HAL
open science

RMI Dump

Mathieu Colonna

To cite this version:

Mathieu Colonna. RMI Dump. [Stage] 2006, pp.24. inria-00105918

HAL Id: inria-00105918
https://inria.hal.science/inria-00105918v1

Submitted on 12 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Rapport de stage 2A

RMI Dump

2 Année TRS – ESIAL Juillet – Août 2006

É ` :
Mathieu C

É ′ :
Équipe M
LORIA
Campus Scientifique - BP 239
54506 Vandoeuvre-lès-Nancy Cedex

E :
Laurent A

E :
Marc T

Avant-propos

Le stage "technicien" à un rôle important pour les étudiants de deuxième année. Il s'agit de la première réelle expérience en milieu professionnel en tant qu'informaticien. Ce stage doit permettre aux élèves-ingénieurs de découvrir et de pratiquer les techniques et outils utilisés dans les métiers de l informatique et de la production industrielle et d être confrontés aux contraintes temporelles, économiques et humaines associées. D'une durée de 6 à 10 semaines à partir de fin juin, c'est aussi une occasion pour l'étudiant d'appliquer les connaissances vues pendant deux années, ce qui lui permet de se faire une idée plus précise quant à l'usage de l'informatique en dehors de l'enceinte scolaire.

Le stage permet aussi à l'élève de faire le point sur orientation professionnelle et préciser ses choix s'ils ne sont pas encore faits. Mon expérience dans l'équipe Madynes a pleinement joué son rôle dans ce domaine. J'ai toujours été intéressé par la recherche, sans réellement savoir si je souhaitais en faire un objectif professionnel. Ces 2 mois ont été l'occasion de découvrir de monde de l'intérieur. Le sujet du stage n'était pas un réel thème de recherche, mais toutes les conversations que j'ai pu avoir avec l'équipe m'ont aidé à me décider sur mon choix de carrière.

L'équipe a su dès les premiers jours me mettre à l'aise, et a toujours été à l'écoute de mes questions. C'est donc à eux que je dois la réussite de ce stage. ce titre, je tiens donc à remercier Olivier Festor, responsable scientifique de l'équipe Madynes, pour son accueil et l'attention qu'il a su accorder à mon stage, Laurent Andrey, pour avoir su m'encadrer tout en me laissant libre de nombreux choix dans la façon de réaliser le projet. Merci aussi à Isabelle Chrisment pour ses conseils, à Josianne Reffort pour son amabilité et aux étudiants en thèse (Rémi Badonnel, Vincent Cridlig, Humberto Jorge Abdelnur) pour leurs réponses à toutes mes questions. Je remercie également tout le reste de l'équipe pour m'avoir si bien accueilli.

Sommaire

Avant-propos **2**

Introduction **4**

Choix du sujet **4**

I Présentation de l'environnement de travail **6**
I.1 Le Loria, point de convergence 6
I.2 Équipes et Projets 7
I.2.1 Organisation et Services de support à la recherche 8
I.2.2 La Recherche 9
I.3 La vie au Loria 12
I.4 L'Équipe Madynes 13

II Analyse du problème **14**
II.1 Présentation de Java RMI 14
II.2 Intérêt de RMI Dump 14

III Organisation **15**
III.1 Démarche 15
III.2 Deux réponses différentes 18
III.2.1 RMI Dump 18
III.2.2 RMI Deserializer 19

IV Résultats et Conclusion **22**

V Glossaire **23**

Références **24**

Introduction

Ce rapport présente le travail que j'ai pu effectuer du 19 Juin au 27 Août 2006 dans le cadre du stage "technicien" de deuxième année de l'ESIAL. Ce stage a été réalisé au LORIA (Laboratoire Lorrain de Recherche en Informatique et ses Applications) à Nancy. Le sujet était la conception et la réalisation de deux outils permettant la compréhension des échanges entre client-serveur dans le cadre d'une conversation Java RMI. Le protocole RMI (de l'anglais Remote Method Invocation) permet l'échange d'informations Java entre deux ordinateurs distants. Le problème est que ces échanges se font sous la forme d'un code, qui est difficilement compréhensible sans être traité. Mon travail était donc l'implantation de deux outils permettant ce décriptage.

Tout d'abord une bibliothèque Java, qui permet l'analyse de traces réseaux et l'extraction des informations RMI. L'intérêt de ce premier projet était multiple : il peut soit être utilisé seul pour un apperçu rapide des données échangées, soit incorporé dans un programme plus grand, pour une étude plus approfondie des échanges, la réalisations de statistiques ou encore de l'évaluation de performances. La seconde partie du stage a été la réalisation d'un module pour Ethereal (appelé Wireshark depuis Juin 2006). Ethereal logiciel libre d'analyse de traces, reconnu pour le grand nombre de protocoles qu'il reconnait et la visualisation graphique des informations qu'il propose.

Après un rapide historique et une présentation du Loria et de l'équipe Madynes, j'exposerai plus précisément les objectifs et le déroulement des deux projets, les dificultés rencontrées et les enseignements que j'en ai tiré.

Choix du sujet

Nous avons eu l'occasion au cours de notre deuxième année de réaliser un projet d'initiation à la recherche (PIR) en partenariat avec l'equipe ECOO (Environnements pour la COOpération) du LORIA. J'avais trouvé cette échange très enrichissant, c'est pourquoi j'ai souhaité prolonger cette expérience durant le stage de deuxième année. C'est ainsi que Isabelle Chrisment, responsable de la fillière TRS (Télécommunications, Réseaux et Services) à l'ESIAL et chargée de recherche dans l'équipe Madynes, m'a mis en contact avec Laurent Andrey.

L'intérêt de ce stage était double. D'une part le sujet m'intéressait beaucoup, j'ai toujours cherché à m'orienter dans le domaine des réseaux au travers de mes options à l'ESIAL et ce stage me permettait de rester dans cette thématique. Il mettait aussi en application de nombreuses matières enseignées comme les langages Java, C, mais aussi quelques notions de Traduction.

D'autre part celà me permettait de passer deux mois complets au sein d'une équipe de recherche, me permettant d'en savoir plus sur les parcours de chacun, les avis des étudiants en thèse, les thèmes abordés, les conditions de travail etc ... Certains membres de l'équipe étaient directement impliqués dans les enseignements de Master II Recherche de l'Université Henri Poincaré et de troisème année de l'ESIAL, ce qui m'a aidé dans le choix de mon orientation pour cette dernière année.

