802.1X et la sécurisation de l'accès au réseau local

Luc Saccavini
Direction des Réseaux et Systèmes d'Information, INRIA
Luc.Saccavini@inria.fr
Date : 15 octobre 2003

Résumé

Cet article présente en détail le protocole 802.1X, dont l'objectif est d'autoriser l'accès physique à un réseau local après une phase d'authentification. Ce protocole s'appuie sur l'encapsulation EAP pour mettre en relation le serveur d'authentification et le système à authentifier. Le protocole EAP spécifié par le RFC2283 est décrit succinctement. Enfin, les faiblesses et limites d'usage de 802.1X sont analysées, ainsi que les dernières évolutions du standard actuellement en cours d'élaboration.

Mots clefs

802.1X, Ethernet, EAP, Sécurité, Radius, Authentification, Accès, Réseau

1 Pourquoi le protocole 802.1X ?

Ce standard [1], mis au point par l'IEEE en juin 2001, a comme objectif de réaliser une authentification de **l'accès au réseau** au moment de la connexion physique à ce dernier. Cette authentification intervient avant tout mécanisme d'autoconfiguration (ex. DHCP, PXE...). Dans la plupart des cas, le service autorisé en cas de succès est le service Ethernet. L'objectif de ce standard est donc uniquement de valider un droit d'accès physique au réseau, indépendamment du support de transmission utilisé, et en s'appuyant sur des mécanismes d'authentification existants.

2 Le modèle et les concepts du standard IEEE

Dans le fonctionnement du protocole, les trois entités qui interagissent (Figure 1) sont le système à authentifier (*supplicant*), le système authentificateur (*authenticator system*) et un serveur d'authentification (*authentication server*). Le système authentificateur contrôle une ressource disponible via le point d'accès physique au réseau, nommé PAE (*Port Access Entity*). Le système à authentifier souhaite accéder à cette ressource, il doit donc pour cela s'authentifier.
Dans cette phase d'authentification 802.1X, le système authentificateur se comporte comme un mandataire (*proxy*) entre le système à authentifier et le serveur d'authentification ; si l'authentification réussit, le système authentificateur donne l'accès à la ressource qu'il contrôle. Le serveur d'authentification va gérer l'authentification proprement dite, en dialoguant avec le système à authentifier en fonction du protocole d'authentification utilisé.

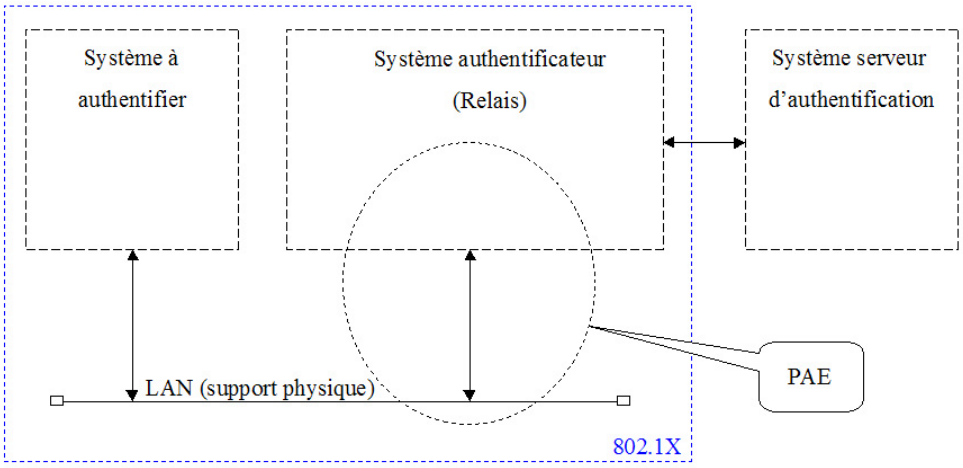

Figure 1 : les trois entités qui interagissent dans 802.1X

C'est au niveau du PAE que porte l'essentiel des modifications introduites par le protocole 802.1X.

