medium. It manages also the exchanges of security data under the authority of the CSS. These data include particularly reservation data emitted from *local S^3* and level settings for the medium which are emitted from the *CSS*.

SMAC manages two functioning modes of the interface unit: a *user* mode and a *security* mode. In the security mode, only *local S^3* can use the medium M to exchange security data with the *CSS*. In user mode, send and receive operations can be performed by the interface units according to values of their own level and of the level of M. The CSS computes time slices for sessions of exchanges in user mode; each session corresponds to a value assigned to the level of M. These values are set in accordance with reservations previously received. At the end of a slice, the interface unit always returns to the security mode. In security mode, the *CSS* may ask to *local S^3* if reservations are pending. If yes, *local S^3* may answer by giving the content of their pending reservations. The protocol for this dialogue is a synchronous one. The *CSS* fixes a transmission slot for each local S^3 to answer and each local S^3 may answer during its reserved slot. The CSS broadcasts then a new value for the level of the medium and a new session in user mode is started. In user mode, a Medium Access Control (MAC) protocol arbitrates the access to the medium between units which are allowed to access it: this protocol is CSMA/CD in our case.

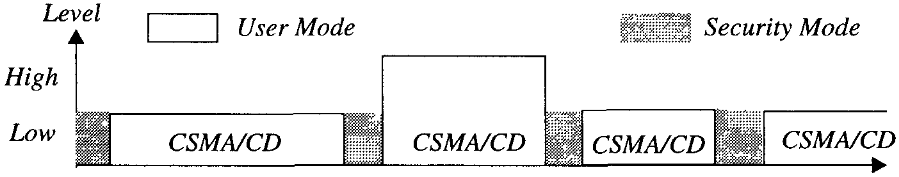

Fig. 3 Alternating modes of functioning

The SMAC protocol is similar to protocols used in the real time world where requirements on the amount of delay between the time a packet is ready and the time it is received at destination are stringent. In these protocols, some sources must reserve transmission slots before they can begin transmission [12].

3.2 Security mode

This mode of functioning is dedicated to the *Distributed Security Subsystem* or *DS^3*. Only the local S^3 on the network interface units and the *CSS* are allowed to access the medium that is used to implement the security subnetwork. In the first version of SMAC, the master of exchanges is the *CSS*; it can perform various security operations. Mainly:

- It can ask to several *local* S^3 (or to all of them) if pending reservations of levels were requested by the user on the host. This is done by broadcasting a *GET_REQUESTS* frame to these *local* S^3. This frame contains a slot number that is assigned to each *local* S^3 and that can be used to emit a pending request to the *CSS*.
- It can accept or reject a reservation request. This is done by sending a *ACK_LEVEL* or *NACK_LEVEL* frame to the *local* S^3 that emitted the request. When receiving the *ACK_LEVEL* frame, the *local* S^3 sets the level of the interface unit to the requested value of this level and sets a watchdog to the value of duration assigned by the user to the functioning of the interface at this level.
- It can change the current level of the medium by broadcasting a *SET_BUS_LEVEL* frame to all the *local* S^3. A duration value is embedded in the frame: this value gives the duration that is assigned to the next session of exchanges at the new current level. This value is kept by all the *local* S^3: they use it to set a watchdog. When this frame is received, the *local* S^3 enter the user mode, and the network interface units are set in accordance to their own level and the new current level. A new session of exchanges can start. When the watchdog indicates that time has elapsed, the *local* S^3 enters the security mode.

3.3 User mode

This mode of functioning is dedicated to the network interface units. They can perform send or receive operations in accordance with settings done by the *local* S^3 with respect to the values of the levels assigned to the interface and to the medium. The enforced MAC protocol is CSMA/CD.

This user mode lasts the time that was fixed by the *CSS* when sending the *SET_BUS_LEVEL* frame. The switching of modes can cause an interruption in sending operations. This fact is assimilated to a collision in the CSMA/CD protocol that will retry to send the frame at a next session that will be running at the same current level.