I Présentation de l'environnement de travail

Mon stage a eu lieu au Loria à Nancy, dans l'équipe Madynes. Après une présentation de ce laboratoire, de ses activités et des équipes qui y travaillent, j'exposerai plus en details le fonctionnement de l'équipe Madynes.

I.1 Le Loria, point de convergence

Le Loria est une Unité Mixte de Recherche (UMR 7503), c'est à dire une équipe de recherche rattachée à une université et co-habilitée avec le ministère de la Recherche et un organisme public civil de recherche français. L'unité du Loria, officialisée le 19 décembre 1997 par la signature du contrat quadriennal avec le Ministère de l'Education Nationale, de la Recherche et de la Technologie, fait intervenir cinq partenaires qui sont :

CNRS, Centre National de Recherche Scientifique Le centre est créé par décret du Président de la République Albert Lebrun, le 19 octobre 1939. Il a alors pour vocation de regrouper tous les organismes d'État, non spécialisés, de recherche fondamentale ou appliquée, et de coordonner les recherches à l'échelon national. Le CNRS s'est lancé aujourd'hui dans une politique d'actions interdisciplinaires de recherche qui permet à tout établissement d' enseignement supérieur qui le souhaite d'établir un contrat de partenariat quadriennal avec le ministère chargé de la recherche et le CNRS, dans lequel les partenaires s'engagent, pour quatre ans, sur un programme scientifique précis, un budget et des modalités de mise en œuvre. L'information et les sciences et technologies de communication ne représente qu'un des huit départements, parmis le nucléaire, la chimie, les sciences sociales, les sciences de la vie etc... Sous la direction de Arnold Migus, le CNRS emploie aujourd'hui 26 000 personnes pour un budget annuel de plus de deux milliard d'euros.

INPL, L'Institut National Polytechnique de Lorraine Fondé par décret du 23 décembre 1970, en même temps que les deux autres Instituts Nationaux Polytechniques de Grenoble et Toulouse, L'INPL est un Etablissement d'Enseignement Supérieur et de Recherche regroupant 26 laboratoires et délivrant des diplômes de 2ème cycle et de 3ème cycle. Présidé par Louis SCHUFFENECKER, l'INPL est constitué à ce jour de sept Écoles d'Ingénieurs comme l'ENSAIA, l'ENSEM, l'École Nationale Supérieure de Géologie, l'ENSIC, l'École Nationale Supérieure des Mines de Nancy, l'EEIGM et l'ENSGSI.

INRIA, Institut National de Recherche en Informatique et en Automatique Placé sous la double tutelle des ministères de la recherche et de l'industrie, l'INRIA a pour vocation d'entreprendre des recherches fondamentales et appliquées dans les domaines des sciences et technologies de l'information et de la communication. Créé le 25 août 1967 par décret sous le nom d'IRIA (Institut de recherche d'informatique et d'automatique), l'INRIA accueille aujourd'hui dans ses 6 unités de recherche situées à Rocquencourt, Rennes, Sophia Antipolis, Grenoble, Nancy et Bordeaux, Lille, Saclay et sur d'autres

sites à Paris, Marseille, Lyon et Metz, 3 600 personnes dont 2 800 scientifiques, issus d'organismes partenaires de l'INRIA (CNRS, universités, grandes écoles) qui travaillent dans plus de 138 projets de recherche communs. L'INRIA développe de nombreux partenariats avec le monde industriel et favorise le transfert et la création d'entreprises (une soixantaine) dans le domaine des STIC, se qui lui permet grâce aux contrats de recherche etaux produits de valorisation de financer environ 20% de son budget annuel de 160 millions d'euros. L'institut est actuellement présidé par Michel Cosnard.

UHP, Université Henri Poincaré L'UHP prend naissance en 1854, sous le nom de Faculté des Sciences de Nancy. L'Université de Nancy est ensuite partagée en 1970 en 3 établissements : Université de Nancy 1, Université de Nancy 2 et Institut Polytechnique de Lorraine. Nancy 1 rassemble alors Médecine, Odontologie, Pharmacie, l'ISIN, l'IUT Nancy-Brabois et la Faculté des Sciences qui sera réorganisée en 1989 en 3 UFR (Sciences et Techniques Biologiques ; Mathématiques- Informatique-Automatique ; Matière et Procédés). Ce n'est qu'en 1994 que l'Université de Nancy 1 devient par approbation de son Conseil d'Administration l'Université Henri Poincaré. LUHP regroupe à Nancy et en Lorraine, 5 facultés : Sciences et Technologies, Médecine, Pharmacie, Odontologie et Sport, 3 écoles dingénieur : ESSTIN, ENSTIB et ESIAL et 3 IUT à Nancy, Longwy et St Dié. En 2006, l'université accueillait sous la présidence de Jean-Pierre Finance plus de 18 000 étudiants, pour 2521 enseignants, enseignants-chercheurs et personnels administratifs.

NANCY 2 Université pluridisciplinaire, Nancy 2 propose des formations principalement dans les domaines suivants :
- Lettres / Langues
- Sciences Humaines et Sociales
- Droit, Economie / Finances
- Gestion / Management
- Mathématiques - Informatique
- Information - Communication / Audiovisuel

L'université a accueilli à la rentrée 2005 18 770 étudiants, répartis sur plusieurs sites, principalement le Campus Lettres, Sciences Humaines et Sociales. Nancy 2 offre, au-delà de la formation initiale, un large panel de programmes en formation continue et en enseignements en formation ouverte et à distance. Elle encourage traditionnellement les formations professionnalisantes et le développement des formations à et par la recherche. Son dynamisme dans le domaine de la recherche se retrouve dans les projets interdisciplinaires qu'elle mène en partenariat avec le CNRS.

I.2 Équipes et Projets

25 projets scientifiques ont pris place au sein du Loria, regrouper par grands thèmes de recherche. Pour soutenir ces travaux et faire vivre le laboratoire, une encadrement solide et un grand nombre de services se sont organisés autour des chercheurs.

I.2.1 Organisation et Services de support à la recherche

La Direction du Laboratoire est assistée dans ses fonctions par différentes instances. Celles ci garantissent la cohérence de la politique scientifique et le bon fonctionnement au quotidien. Voici un rapide descriptif de ses principaux éléments.

Équipe de Direction Son rôle est d'assister le directeur dans ses décisions et leur mise en oeuvre. Elle se réunit toutes les semaines, alternativement en formation plénière ou réduite aux scientifiques pour traiter de problèmes qui leur sont spécifiques.