Dans la plupart des implémentations actuelles, le système authentificateur est un équipement réseau (par exemple un commutateur Ethernet, une borne d'accès sans fil, ou un commutateur/routeur IP), le service dont il

contrôle l'accès est le service Ethernet (ou le routage des datagrammes IP). Le système à authentifier est un poste de travail ou un serveur. Le serveur d'authentification est typiquement un serveur Radius, ou tout autre équipement capable de faire de l'authentification.

3 Le point d'accès au réseau (PAE)

La principale innovation amenée par le standard 802.1X consiste à scinder le port d'accès physique au réseau en deux ports logiques, qui sont connectés en parallèle sur le port physique. Le premier port logique est dit « contrôlé », et peut prendre deux états « ouvert » ou « fermé ». Le deuxième port logique est, lui, toujours accessible mais il ne gère que les trames spécifiques à 802.1X.

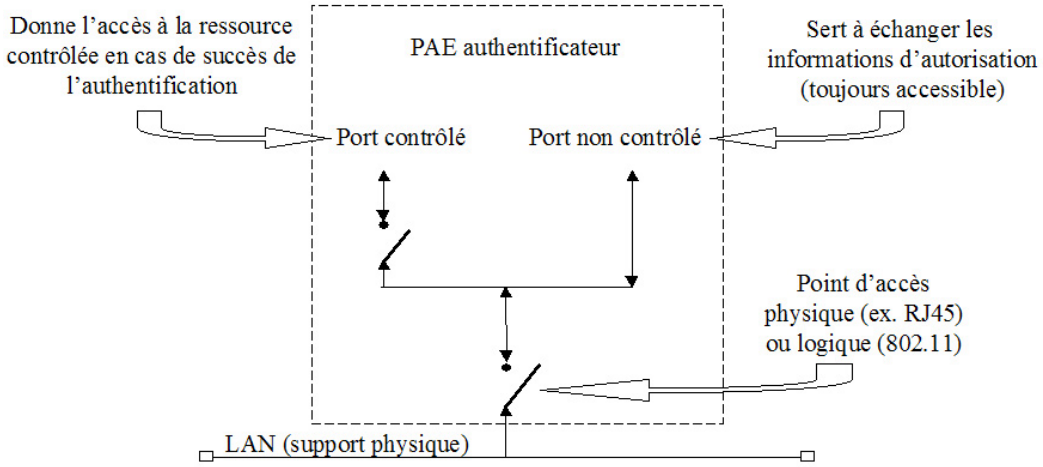

Figure 2 : le PAE

On notera que ce modèle ne fait pas intervenir la nature physique de la connexion. Elle peut être matérialisée par une prise RJ45 (cas d'un support de transmission cuivre), des connecteurs SC, MT-RJ (cas d'un support de transmission en fibre optique) ou par l'accrochage logique au réseau (cas d'un support de transmission hertzien en 802.11{a,b,g}).

4 Fonctionnement général du protocole

4.1 La circulation des paquets d'authentification

Le standard 802.1X ne crée pas un nouveau protocole d'authentification, mais s'appuie sur les standards existants. Le dialogue entre le système authentificateur et le système à authentifier se fait en utilisant le protocole EAP [2] (PPP *Extensible Authentication Protocol*) défini par le RFC2284[2]. Les paquets EAP sont transportés dans des trames Ethernet spécifiques EAPOL (*EAP Over Lan*) qui sont marquées avec le numéro de type (Ethertype) égal à 88FE, ce qui permet une encapsulation directe de EAP dans Ethernet. Le dialogue entre le système authentificateur et serveur d'authentification se fait par une simple « ré-encapsulation » des paquets EAP dans un format qui convient au serveur d'authentification, sans modification du contenu du paquet par le système authentificateur (voir Figure 3). Ce dernier effectue cependant une lecture des informations contenues dans les paquets EAPOL afin d'effectuer les actions nécessaires sur le port contrôlé (blocage ou déblocage). Ainsi, le système authentificateur débloquera le port contrôlé en cas d'authentification réussie, ou il le bloquera s'il y a une demande explicite en ce sens du système à authentifier comme on le verra un peu plus loin.