This user mode is particularly concerned with performances. Hence, it has to be submitted to performance evaluation measures.

4 Performance evaluation

We carried out a performance evaluation of the SMAC protocol. This effort had several main objectives:

- To estimate the cost of security, in particular by comparing the secure protocol and the standard but not secure CSMA-CD protocol. We could hope for a performance advantage, due to the partitioning of the users into several communicating classes (one per security level).
- To calculate some parameters for the SMAC protocol: for example the time to assign to each security level in the communication medium multiplexing (the *slot time*).
- To examine the overall performance of the network concerning contention (number of collisions) and effective utilization.

Before performing any experiments on a real system, we chose to proceed to exhaustive simulations. The following sections describe the simulation environment, the simulation model, the performed experiments, the first results of our evaluation and the work that remains to do.

4.1 Simulation environment

The quality of a simulation depends partially on the software tools which are used. The list of requirements may be long [10]: reliability, readability, adaptability, portability, efficiency. These requirements are more or less achieved easily if software is well structured. Commercial products meet them, but we chose to use the MIT Network Simulator NETSIM (cf. [7]) for two main reasons:

- NETSIM is available by ftp.
- NETSIM was used in several research projects (cf. [9]) and in particular in a study [12] that is close to this one, an extension to the CSMA-CD protocol to support real-time traffic.

NETSIM is a discrete-event simulator. It can simulate anything that can be modelled by a network of components that send messages to one another. The components schedule events in order to make appear the evolving of the simulation. The model being simulated and the action of components are determined entirely by the code that controls the components, and not by the framework of the simulator. A new code must be written to develop a new component: data structures and action functions are written in C language in accordance with a given frame.

The program provides the user with the means to display (XWindow system) the topology of the network and the parameters of the simulation, to modify them, and to save and load the various configurations. It allows also a user to control the simulation process, to log various events and to produce statistics.

4.2 Simulation model

A simulation requires models that can be viewed as simplified representations of complex processes. By definition, they do not express the full behaviour of processes but they capture the most significant phenomena that may affect the predicted results.

We limited the models to the exchanges of packets rather than elementary exchanges of bits. The granularity of simulation is then greater, but that is not really a problem, because exchanges of packets are always performed in accordance with the security rules. Therefore the simulation time is shorter and more acceptable and the model is closer to the user point of view. We simulated also the collisions between concurrent communications.

The main problem was to simulate the SMAC protocol but also the standard CSMA-CD protocol. NETSIM contains a detailed implementation of this last protocol developed at the University of Washington (cf. [6]) and many other components of the local network world.

The components that were used with minimal changes only were:

- UWETLINK: It models the passive Ethernet cable with its physical characteristics such as propagation delay, channel contention, and delivery of packets to the destination interface.
- HOST: Implements the standard CSMA-CD protocol at the Medium Access Sublayer (MAC), connected to UWETLINK.
- TCP: Is an implementation of the TCP protocol (Transmission Control Protocol) and is connected to an HOST component. This TCP component includes the latest version of TCP enhancements including the slow-start congestion avoidance and retransmission timer estimation algorithms [11].
- USER: The supplier and consumer of data for the TCP module.
- PSOURCE:A simple Poisson traffic source, directly connected to an HOST.
- SINK: A packet sink.