Sous la direction depuis 2001 de Hélène Kirchner, l'équipe

- Élabore des propositions de politique scientifique (évolutions thématiques, politique internationale, formation, valorisation, recrutement,...) et suit leur mise en oeuvre en s'appuyant sur les services et des commissions spécifiques
- Réfléchit aux grandes orientations des services pour accompagner la politique scientifique de l'unité (bâtiment, budget, équipement, communication, ressources humaimes, IST, valorisation)
- Prépare les assemblées de responsables d'équipes et les comités des projets
- Représente la Direction en tant que de besoin auprès des Établissements partenaires ou des Collectivités locales
- Instruit des projets de création d'équipes ou projets communs en s'appuyant sur le Comité des Projets

Comité de Gestion Réunion des responsables des services communs à l'INRIA-Lorraine et au LORIA, le Comité de Gestion coordonne les actions internes et externes d'accompagnement de la recherche. Il assiste la Direction dans le fonctionnement administratif et technique quotidien du laboratoire et participe à son évolution. Il met en place des décisions liées notamment à l'aménagement du bâtiment, la sécurité des biens et des personnes, la répartition des ressources d'équipement et des ressources documentaires, les actions de valorisation, de communication internes et externes, et les manifestations scientifiques.

Les Services de support à la recherche assistent les chercheurs dans leur travail, et permettent le bon fonctionnement du laboratoire. En voici les principales composantes.

Les MI, Moyens Informatiques Dirigé par Bertrand Wallrich, l'équipe des Moyens Informatiques assure la gestion, l'installation et la maintenance du parc informatique du Laboratoire. Elle administre les équipements réseaux, les équipements systèmes, les postes de travail, les logiciels communs, la téléphonie et garantit l'assistance, le dépannage, ... Les MI ont pour mission d'assurer la mise à disposition, la maintenance, la sécurité et l'évolution de l'infrastructure informatique, la mise à disposition les moyens logiciels et le matériel informatique nécessaires à l'activité des équipes de recherche et des services, et enfin l'évaluation

des les solutions, l'optimisation des choix en matière d'investissements et le conseil de la Direction sur des grandes orientations.

Le CDD, Centre De Documentation Il fait vivre et développe le fonds documentaire du LORIA. C'est cette équipe qui gère les publications du laboratoire. Le Centre De Documentation est commun au LORIA et à L'INRIA Lorraine. Il met à disposition de la communauté un fonds spécialisé (sur un plan régional et national) et des ressources électroniques et gère également les demandes de documents spécifiques

Les SG, Services Généraux Ils se mettent à la disposition de l'intérêt général du Laboratoire et gèrent la maintenance des bâtiments, des équipements et des installations techniques. Cette équipe garantit la gestion du patrimoine et la maintenance de ses installations, le suivi des travaux de la 3e tranche du bâtiment, la mise en sécurité des biens et des personnes et l'accueil du LORIA.

I.2.2 La Recherche

À travers ses équipes de recherche, le LORIA possède des compétences reconnues en Sciences et Technologies de l'Information et de la Communication. Les équipes se structurent dans 6 thèmes de recherche et offrent des compétences scientifiques variées dans plusieurs domaines dapplication. Certaines équipes sont presentes dans différents thèmes en raison de leur projets interdisciplinaires.

Calculs, simulation et visualisation à haute performance La physique, la chimie ou la biologie soulèvent aujourdhui des problèmes de modélisation qui constituent de vrais défis pour linformatique : la taille des modèles numériques à calculer, transférer et visualiser, le besoin impérieux de précision géométrique et physique, la prise en compte de la composante temporelle et de linteractivité dans les techniques de simulation et de visualisation. Lalliance en Lorraine du calcul scientifique, des réseaux et de la visualisation haute performance est un atout pour aborder ces défis.

- ADAGIo : Algorithmique Discrète et ses Applications à la Génomique et à l'Imagerie
- ALICE : Géometrie et Lumière
- MAGRITE : Augmentation visuelle d'environnements complexes
- SPACES : Systèmes Polynomiaux, Arithmétiques, Calculs Efficaces et Sûrs
- VEGAS : Algorithmes géométriques effectifs pour la visibilité et les surfaces

Qualité et sûreté des logiciels Les problèmes de fiabilité et de sécurité des systèmes informatiques sont abordés en se focalisant sur deux objectifs : la conception de logiciels sûrs et permettant le traitement dapplications de taille réelle et la vérification des systèmes et services critiques, embarqués ou enfouis. Les équipes qui la compose sont

- CASSIS : Combinaison d'Approches pour la Sécurité des Systèmes InfiniS (sécurité des protocoles et vérification)
- DEDALE : Développement de spécifications (interface entre les méthodes semi-formelles et formelles)
- ECOO : Environnements pour la COOpération (travail coopératif)
- MACSI : Modélisation, Analyse et Conduite des Systèmes Industriels (étude des systèmes à événements discrets)
- MOSEL : Méthodes formelles et applications (spécification formelle et ve´rification)
- PROTHEO : Contraintes, déduction automatiques et preuves de propriétés de logiciels (vérification)
- TRIO : Temps Réel et InterOpérabilité (temps réels et systèmes embarqués)
- TYPES : Logique, Théorie de la Démonstration et Programmation (démonstration automatique)

Systèmes parallèles, distribués et communicants Calculer sur des plates-formes hétérogènes distribuées, maîtriser le fonctionnement des réseaux hétérogènes et complexes, nécessite des recherches et des développements autour de modèles et d'architectures logiciels pour la surveillance des services à forte valeur ajoutée, darchitectures dapplications, entre autres pour la grille, les entreprises virtuelles ou lintelligence ambiante. Le travail de cette équipe porte sur la supervision, la coopération, et la distribution des programmes et des services, mais aussi sur létude de protocoles, la sûreté de fonctionnement, et lévaluation de performances. Cette thématique regroupe les équipes

- ALGORILLE : Algorithmes pour la Grille
- CORTEX : Intelligence neuromimétique
- ECOO : Environnements pour la COOpération
- MACSI : Modélisation, Analyse et Conduite des Systèmes Industriels
- MAIA : MAchine Intelligente Autonome
- MADYNES : Supervision des Réseaux et des Services Dynamiques (équipe dans laquelle j'ai effectué mon stage et qui sera plus longuement décrite par la suite)
- MOSEL : Méthodes formelles et applications
- TRIO : Temps Réel et InterOpérabilité