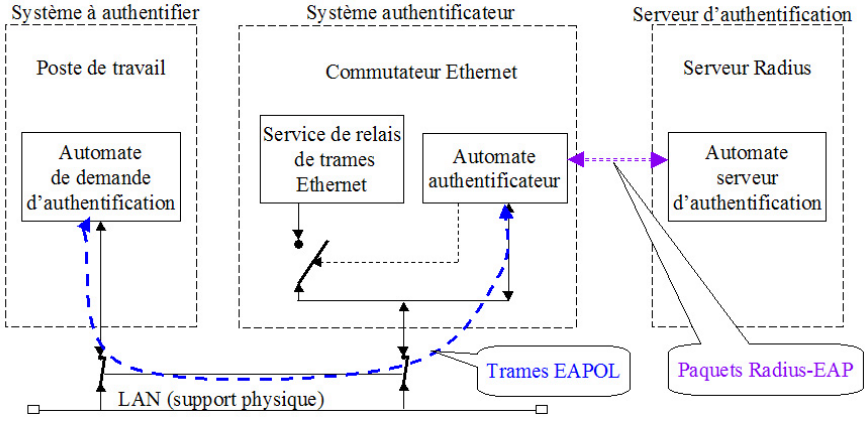

Figure 3 : 802.1X et serveur d'authentification

4.2 Les paquets EAP et EAPOL

Le RFC2284 définit les quatre types suivants de paquets EAP (champ *code*, sur un octet)

- *Request* : le système authentificateur émet une requête d'information,
- *Response* : réponse du système à authentifier à un paquet *Request,*
- *Success* : le système authentificateur indique une authentification réussie,
- *Failure* : le système authentificateur indique un échec de l'authentification.

Le paquet EAP contient aussi un champ *identifier* (sur un octet) pour identifier une session d'authentification. Dans le cas de paquets de type *request* ou *response*, un champ (*type)* définit la nature des informations qui sont contenues dans le paquet. Par exemple :

- *Identity* : chaîne de caractères identifiant l'utilisateur (par exemple une adresse mail, un nom de login, etc.),
- *Notification* : chaîne de caractères envoyée à l'utilisateur final,
- *Nak* : refus d'un type d'authentification et proposition d'un autre,
- *MD5-Challenge* : défi (*challenge*) ou réponse (idem authentification Chap),
- *One-Time-Password* : défi ou réponse,
- *Generic Token Ring Card* : défi ou réponse,

……

L'encapsulation EAPOL est définie pour les trois types de réseaux suivants : 802.3/Ethernet MAC, 802.5/Token Ring et FDDI/MAC. Dans le cas d'Ethernet, les paquets EAP tels que définis précédemment sont insérés dans une trame dont le champ type à la valeur 88-8E. Ces trames EAPOL peuvent être des quatre types suivants :

- *EAP-Packet* : paquet de dialogue EAP,
- *EAP-Start* : authentification explicitement demandée par le système qui s'authentifie,
- *EAP-Logoff* : fermeture du port contrôlé explicitement demandée par le système qui s'authentifie,
- *EAPOL-Key :* si chiffrement disponible (ex 802.11),
- *EAPOL-Encapsulated-ASF-Alert*.

4.3 Exemple de session 802.1X/EAP

Avant la connexion du système à authentifier au port physique du PAE du système authentificateur, le port contrôlé de ce dernier est bloqué, et seul le port non contrôlé est accessible. Lorsque le système à authentifier se connecte au port physique du système authentificateur, il reçoit un paquet EAP l'invitant à s'authentifier. Sa réponse est reçue sur le port non contrôlé du système authentificateur, puis est retransmise au serveur d'authentification par ce dernier. Par la suite, un dialogue s'établit entre le serveur d'authentification (voir Figure 4) et le système à authentifier par le biais du relais offert par le port non contrôlé du PAE du système authentificateur.

