

Sécurité et performances des réseaux VoIP

Présentation

Les entreprises sont aujourd'hui équipées de réseaux de voix sur IP (VoIP) pour leur téléphonie. Une technologie encore jeune, et dont les performances sont difficiles à tester.

Il s'agit pourtant d'un aspect primordial de l'entreprise. Un tel réseau doit pouvoir résister aux attaques comme le déni de service.

Il est nécessaire de disposer d'outils permettant de tester tous les aspects d'un réseau VoIP. L'idéal étant un logiciel s'intégrant dans le réseau de téléphonie, et capable de produire, n'importe quelle situation critique.

En passant par un modèle, le logiciel devrait permettre de créer, exécuter et enregistrer le déroulement d'une communication VoIP.

Un générateur intelligent

Le logiciel de test doit être capable de générer et d'entretenir des appels corrects. Il doit donc pouvoir réagir en fonction des erreurs, ou des événements qui surviennent en cours de communication.

Toute communication passe par l'envoi et la réception de messages. Ces mêmes messages doivent être construits en fonction du déroulement de la communication (le contexte). Leur lecture influence ce même déroulement.

Le générateur doit également permettre d'introduire des erreurs, avec une probabilité plus ou moins élevée. Des erreurs qui peuvent toucher deux aspects :

- Le comportement de la communication
- La syntaxe des messages

Machines à états finis

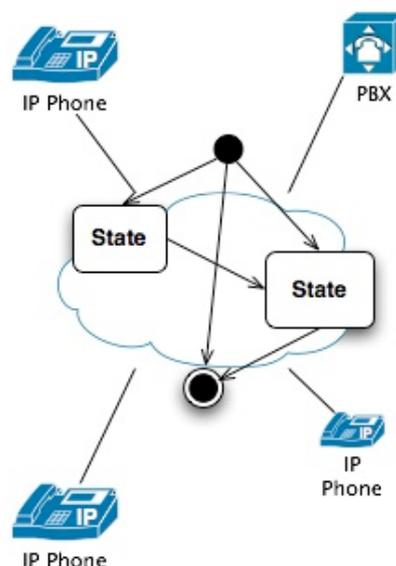
Le développement d'un logiciel de test aussi complet passe par une modélisation de la communication VoIP. Les diagrammes d'états d'UML sont une solution idéale pour une compréhension rapide du scénario.

Si une communication est une suite de messages les uns après les autres, la construction d'un message est également une suite de champs et d'octets successifs. La construction des messages peut donc se faire de la même manière que la construction des échanges : avec les machines à états.

Intérêts

L'utilisateur garde une liberté totale dans la création du scénario de test, via une utilisation simple (UML).

Des règles prédéfinies et un transformateur se chargent de créer un logiciel de test à l'image du modèle utilisateur.



Auteur: David Geretti
Répondant externe: Jürgen Ehrensberger
Prof. responsable: HEIG-VD
Sujet proposé par: HEIG-VD

Hes·SO
Haute Ecole Spécialisée
de Suisse occidentale