Modèles et algorithmes pour les sciences du vivant En bioinformatique, un des objectifs de recherche est dexpliquer comment les composants dun système biologique interagissent pour assurer une certaine fonction ; un modèle informatique permet danalyser, de simuler et à terme de prédire le comportement dun système biologique ainsi que de raisonner avec des méthodes formelles sur les propriétés de ce système. Un autre objectif est de mieux comprendre les mécanismes dexpression de génomes à partir de leurs séquences nucléotidiques ; cela nécessite la mise au point de nouvelles méthodes algorithmiques efficaces, tenant compte dune prolifération de données génomiques et de nouvelles connaissances biologiques. Linformatique bio-inspirée sappuie sur la conception de systèmes multi-agents, de

systèmes neuromimétiques continus ou à évènements discrets, de systèmes connexionistes de très grande taille. Lobjectif est détudier la capacité dapprentissage de ces systèmes bio-inspirés, grâce à des structures et algorithmes performants alliant méthodes mathématiques et connaissances biologiques.

- ADAGIo : Algorithmique Discrète et ses Applications à la Génomique et à l'Imagerie
- CORTEX : Intelligence neuromimétique
- MAIA : MAchine Intelligente Autonome
- MODBIO : MODèles informatiques en BIOlogie moléculaire
- ORPAILLEUR : Extraction et représentation de connaissances
- PROTHEO : Contraintes, déduction automatiques et preuves de propriétés de logiciels

Traitement de la langue naturelle et communication multimodale Le LORIA rassemble des compétences en traitement automatique de la parole, en informatique linguistique, en interaction homme-machine, en vision, en analyse de documents, en simulation de processus perceptifs et communicatifs. Cette convergence permet de sattaquer à linterprétation et à la présentation de données multimodales, avec pour objectif de modéliser la perception et la cognition humaines et den tirer parti pour mieux communiquer et interagir avec lutilisateur.

- CALLIGRAMME : Logique Linéaire, Réseaux de Démonstration et Grammaires Catégorielles
- CORTEX : Intelligence neuromimétique
- LANGUE ET DIALOGUE : Informatique linguistique pour le dialogue homme-machine multimodal
- MAIA : MAchine Intelligente Autonome
- MERLIN : Méthodes pour l'Ergonomie des Logiciels Interactifs
- PAROLE : Analyse, Percception et Reconnaissance automatique de la parole
- QGAR : Navigation dans les documents graphiques par l'analyse et la reconnaissance
- READ : Reconnaissance de l'écriture Et Analyse de Documents

Représentation et gestion des connaissances Si les capacités de stockage et dorganisation des systèmes dinformation permettent denvisager des masses de données gigantesques, encore faut-il se doter des moyens de les exploiter et de les partager, au travers doutils dindexation, de navigation, de recherche dinformations, de coopération. Cette notion soulève des problématiques nombreuses, liées au caractère très hétérogène des documents manipulés et aux critères dorganisation des données qui peuvent être très variables suivant les organisations concernées. La représentation et la gestion des connaissances sattaquent à ces problèmes complexes, dans le cadre du Web sémantique et des entreprises virtuelles.

- ADAGE : Algorithmique Discrète et ses Applications à la Génomique et à l'Imagerie
- CORTEX : Intelligence neuromimétique
- ECOO : Environnements pour la COOpération
- LANGUE ET DIALOGUE : Informatique linguistique pour le dialogue homme-machine multimodal

- ORPAILLEUR : Extraction et représentation de connaissances
- QGAR : Navigation dans les documents graphiques par l'analyse et la reconnaissance
- READ : Reconnaissance de l'écriture Et Analyse de Documents
- SITE : Modélisation et Développement de Systèmes d'Intelligence Économique

I.3 La vie au Loria

Ambiance de travail Le Loria est vraiment une structure agréable, en évolution régulière depuis sa création. Cela se traduit par exemple par l'ouverture de la troisième aile du batiment, ouvrant de nouveaux bureaux, des salles de conférence, un refectoire et une cafeteria, ce qui permet au laboratoire d'accueillir désormais 500 personnes. Tous les locaux sont ponctués d'espace-détente avec distributeurs et machines à café lorsque celles ne sont pas directement dans les bureaux. Ces espaces de vie commune favorisent aussi les rencontres entre membres d'équipes différentes. Les employés du laboratoire sont aussi fréquemment informés par mail des activités importantes des autres équipes, la tenue de conférence, l'arrivée de nouveau personnel etc... La fete du loria renforce aussi cette esprit de collectivité avec la tenue chaque année au mois de Juin d'un repas réunissant toutes les composantes du Loria.
Les heures de travail ne sont pas précisément comptées, mais un créneau horaire de présence commune est demandé, pour permettre aux équipes de se voir au complet. Ensuite chacun est libre de venir avant ou de rester plus tard. Les locaux sont ainsi occupés de 8h00 à tard dans la nuit, selon les préférences et les disponibilités de chacun. On pourrait penser que cette tolérence serait abusée et les batiments désertés mais en fait pas du tout, les gens sont "libérés" de la montre, et sont présent en fonction du travail qu'ils ont a fournir et c'est alors tout naturel de prlonger les journées en cas de besoin.

Un brassage scientifique Ce qui m'a beaucoup plu dans mon expérience au Loria, c'est l'apport continu de connaissance qui s'y trouvait. Tout d'abord les gens que je cotoyais dans l'équipe qui m'ont beaucoup appris, que ce soit en m'aidant sur mon projet, ou en discutant des leurs, de leur parcours, les sujets de thèse, leur vision du domaine de la recherche, etc... C'est précisément l'expérience que je recherchais : me faire une idée précise de la recherche en étant directement en contact avec les gens qui en étaient le moteur.
D'autres étudiants de l'ESIAL étaient aussi en satge au Loria, et nous nous retrouvions fréquemment autour d'un café ou au moment des repas. C'était alors l'occasion de parler de ce que faisait chacun, le travail de son équipe etc. Là encore celà permettait de se faire une idée globale des différents thèmes de recherche du Loria et lever le voile sur des domaines encore flous.
Des conférences internationnales prennent place au Loria, et nous avons eu la chance d'assister à l'une d'entre elles : RNC 7 (7th Conference on Real Numbers and Computers). Cette conférence nous a permis là encore de ne pas se restreindre uniquement à nos stages, mais de découvrir des sujets dont nous aurions ignoré l'existence autrement.