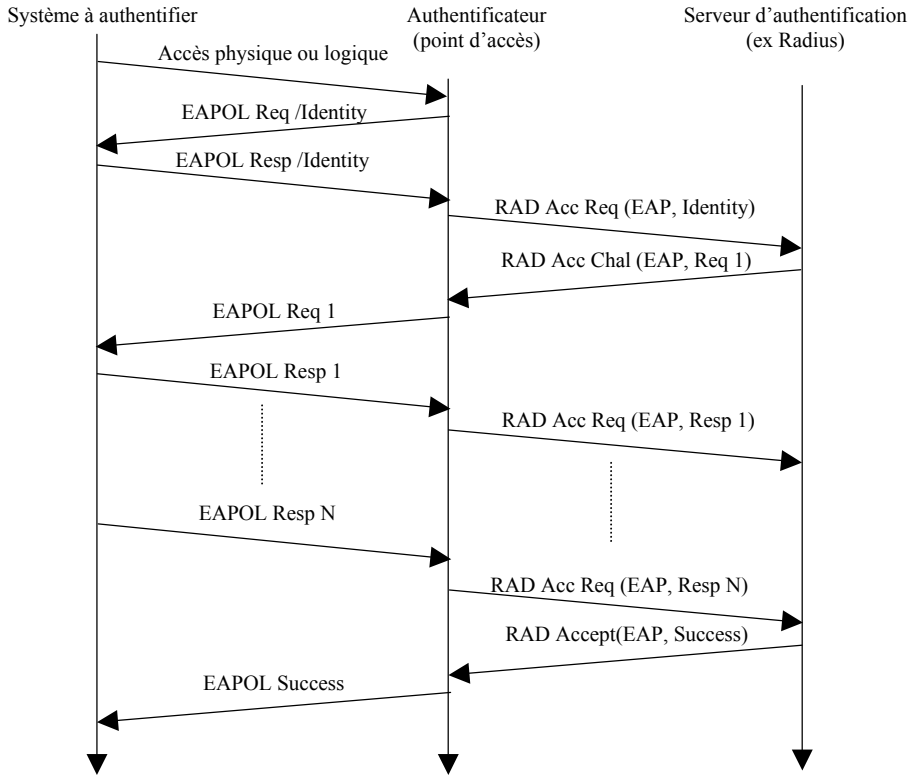

Figure 4 : Séquence d'authentification

Quand l'automate 802.1X du système authentificateur voit passer un acquittement positif d'authentification (en provenance du serveur), il débloque son port contrôlé (interrupteur fermé), donnant ainsi au client authentifié l'accès au service.

À partir de cet instant, le schéma logique du PAE du système authentificateur devient tel que décrit ci-dessous (voir Figure 5), et le trafic Ethernet est assuré normalement.

Cependant, les automates implémentant le protocole 802.1X restent actifs et peuvent à nouveau réactiver un processus d'authentification en cas, par exemple, de demande explicite du client ou de déconnexion physique au réseau.