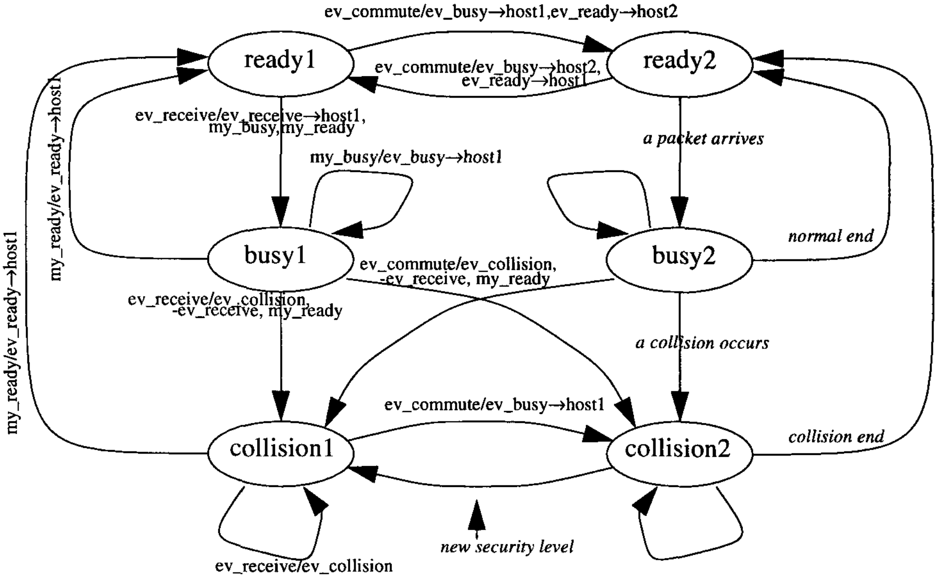

Fig. 4 SUWETLINK automaton

Let us detail the UWETLINK component, that was mainly modified in order to model the SMAC protocol. In this close shot, the events manipulated by the simulator are of two types:

- The external events (for the communication between the components):
  - EV_RECEIVE: A packet reception signal.
  - EV_READY: Component ready signal.
  - EV_BUSY: Component busy signal.
  - EV_COLLISION: A collision has occurred.
- And the private events:
  - MY_BUSY: A packet is sent.
  - MY_READY: End of the sending.

The automaton, that was simplified for understanding needs, is shown on the left side of the Fig. 4 . The EV_RECEIVE reception in the ready state will cause at a given time a EV_RECEIVE reception for the next host, and the MY_BUSY and MY_READY sending for internal purposes. The MY_BUSY reception indicates the end of the collision detection period, and the component notifies the other hosts of the channel acquirement (EV_BUSY event). A second EV_RECEIVE reception announces a collision case, and implies to send EV_COLLISION and to remove the sent EV_RECEIVE. At the MY_READY event reception, the channel becomes free and the EV_READY signal is the opposite of the EV_BUSY one.

To simulate the SMAC protocol, the most significant problem is to represent the temporal multiplexing of the network, and therefore to develop a new component, called SUWETLINK (S for Secure). The new automaton corresponds to the whole Fig. 4 for two security levels. The states are now subscribed by the security level and a new event, denoted by EV_COMMUTE, indicates the context switches. The trick is to consider the busy state of the HOST component either as the standard state (an host is emitting) or as a state where the medium is at a different security level and where the communications are prohibited. A switch during a communication interrupts it, and, in order to simplify, this is considered as a new case of collision (outside the usual collision detection period).

Fig. 5 and Fig. 6 visualize the simulated worlds. It was not necessary to calibrate the components. The key point here was to compare the CSMA-CD and SMAC protocols with models of the same complexity.

4.3 Workload model

The numbers of subscribers and security levels vary in the simulated worlds, but also the simulation workload, which consists of two classes: the strong and the weak workload models.

Strong workload

Simultaneous file transfer sessions, and consequently TCP and USER components, are used. An example is given in Fig. 5 : user1 sends a file to user2, 3 to 4, 5 to 6 and 7 to 8, and each users pair belongs to a different security level. For a file size of 20000 bytes, the results show a network utilization between 60 and 70%.

These worlds are rather realistic if we consider that the algorithms of the TCP components are close to the real implementations

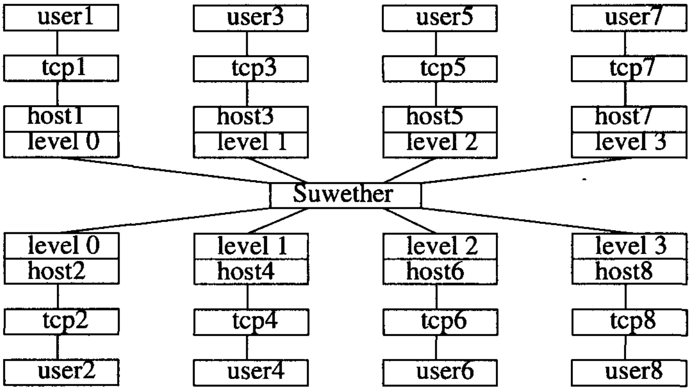

Fig. 5 TCP communication (20000 bytes, 8 points, 4 levels) example: the simulated world.