I.4 L'Équipe Madynes

L'équipe Madynes dans laquelle j'ai effectué mon stage a pour thème de recherche la Supervision des Réseaux et des Services Dynamiques sous la direction d'Olivier Festor.

L'équipe de recherche MADYNES vise la conception, la validation et la mise en oeuvre de nouveaux paradigmes et architectures de supervision et de contrôle capables de maîtriser la dynamicité croissante des services et résistantes au facteur d'échelle induit par l'Internet ubiquitaire. Axes thématiques

Les axes thématiques de l'équipe sont :

- élaboration de méthodes d'auto-organisation des entités de gestion,
- conception, évaluation et mise en oeuvre d'architectures de supervision exploitant le modèle pair-à-pair, le routage applicatif, et de nouvelles approches pour la représentation de l'information de gestion,
- modélisation et benchmarking des infrastructures de supervision.
- sécurité : nouveaux protocoles de distribution de clefs et infrastructures pour le respect de l'anonymat et de la vie privée,
- la configuration et la provision de services,
- la mesure, l'analyse et l'instrumentation automatique des services.

L'Internet nouvelle génération est le domaine d'application principal des résultats des axes précédents. Son architecture ainsi que les services qui s'y déploient offrent toutes les caractéristiques de dynamicité et de besoin de passage à l'échelle que Madynes aborde dans les autres axes du projet.

L'équipe travaille en relation avec d'autres partenaires scientifiques et industriels tels que :

- Partenaires académiques : LAAS-CNRS, LIP6, ENST, INSA de Lyon, LSR-IMAG, Twente University, Concordia University, UQAM, Macquarie University
- Partenaires industriels : 6Wind, Alcatel, France Telecom R&D, IBM et Thalès
- Ces coopérations sont renforcées par notre participation à de nombreux programmes nationaux (RNRT, RNTL, ACI, AS CNRS) et internationaux (projets 6Net, EUNICE) ainsi qu'à des groupes de recherche et de standardisation comme le NMRG à l'IRTF

II Analyse du problème

Cette partie est consacrée à la présentation détaillée du sujet de stage. Il sera mis en avant l'intérêt de l'outil développé, le conexte technologique et les fonctionnalités recherchées.

II.1 Présentation de Java RMI

Java RMI (pour Remote Method Invocation) est une interface de programmation (API) pour le langage Java qui permet d'appeler des objets distants. Cette bibliothèque qui se trouve en standard dans Java J2SE, est une technologie qui permet la communication via le protocole HTTP entre des objets Java éloignés physiquement les uns des autres, autrement dit s'exécutant sur des machines virtuelles java distinctes. RMI est relativement simple d'utilisation et facilite le développement des applications distribuées en masquant au développeur la communication client / serveur. Les données réelles échangées entre les ordinateurs clients et serveurs ne sont donc pas directement maitrisable par l'utilisateur.
Chaque objet codé par un serveur doit être enregistré dans un registre, appelé RMI Registry. Dans ce registre, on associe le nom de l'objet à sa localisation sur le serveur (le serveur et le RMI Registry peuvent être sur des ordinateurs distant). Á chaque demande d'un objet, le client interroge tout d'abord le registre. Celui-ci puise dans sa liste d'objets référencés et redirige le client à l'emplacement de l'objet sur le bon serveur. Le programmeur coté serveur peut à tout moment effectuer des modifications tant que l'objet est toujours correctement référencé dans le registre. Ces modifications sont complétements invisibles du point de vue du client.
La récupération et l'analyse des données brutes échangées n'est pas possible à l'origine. RMI se charge lui même du codage/décodage et envoi/reception des informations. Toutes ces opérations sont transparentes pour l'utilisteur ce qui présente des avantages mais aussi des inconvéniants.

II.2 Intérêt de RMI Dump

Le but de ce stage était la création d'un outil permettant la récupération et l'analyse des données échangées entre client, registre et serveur. Ces informations peuvent être très utiles pour mieux comprendre comment l'information est véhiculée, surveiller les échanges, faire de l'évaluation de performances ou identifier des problèmes dans un code. L'idée était de répondre à cette demande par deux outils complémentaires.

RMI Dump Il s'agit d'une application java, qui à partir d'une capture réseau réalisée pendant des échanges clients serveur, permet de filtrer les paquets RMI et d'afficher dans une fenetre de commande la nature des informations échangées (appel de fonction, retour de donnée etc). RMI Dump reprend le fonctionnement de TCP Dump, mais spécialisé aux paquets Java RMI. TCP Dump ne permet que d'intercepter et enregistrer des traces réseaux, il

ne permet pas une identification précise des données échangées et l'extraction des donncées contenues dans les paquets. RMI Dump peut aussi être utilisé sous la forme d'une bibliothèque pour être incorporé dans une application de surveillance ou d'analyse des échanges.

RMI Deserializer La seconde partie du stage s'est portée sur le développement d'une extension (ou plug-in) du logiciel Wireshark (anciennement Ethereal). Wireshark est tout comme TCP Dump un logiciel libre d'analyse de protocole, ou « packet sniffer », utilisé dans le dépannage et l'analyse de réseaux informatiques, à la différence près qu'il dispose d'une interface graphique et propose bien plus de service que son aîné. Wireshark reconnait 759 protocoles différents, mais ses auteurs ont prévu des outils permettant l'ajout d'extensions pour reconnaître de nouveaux protocoles ou améliorer la reconnaissance d'anciens. C'est le cas pour le plug-in que nous avons développé : Java RMI était déjà reconnu par Wireshark, mais il ne donnait que des informations d'ordre général (appel d'un objet, accusé de réception, retour d'erreur ...). Le nom des objets appelés ne figurait pas, et seules les communications entre le client et le registre étaient visibles. Le dialogue avec le serveur est fait sur un port différent, communiqué au client par le registre. Toutes ces données apparaissaient donc comme "non-identifiées".

III Organisation

III.1 Démarche

Il s'agit de la première application qui a été développée durant le stage. Entièrement en Java, elle permet d'extraire les données RMI échangées entre le client, le registre et le serveur depuis une trace réseau. Elle ne nécessite à son execution rien d'autre qu'une machine virtuelle Java classique. RMI Dump prend en charge les traces enregistrées sous le format Libpcap qui est le plus couramment utilisé. La compatibilité avec d'autres formats est tout à fait possible si on connait de façon précise la structure de l'enregistrement. Cette fonctionnalité a été prévue dès le début du projet.
Dans un premier temps, un gros travail de recherche d'information a été fait sur le format Libpcap et sur Java RMI.