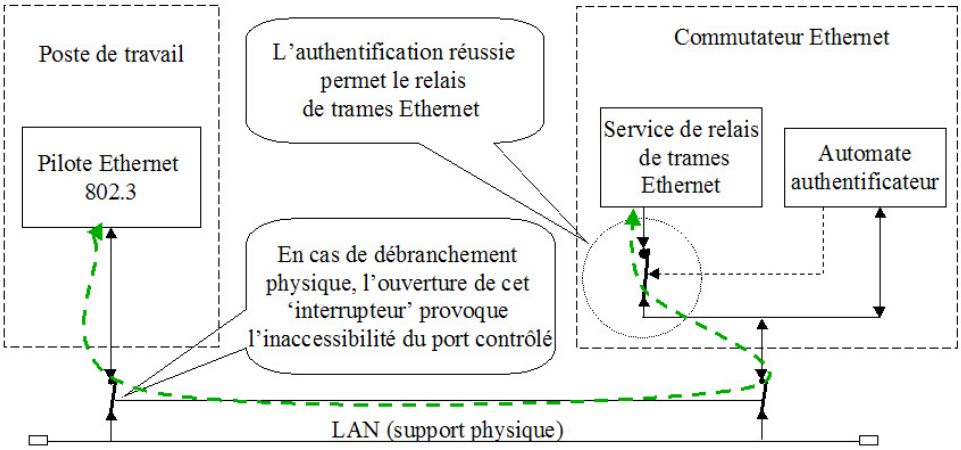

Figure 5 : Exemple de situation après une authentification réussie

5 Les automates à états finis du PAE

Pour bien comprendre le fonctionnement du protocole 802.1X, il est nécessaire de regarder en détail les différents automates à états finis qui régissent son fonctionnement. À titre indicatif une version simplifiée des deux principaux automates du standard 802.1X est donnée ci-après. Les simplifications ont consisté à supprimer une grande partie des variables d'état et certaines transitions temporelles de bouclage d'un état sur lui même pour ne laisser apparaître que la séquence principale qui va de l'état « INITIALISATION » à l'état « AUTHENTIFIÉ ».

5.1 PAE du système authentificateur

5.1.1 Schéma simplifié de l'automate à états finis

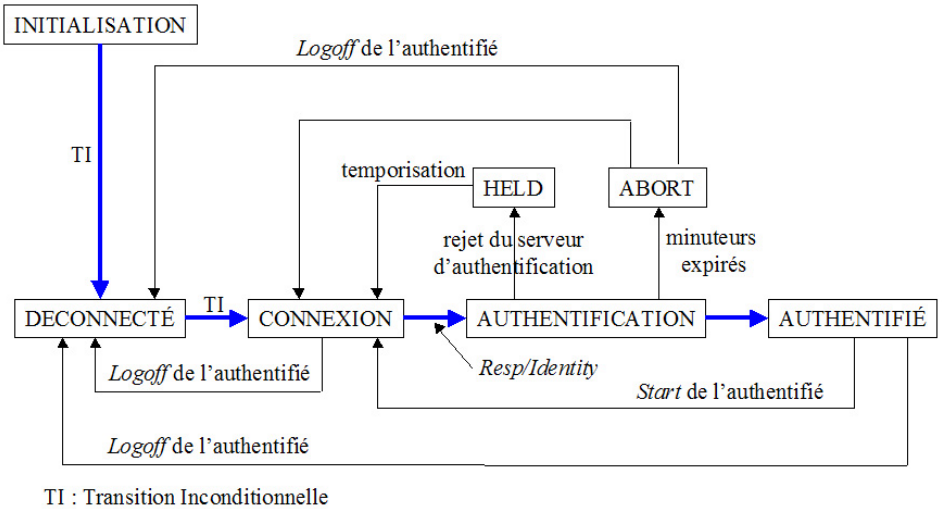

Figure 6 : Automate à états finis du système authentificateur

5.1.2 Définition des états de l'automate à états finis

INITIALISATION : cet état est atteint quand le protocole 802.1X est activé (par exemple suite à une opération de gestion) ou quand le port physique du PAE a été débranché, ou enfin lorsque l'équipement est mis sous tension.

DÉCONNECTÉ : dans cet état, le port physique est actif (par exemple si l'équipement à authentifier est branché), mais les services contrôlés par le port protégé sont inaccessibles.

CONNEXION : l'automate de l'authentificateur envoie une requête EAP avec un type *Identify* au système connecté (équipement à authentifier) physiquement sur son port physique, puis il se met en attente d'une réponse.