Weak workload

Packets generators use a Poisson distribution and send data to simple sinks. These worlds can model interactive sessions. An example is given Fig. 6 where two security levels are defined. It illustrates a network utilization less than 5%.

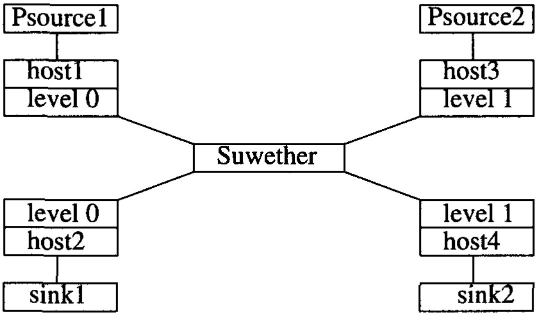

Fig. 6 Weak workload communication (4 points, 2 levels) example: the simulated world.

4.4 Performance measures

Among many results that were produced, we emphasized two parameters in the case of strong workload:

- Simulation time: It is the simulated time since the instant 0 until the instant of the last file transfer session. In this paper all the transfers have the same beginning and the same length, so the simulation time depends on the effective network utilization and the global number of packets, that may vary in function of the retransmissions and the TCP optimizations.
- Collisions: We measured the total number of collisions for each experiment, which includes usual collisions between packets and security level switching collisions.

In the case of the weak workloads the simulation time is an input, and in this time sources and sinks are running, Therefore the main performance measurement is the mean packet transfer time: this value is the difference between the reception by the sink and the emission by the source.

4.5 Experimental results

Only two significant results among the first experiments are showed in this paper. Parameters are standard: for example the maximal packet length is 1024 bytes, and the network data rate is 10 Mbits/s (Ethernet). Experiments were repeated with several security slots (the duration where the medium security level is constant): 1, 2, 3, 4, 5 and 10 ms. The null value can be considered as the standard protocol.

The results in Fig. 7 correspond to the strong workloads (TCP world). These curves are noteworthy, the SMAC protocol is more efficient than the CSMA-CD protocol in saturated and well-balanced worlds where the problem is to share the bandwidth.

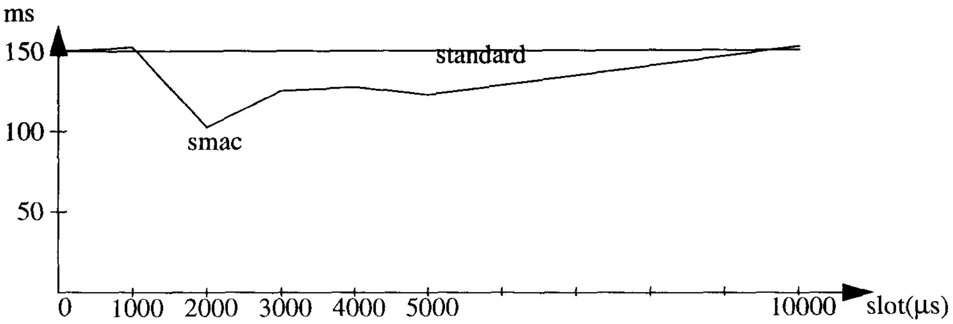

Fig. 7 TCP communication (20000 bytes, 8 points, 4 levels) example: the simulation time.

The network efficiency depends on the packet retransmission rate (and depends also on the number of dropped packets in the case of multiple checks). The number of collisions is a good indicator and this number is decreasing (see Fig. 8). This constitutes the main reason for the better performance of the CSMA-CD protocol.