Libpcap Pcap est une interface de programmation pour la capture de paquets. L'implantation de pcap pour les systèmes Unix est Libpcap. L'équivalent pour Windows s'appelle WinPcap. Lipcap peut être utilisé dans un programme pour la capture de paquets qui sont véhiculés sur un réseau, et dans les versions récentes pour envoyer des paquets sur un réseau ou pour obtenir automatiquement une liste des interfaces réseaux utilisables par libpcap. Libpcap est l'élément central de capture et de filtrage de paquets de nombreuses applications libre-source ou commerciales, telles que des analyseurs de protocole, des moniteurs réseau, des systèmes de detection d'intrusion, des générateurs de traffic ou des programmes de test

de réseaux. Il est utilisé par exemple dans des logiciels tels que Tcp Dump, Wireshark, Nmap. Le format de sauvegarde est un format très simple. Celui-ci est fixé depuis la version 0.4 d'Ethereal (1998), ce qui permet d'assurer la pérénité de toutes les applications qui en dépendent (dont par extension RMI Dump et RMI Deserializer). Un fichier de sauvegarde est constitué de la façon suivante :

- Entête globale : dans laquelle on trouve des informations générale sur la capture comme la version de Libpcap utilisée à l'enregistrement, la longueur maximum des paquets, l'interface sur laquelle la capture a été réalisée. La longueur de cette entête est fixe de 24 octets.
- Entête du paquet 1 : contient des informations sur la paquet 1, comme le temps en seconde et microsecondes où la capture a été réalisée ou la longueur du paquet en octets. Cette entête possède elle aussi une longueur fixe, de 16 octets.
- Données du paquet 1 : les données effectivements transmises suivent directement l'entête, sur une longueur définie par l'entête prédente.
- Entête du paquet 2
- Données du paquet 2
- etc...

Ce schéma se répète jusqu'à la fin de l'enregistrement. Le problème avec ce type d'enregistrement est que pour accéder au paquet i, il faut parcourir toutes les entêtes des paquets précédents, lire la taille du paquet et sauter au suivant, ce qui représente un processus assez lent.

Grammaire RMI Les échanges RMI respectent une grammaire. Une information RMI est un Stream. Il est ensuite spécifié suivant le message à véhiculer. Un stream peut être réparti sur plusieurs paquets. Voici un extrait de la grammaire :

```
Stream:
    Out
    In

Out:
    Header Messages
    HttpMessage

Header:
    0x4a 0x52 0x4d 0x49 Version Protocol

Version:
    0x00 0x01

Protocol:
    StreamProtocol
    SingleOpProtocol
    MultiplexProtocol

Messages:
```

```
21     Message
22     Messages Message
```

L'objectif était donc le parcours de cette grammaire pour identifier le type de message échangé. "In" et "Out" définissent si le paquet est entrant ou sortant (du point de vue du client), et "Message" contient les données utiles (adresse d'un objet sur un erveur, structure d'un objet, retour d'une exception ...) sous forme sérialisée. La grammaire RMI est relativement simple et il n'y a à aucun moment d'ambigüité.

La sérialisation Java Là aussi il s'agit d'une grammaire, plus complexe que la précédente car elle doit permettre de sauvegarder tout type d'objets (structure, champs, fonctions, héritages...). La sérialisation Java recherche les buts suivants :

- Être compacte et structurée pour une lecture efficace.
- Ne pas reécrire des éléments précédemment spécifiés, utiliser la connaissance des héritages pour ne pas avoir à tout écrire.
- Être suffisant pour reconstituer n'importe quoi.

Dans les faits, la documentation fournie par Sun Microsystem, propriétaire du langage Java, s'est révélée incomplète, ambigüe voire fausse sur certains points ce qui a présenté quelques problèmes et des erreurs difficiles à identifier. Voici un extrait de la grammaire en question :

```
1  stream:
2    magic version contents
3   contents:
4    content
5    contents content
6   content:
7    object
8    blockdata
9   object:
10   newObject
11   newClass
12   newArray
13   newString
14   newEnum
15   newClassDesc
16   prevObject
17   nullReference
18   exception
19   TC_RESET
20  newClass:
21   TC_CLASS classDesc newHandle
22  classDesc:
23   newClassDesc
24   nullReference
25   (ClassDesc)prevObject
```

Une fois ce travail d'information réalisé, nous avons ensuite débuter la conception des deux logiciels.

III.2 Deux réponses différentes

III.2.1 RMI Dump

Rédigé entièrement en Java, il s'agit du premier outils qui a été développé. Plus simple que le suivant, il m'a permis de me familiariser avec Java RMI, la sérialisation Java et le format Pcap. Nous avons commencé par créer une batterie de fonctions permettant la lecture et le découpage par paquet des fichiers de sauvegarde.

Lecture des fichiers Pcap Tout d'abord le type de fichier de sauvegarde est reconnu par quelques tests effectués sur le debut du fichier. Une première lecture est alors faite selon le format de sauvegarde détecté. Il est possible d'ajouter de nouveaux format, il suffit qu'ils implantent certaines fonctions qui doivent être communes et qu'ils soit référencés dans le fichier principal de lecture. Cette première lecture permet d'obtenir les informations générales sur la capture, et de séparer les paquets pour un accés direct lors des appels suivants. Que le paquet soit de type RMI ou non, de nombreux tests et fonctions sont alors disponibles pour récupérer des informations sur les paquets, comme extraire les adresses MAC source et destination, etc..). Le paquet est analysé pour connaître un minimum d'informations et comprendre les échanges en cours (Á quel moment le paquet a-t-il été enregistré ? Quel est le type de paquet ? TCP, UDP, autre ? S'agit-il d'un accusé de réception ? etc ...).
C'est aussi à cette étape que des tests sont effectués pour savoir si le paquet contient des informations RMI. Pour qu'un paquet soit retenu comme étant de type RMI, il doit s'agir d'un envoie de données, et il doit transiter par un port connu pour les échanges de données RMI (celui par défaut proposé par Sun, ou spécifié manuellement par l'utilisateur). Lorsqu'un paquet RMI est détecté, il est marqué comme tel, là encore pour une identification plus rapide par la suite.
Durant cette étape, une bonne connaissance de la structure des fichiers Pcap et un retour sur les cours de Réseaux de deuxième année ont été nécessaire, car la lecture des fichiers de sauvegarde se faisait octet par octet. Une révision de certaines normes a donc été utile (taille des entêtes ethernet, IP, dans quel octet se trouve quelle information etc...).