AUTHENTIFICATION : dans cet état, l'automate de l'authentificateur sert de relais pour le dialogue entre le système à authentifier et le serveur d'authentification. Il interprète seulement (tout en les relayant) les réponses finales (*Reject* ou *Accept*) du serveur d'authentification pour passer dans l'état correspondant.

AUTHENTIFIÉ : les services contrôlés par le port protégé sont accessibles.

HELD : état de temporisation (60s par défaut) avant retour à l'état CONNEXION. Cette attente est une parade à une attaque de type « force brute ». Dans cet état, tous les paquets sont ignorés.

ABORT : la procédure d'authentification est interrompue (demande de ré-authetification, *Logoff*, *Start*…).

Remarque : la notion de Transition Inconditionnelle (TI) qui apparaît entre les états « INITIALISATION » et « DÉCONNECTÉ » est à considérer comme telle uniquement dans le contexte du protocole 802.1X. C'est en effet une action extérieure (par exemple le branchement sur le port physique) qui fait passer de l'état « INITIALISATION » à l'état « DÉCONNECTÉ ».

5.2 PAE du système à authentifier

5.2.1 Schéma simplifié de l'automate à états finis

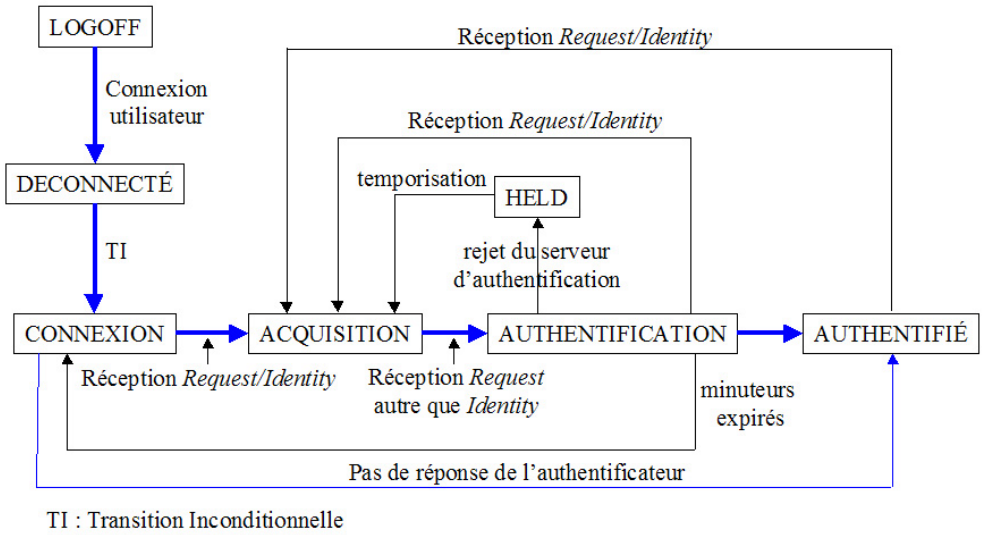

Figure 7 : automate à état finis du système à authentifier

5.2.2 Définition des états de l'automate à états finis

LOGOFF : cet état est atteint quand l'utilisateur du système quitte sa session. On remarquera que les concepteurs du standard permettent, par cette fonctionnalité, de relier l'état du port contrôlé à l'état de la session d'un utilisateur (à condition bien sûr que la connexion physique soit toujours active).

DÉCONNECTÉ : dans cet état, le port physique est actif (par exemple, l'équipement à authentifier est branché).

CONNEXION : l'automate du système à authentifier est en attente d'une requête EAP avec un type *Identify* du système authentificateur du port où il est physiquement connecté. Dès l'arrivée d'une telle requête, il passe dans l'état « ACQUISITION ». Pour permettre la compatibilité avec des équipements ne supportant pas le protocole 802.1X, le système à authentifier émet régulièrement des requêtes EAP avec un type *Start*. Au bout de 3 (valeur par défaut) non réponses, il passe dans l'état « AUTHENTIFIÉ » (considérant qu'il n'y a pas de PAE authentificateur sur ce port).