We expected this result because there is no interference between communications of different security levels due to the security conditions that are enforced. Nevertheless the switching of security level introduces new collisions, when the commutation occurs during a packet transmission. But, globally. we observed on the simulator, that the number of collisions is lower.

The number of collisions is higher for 3 ms than for 2 ms (idem for 3 ms vs 5ms) and this fact could appear amazing. The explanation is in the fitting between the slot time and the effective time to transmit one, two, three,... packets. In the first case the commutation occurs rather when the host is processing, and there is no collision. In the second case the commutation occurs more often when the host is sending a packet.

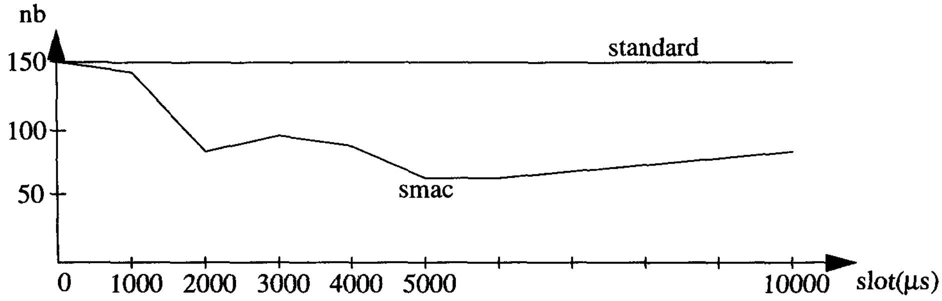

Fig. 8 TCP communication (20000 bytes, 8 points, 4 levels) example: the number of collisions.

We observed the same kind of results with various numbers of hosts and there is often a gain in performances when the security levels are more numerous. The Fig. 9 gives results for the same world but with two security levels. If we compare the SMAC protocol curves in the two cases, the simulation time is generally greater. For slot = 4ms, the simulation time equals 118ms for SMAC and 152ms for the standard protocol.

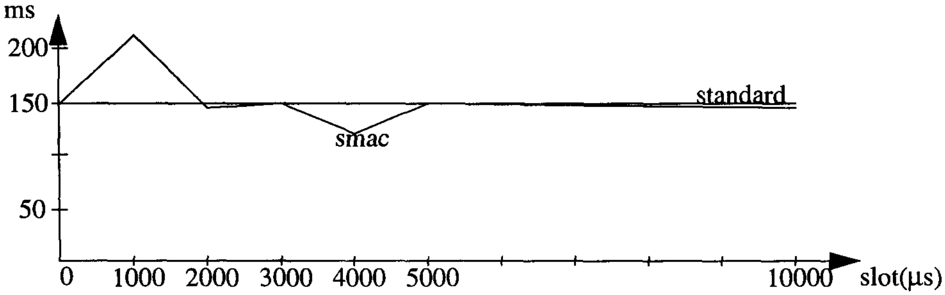

Fig. 9 TCP communication (20000 bytes, 8 points, **2 levels**) example: the simulation time.

As expected the number of collisions (Fig. 10) is greater, because there are now more users of the same security level, which compete for the channel possession. For slot = 2ms, we obtain 84 collisions for 4 levels and 153 collisions for 2 levels. The optimal value to assign to the slot duration is not easy to find, it varies all along the experiments.

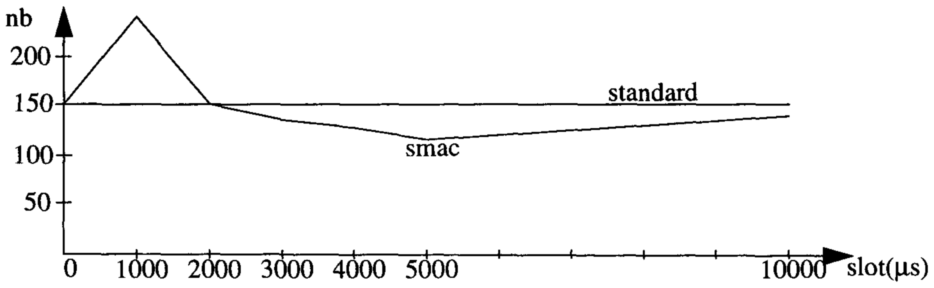

Fig. 10 TCP communication (20000 bytes, 8 points, **2 levels**) example: the number of collisions.