Analyse RMI Dans le cas d'un paquet RMI, on se réfère à la grammaire RMI proposée par Sun. Après avoir tiré les données utiles du paquet grâce aux fonctions créées pour la lecture du paquet, on détermine tout d'abord s'il s'agit d'une demande, d'un retour d'information, d'un accusé de réception etc...). Pour chacun de ses types une analyse plus poussée est effectuée, ce qui permet par exemple de connaître l'objet appelé dans le cas d'un appel.
Lors d'un appel sur le registre, celui-ci communique l'adresse du serveur possédant l'objet au client, qui répond en guise d'accusé de réception l'adresse et le port depuis lesquels il va poursuivre la conversation. C'est précisément cet échange qui permet à RMI Dump de poursuivre le déchiffrage des informations. Si cette communication n'est pas interceptée, les échanges suivants ne seront pas marqués comme étant de type RMI s'ils ne sont pas faits sur le port par défaut. Ils ne seront donc pas décodés. Si la communication est prise en

"cours de route" et que l'utilisateur connait le port d'échange, il peut toujours le spécifier manuellement pour que l'analyse se fasse.
Là encore le programme est assez souple puisqu'il permet l'ajoût d'autres analyseurs s'ils respectent un certain format et qu'ils sont déclarés dans le fichier principal d'analyse.

Tests et mise au point Les tests ont débuté sur une capture "basique" qui présentait un appel du client au registre, une réponse du registre, un échange client/serveur et un retour de la structure de l'objet demandé (elle aussi assez simple dans un premier temps). Cet exemple nous a servi de fil conducteur, nous permettant de nous ajuster par rapport à la documentation fournie qui n'était pas précise ou toujours exacte.

III.2.2 RMI Deserializer

Le développement d'une extension pour Wireshark/Ethereal m'a occupé plus longument. La principale difficulté était de désérialiser des données Java depuis un programme rédigé en langage C.

Création d'un plug-in Wireshark Wireshark est compatible Windows et Unix, donc pour s'assurer que le module développé le soit aussi, il doit respecter certaines contraintes. Les développeurs de Wireshark ont choisi pour celà la librairie GLib.

GLib est une librairie multi-plateforme (c'est-à-dire compatible Windows et Unix). Elle a vu le jour avec le projet GTK+, mais elle est désormais utlisée dans d'autres applications. Á l'origine, il s'agissait d'une librairie pratique pour recueillir des fonctions de bas-niveau, mais avec son développement, il contient maintenant des méthodes qui fonctionnent différemment selon les plateformes. GLib contient une allocation mémoire qui lui est propre, un système d'objets et de types, des outils pour la manipulation de chaînes de caractères ainsi que des structures de données et les foncions associées comme des chaînes de caractères de longueur dynamique, des tableaux de longueur dynamique, des arbres binaires ou Naires, et une redéfinition des types basiques (comme les entiers signés ou non, codés de 1 à 8 octets). Toujours dans un soucis de compatibilité, les règles de compilation du programme étaient très strictes (de type ANSI Pedantic). Par exemple on ne peut déclarer des variables qu'en début de fichier ou de fontcion. Le plug-in était donc rédigé en C, mais il n'avait plus beaucoup de points communs avec celui déjà pratiqué.

Une extension Wireshark doit aussi respecter certaines règles précises pour être pris en compte à l'exécution, comme l'appel de certaines fonction d'enregistrement pour être référencé comme plug-in, donner quelques précisions sur le rôle du plug-in, le protocol qu'il reconnait, le ou les ports concernés etc... Pour la compilation du logiciel, le nom du plug-in et les fichiers qu'il contient doivent aussi figurer dans les fichiers de construction automatique ;

c'est à dire les fichiers Makefile pour Unix et Windows.

Etat des lieux RMI Deserializer reprend le développement commencé par Michael Stiller, qui avait posé les bases d'un analyseur RMI pour Ethereal. La version qu'il propose (et qui est actuellement fournie avec Ethereal) se contente de reconnaître la grammaire RMI, mais pas de désérialiser les données Java. Une fois les règles de codage un peu complexes mises de côté, il s'avère que Wireshark est totalement configurable pour ce qui est de la reconnaissance de protocol (les possibilités offerts sont vraiment impressionnantes, et je n'ai pas pu tout explorer durant le mois qu'il me restait pour réaliser le second projet. Il est possible de réaliser des statistiques sur des captures, ce qui aurait pu être très intéressant, et c'est sans doute une piste à suivre si le développement de cet outils doit être poursuivi).

Développement et difficultés rencontrées Plusieurs problèmes sont apparus avec l'avancée du projet. Tout d'abord désérialiser des données Java dans un programme C n'est pas simple. L'expérience a dans doute déjà été tentée, mais je n'ai rien trouvé qui puisse m'aider dans cette initiative.
Après avoir adapté le plug-in déja existant et récupéré les données sérialisées, nous avons commencé par créer un automate qui parcours la grammaire.
Cette partie est assez longue car la documentation fournie n'était pas assez précise sur certains points. Il a fallu regarder dans les sources de Java et parfois regarder des exemples de fichiers sérialisés à l'aide d'un éditeur hexadécimal pour comprendre vraiment le fonctionnement de la sérialisation Java. Chaque fonction doit lire des informations sur un nombre variable d'octets, ce qui permet de passer à une fonction suivante ou de terminer l'analyse s'il s'agissait de la dernière information. Ce qui a posé un deuxième problème, en rapport avec le fonctionnement de Wireshark.
L'avantage de Wireshark est de proposer à l'utilisateur une interface sur laquelle il peut visualiser tous les paquets enregistrés, leur contenu en hexadécimal. En cliquant sur chaque octet, le logiciel précise l'information qu'il code. Pour celà, dans le plug-in, il faut définir à l'avance et de façon très précise quel octet dans le paquet contient quelle information, dans le cas d'une information sur plusieurs octets sur quelle longueur doit-on lire etc... C'était la toute la difficulté, car il faut callibrer tous les paquets d'information avant de les "envoyer" à Wireshark pour l'affichage. Pour schématiser ce problème, prenons le cas d'un groupe d'octets codant la déclaration d'une chaine de caractères. Dans l'exemple suivant, supposons que chaque lettre représente un octet. Un 'A' est un octet codant la déclération d'une chaine, les octets 'B' codent un entier, et les octets 'C' sont des caractères. On pourrait donc voir ...

```
74 00 05 H E L L O
A  B  B  C C C C C
```

... avec A le marqueur codant la déclaration d'une nouvelle chaîne, 'B B' pour coder sa longueur, et autant de C que de caractères dans la chaîne. Au final, la déclaration de la chaî,e a

pris ici 8 octets. Mais on ne peut pas le deviner dès la lecture du caractère 'A'. Avant d'afficher édéclaration d'une chaîne", il faut donc avoir lu au minimum les octets B. Nous avons donc décidé de faire deux lectures de des données sérialisées, ce qui prend effectivement du temps, mais obligatoire.