ACQUISITION : l'automate du système à authentifier envoie son identité au système authentificateur, et passe dans l'état « AUTHENTIFICATION » dès réception du premier paquet EAP avec un type différent de *Identify*.

AUTHENTIFICATION : dans cet état, l'automate du système à authentifier répond aux requêtes du serveur d'authentification qui lui sont transmises par la système authentificateur.

AUTHENTIFIÉ : les services contrôlés par le port protégé sont accessibles.

HELD : état de temporisation (60s par défaut) avant passage à l'état « CONNEXION ». Cette attente est une parade à une attaque de type « force brute ». Dans cet état la réception d'une requête EAP avec un type *Identify* provoque un passage à l'état « ACQUISITION ».

6 Les faiblesses de 802.1X

La principale faiblesse de 802.1X vient de ce qu'il a été conçu au départ dans un contexte de connexion physique (type accès PPP sur RTC). Rien n'empêche en effet un utilisateur d'insérer un *hub* (transparent à 802.1X) et de faire bénéficier d'autres utilisateurs de l'ouverture du port Ethernet d'un commutateur. La plupart des implémentations d'équipementiers permettent de surmonter cette difficulté en permettant de configurer un blocage du port Ethernet si l'adresse MAC du système authentifié change. Les attaques par écoute et rejeu sont aussi possibles, ainsi que le vol de session. On pourra utilement consulter l'article [5] pour une analyse des faiblesses de 802.1X. Les attaques sur 802.1X sont, de plus, facilitées dans le cas de l'Ethernet sans fil.

7 Les évolutions de 802.1X

La révision du standard 802.1X se fait par l'addendum 802.1aa, dont le dernier draft (numéro 5) a été publié en février 2003. Les principales modifications introduites concernent le non rejeu des échanges, l'authentification mutuelle, et la gestion des clés.

L'authentification mutuelle est une amélioration importante, car elle permet de résoudre le cas où le client est lui-même un fournisseur de service réseau, et a besoin d'être sûr qu'il s'adresse bien à un port 802.1X de confiance. Un exemple type concerne le cas du branchement d'un commutateur sur un autre commutateur. Cette authentification mutuelle est faite par l'enchaînement de deux phases d'authentification, où les rôles de chaque entité connectée sont échangés, chacune jouant successivement le rôle d'authentifiant puis d'authentifié.

8 Conclusion

Le principal risque dans l'usage de 802.1X, est de l'utiliser hors de son champ d'application. Il faut donc bien avoir à l'esprit que son usage est simplement de contrôler l'accès physique à un réseau local. Comparée à l'absence de contrôle d'accès, ou à un contrôle d'accès uniquement basé sur la valeur de l'adresse MAC/Ethernet, l'activation de l'authentification 802.1X est un réel progrès. Ce progrès se mesure aussi par la facilité d'ingénierie que procure le recours à un serveur d'authentification externe. Compte-tenu de la présence quasi-systématique de serveurs Radius sur un site, on peut considérer que l'activation de l'authentification 802.1X ne rajoute pas de complexité dans l'administration réseau d'un site.

9 Bibliographie

[1] IEEE Std 802.1X-2001, Part-Based network Access Control, 2001
[2] RFC2284: PPP Extensible Authentication Protocol (EAP)
[3] RFC2865 à RFC2869 (Radius)
[4] http://www.open1x.org : implémentation GPL d'un client 802.1X
[5] Mishra, A. & Arbaugh, W. (2002). An initial security analysis of the 802.1x standard. http://www.cs.umd.edu/~waa/1x.pdf
[6] http://www.cartel-securite.fr/pbiondi/scapy.html : outil de manipulation de paquets permettant de tester un port 802.1X