For the weak workloads (Fig. 11) the results are less favourable. The mean packet transfer time is steadily growing with the security slot and the number of levels. On the given example the host must wait every two times for the good slot to succeed in the packet transmission. Nevertheless the transfer times remains transparent to the user.

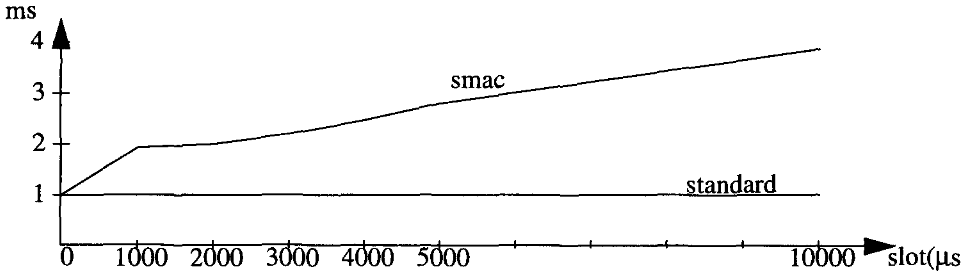

Fig. 11 Weak workload communication (4 points, 2 levels) example: the mean transfer time.

The weak workload explains the very little number of collisions (Fig. 12) in the standard case. For the SMAC protocol the collisions come from the security level switching. The mean transfer time is 1 ms, we have about 1200 packets in this test, and we can verify that the collision rate is about 1/2 for slot=1ms.

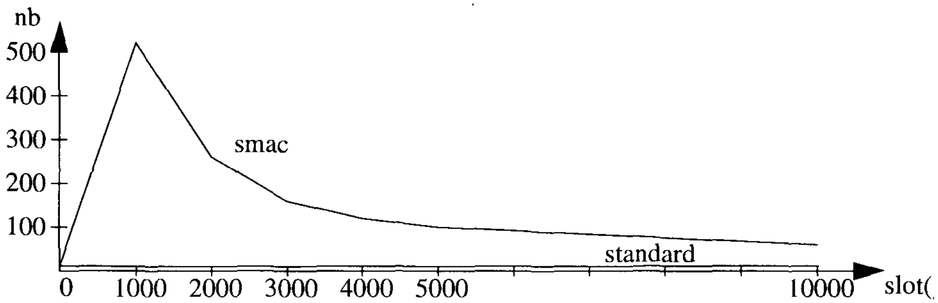

Fig. 12 Weak workload communication (4 points, 2 levels) example: the number of collisions.

We could here develop simplified formulae, that well describe the expected results. With the events traces, this is a kind of validation of the simulator.

4.6 Comparison with a tokenring network

NETSIM has a component that simulates a tokenring network. It was interesting to use it, although we did not verify in details its modelization nor we calibrated it.

The following table shows results for the standard CSMA-CD, the secure SMAC and the standard Tokenring protocols. The simulated worlds include 8 points. The number of security levels is 2 and the security slot is fixed to 2 ms. The strong workload corresponds to 4 file transfers of 100 kbytes.

Table 1: Performance results

	CSMA-CD	SMAC	TOKENRING
simulation time TCP	539 ms	482 ms	465 ms
mean transfer time POISSON	1 ms	2.7 ms	10.2 ms

The results for SMAC are between the results of the CSMA-CD and Tokenring protocols. For the strong workload, SMAC is close to the best protocol: Tokenring. In this case the data transfers are well scheduled by the circulation of a token on a ring. We can do an analogy to the switches of the security levels.