La deuxième difficulté que nous avons rencontré est la simplification dans le cas de répétitions. En effet, depuis le début de la trasmission, chaque élément, lorsqu'il est déclaré pour le première fois est implicitement numéroté des cotés serveur et client. Lorsque Java doit sérialiser à nouveau cette donnée, il envoie seulement ce numéro d'identification. Lors de l'enregistrement de l'échange, si la communication est prise "en cours de route", il manque donc les données de début. Et même dans le cas où toutes les données ont été récoltées, il est difficile de faire le lien entre des données de paquets différents. Wireshark est prévu pour analyser le paquets un à un.

L'envoi depuis le registre de l'adresse d'un objet au client se fait selon un schéma bien particulier que nous avons réussi à identifier et filtrer. De cette adresse (ou Endpoint) est alors extrait le numéro de port sur lequel la communication va se poursuivre. Ce numéro est alors ajouté temporairement à la liste de ports associés au protocole RMI. Là encore, si la communication a été prise en cours de route, l'ajout automatique de ce genre de numéros de port ne se fera pas. C'est à l'utilisateur de regarder les premiers paquets et le port par lequel ils transitent pour l'ajouter manuellement s'il estime suivre le protocole RMI.

IV Résultats et Conclusion

Le stage se termine et les deux outils fonctionnent sur tous les tests que nous avons créés. Les données échangées sont reconnues, ce qui permet à toute personne renseignée de diagnostiquer les erreurs dans le fonctionnement de son application ou l'améliorer en allégeant les transfert par exemple. Ces outils sont perfectibles et leur développement a été pensé dans ce sens. Le code est donc commenté, les noms de variables et de fonctions sont explicites, et l'architectures des programmes a aussi été faite dans ce sens.
Pour RMI Dump, on pourrait par exemple rajouter des formats de sauvegardes (seul le format pcap est reconnu pour le moment) ou l'associer avec le logiciel Jpcap, qui permet de réaliser des captures sans passer par TCP Dump ou Wireshark. Cela permettrait aussi de réaliser un mode "en direct", et non plus "post-mortem" pour visualiser le traffic.
Pour RMI Deserializer, on peut sans doute encore affiner la façon dont les données sont affichées. Actuellement toutes les informations sont présentes, mais sont encapsulées les unes dans les autres. On pourrait peut-être réorganiser certains éléments pour un affichage plus direct des informations utiles. Wireshark dispose aussi d'un système de statistiques qui pourrait être utilisé pour évaluer la fréquence de certains messages, etc...

Ces deux mois ont été a la fois plaisants et instructifs. J'ai beaucoup appris grâce au projet et aux travaux que l'équipe réalisait en parallèle sur les réseaux, les domaines avoisinants et les perspectives futures. Cotoyer d'autres équipes est très intéressant, ce qui permet d'élargir son domaine et d'avoir une bonne culture générale du domaine de l'informatique et des travaux qui s'y font actuellement. C'est souvent avec d'autres stagiaires que j'ai pu discuter, donc je ne connais les acitivités de toutes les équipes du laboratoire, mais ils ont réussi à chaque fois à suciter ma curiosité.
Ce stage m'a permis aussi de faire le point sur mon orientation, et de choisir en toute connaissance de cause de ne pas me diriger vers la voie de la recherche dans un futur proche. Cet environnement est très intéressant et je m'y redigerais peut-être par la suite, mais pas pour le momentje ne me sens pas pret à faire une thèse dans le but de devenir chercheur dans le domaine public.

V Glossaire

Ethereal : ancien nom de Wireshark (voir Wireshark). Ethereal a été renommé en juin 2006 pour des raisons de copyright, le développeur d'ethereal/wireshark n'ayant pu trouver d'accord avec son ancien employeur

Paquet : en réseau, le paquet est une unité de transmission utilisée pour communiquer. Afin de transmettre un message d'une machine à une autre sur un réseau, celui-ci est découpé en plusieurs paquets transmis séparément.

Plug-in : le terme anglais plugin (ou plug-in, du verbe to plug in qui signifie brancher), est employé pour désigner un programme qui interagit avec un logiciel principal pour lui apporter de nouvelles fonctionnalités.

RMI (Java) : RMI (pour Remote method invocation) est une interface de programmation (API) pour le langage Java qui permet d'appeler des objets distants.

Sérialisation : la sérialisation est un processus visant à encoder l'état d'un objet qui est en mémoire sous la forme d'une chaîne d'octets. Cette chaîne d'octet pourra par exemple être utilisée pour la sauvegarde sur disque ou le transport sur le réseau. L'activité inverse, visant à décoder la suite d'octets pour créer une copie conforme des objets d'origine, s'appelle la désérialisation.

Traces : enregistrement d'un échange de paquets.

Wireshark : (anciennement Ethereal) c'est un logiciel libre d'analyse de protocole, ou « packet sniffer », utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro-ingénierie, mais aussi le piratage. Wireshark reconnait actuellement759 protocoles différents.

Références

[1] Sources Java : http://www.sun.com/software/communitysource/j2se/java2/download.xml

[2] Protocol de transport RMI : http://java.sun.com/j2se/1.5.0/docs/guide/rmi/spec/rmi-protocol3.html

[3] tutoriel RMI : http://java.sun.com/docs/books/tutorial/index.html

[4] Documentation développeur pour Wireshark : http://netmirror.org/mirror/ftp.ethereal.com/docs/developer-guide-a4.pdf

[5] Protocol de sérialisation Java : http://java.sun.com/j2se/1.5.0/docs/guide/serialization/spec/protocol.html

[6] Spécification du format pcap : http://wiki.ethereal.com/Development/LibpcapFileFormat?action=show&redirect=Development+2fLibpcapFileFormat