For the weak workload, SMAC is closer to CSMA-CD. This result allows us to differently appreciate the results of the previous paragraph. SMAC seems to keep the advantage of Ethernet, and the waiting time for the good security level seems lower than the waiting time for the token.

5 Conclusion

In this paper, we described a particular Medium Access Control protocol that takes into account of the enforcement of some sufficient conditions of multilevel security over a local area network. The protection is founded on controlling causal dependencies inside the system. Controls are performed exhaustively at the lowest level of the architecture: inside the hardware layer. So, every information flow is controlled and there is no potential covert channel. The SMAC protocol offers a standard functioning to users, CSMA/CD in our case, but this functioning may be sliced into various sessions running at different current levels of security. This fact, necessary to obtain a very high degree of protection, may be criticized by arguing that these rules of functioning decrease performances.

Several lessons were learned from the results of simulation that were reported here. Firstly, when the network is used at a single public level, it functions in a standard way and the rules to access the medium are in accordance with the CSMA/CD protocol. When the network is used in a multilevel functioning mode, two cases may occur.

The first case consists in a good use of the multilevel features: the network is effectively used to exchanges data (files for example) at different levels. This case was modelled by the strong workload world. In this case, there is a gain of throughput, because SMAC organizes the accesses to the medium. The second case consists in a bad use of the multilevel features: the network is used in a sporadic manner at different levels. This case was modelled by the weak workload world. In this case, there is effectively a transfer time growing. But the results of comparison with the Token Ring protocol, for example, show that the advantage of CSMA/CD in case of weak loads is not lost with SMAC. So, extending CSMA/CD with SMAC in order to get secure exchanges over a network may produce an increase of performances in the best case, and preserves the benefits of CSMA/CD over other protocols in the worst case.

We emphasize the reuse of the original TCP and HOST (CSMA-CD) components. We added only the security level parameter to them. This is also a good point before a real implementation. It is probably a good thing to obtain good performances with components that ignore the underlying multiplexing (the delay retransmission computation and the round trip time estimation of a packet could be no more appropriate).

Of course, a lot of work remains to do to complete these first results: in particular to accumulate the simulation experiments. But they constitute an encouragement for designers of secure systems because they make appearing that security and performances are not necessary antagonistic.

6 References

1. P. Bieber, F. Cuppens: A logical view of secure dependencies. In *Journal of Computer Security*, Vol. 1, Nr. 1, IOS Press, 1992

2. D. E. Bell and L. J. Padula: Secure Computer Systems: Unified Exposition and Multics Interpretation, MTR-2997, MITRE Corporation, Bedford, Mass. (1975).

3. B. d'Ausbourg and J.H. Llareus: M^{2}S: A machine for multilevel security, *European Symposium on Research in Computer Security, ESORICS92*, Toulouse, France, 1992

4. B. d'Ausbourg: Implementing Secure Dependencies over a Network by designing a Distributed Security SubSystem, *ESORICS94,* Brighton, UK,1994

5. G.Eizenberg: Mandatory policy: secure system model. In AFCET,editor, *European Workshop on Computer Security*, Paris,1989.

6. H. Golde: University of Washington version of MIT Network Simulator. October 1991. (available by anonymous FTP from june.cs.washington.edu).

7. A. Heybey: MIT Network simulator. *MIT Laboratory for Computer Science*, 1988.

8. G.King: A survey of commercially available secure LAN product, *in Proc. Int. IEEE Conf. on Computer Security Applications*, Tucson, Arizona, December 1989

9. MIT: NETSIM mailing list, info-netsim@lcs.mit.edu.

10. G.R. Sherman: The quality of a scientific simulation in *SIMULETTER vol 15 , n 3,* July 1984.

11. Van Jacobson: Congestion avoidance and control. *in Proc. of ACM SIGCOMM'88 Symposium*. pp. 314-329, August 1988.

12. R. Yavatkar, P. Pai and R. Finkel: A reservation based CSMA Protocol for Integrated Manufacturing networks, *Tecn. Rep. 216-92, Department of Comp. Sc., Univeristy of Kentucky,* Lexington, KY.