

Liste des sujets de projets M2 PRO/ALT

Module SPROJ 2006-2007

Optimisation de route dans Mobile IPv6	3
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)	3
Support de l'algorithme d'adaptation de débit dans le simulateur NS-2	4
Luigi Iannone (luigi.iannone@lip6.fr)	4
Support de l'interférence entre noeuds dans le simulateur NS-2	5
Luigi Iannone (luigi.iannone@lip6.fr)	5
Débit du trafic utilisateur dans 802.11	6
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)	6
Handover vertical	7
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)	7
Protocoles de routage Ad Hoc	8
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)	8
Etude du point d'accès surchargé dans 802.11	9
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)	9
Développement d'une application P2P sur un réseau Wifi	10
Bénédict Le Grand (Benedicte.Le-Grand@lip6.fr), Panayotis Antoniadis, Johanna Chouffane	10
Extension d'applications P2P avec des réseaux sociaux	11
Bénédict Le Grand (Benedicte.Le-Grand@lip6.fr), Panayotis Antoniadis, Johanna Chouffane	11
Créer un sujet de TP mettant en évidence La configuration et le fonctionnement de réseaux Wifi	12
Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)	12
Développement d'un outil de détection d'attaques	13
Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)	13
Protocole de transport en environnement hétérogène	14
Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)	14
Analyse des protocoles multicast pour les réseaux mobile ad hoc.	15
Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)	15
Création d'une application autonome gérant le contrôle, l'accessibilité et le transfert de données topologique vers un serveur central.	16
Timur Friedman (timur.friedman@lip6.fr), copie à Benoit Donnet (donnet@info.ucl.ac.be), copie à Thomas Bourgeau (thomas.bourgeau@lip6.fr), en ligne de sujet : "[PRES LOOK]"	16
Développement d'un système efficace et distribué de découverte de la topologie d'Internet	17
Timur Friedman (timur.friedman@lip6.fr), copie à Benoit Donnet (donnet@info.ucl.ac.be), copie à Thomas Bourgeau (thomas.bourgeau@lip6.fr), en ligne de sujet : "[PRES LOOK]"	17
Création d'un portail captif IPv6	18
Konstantin Kabassanov (mail: Konstantin.Kabassanov@lip6.fr)	18
Déploiement d'une architecture Mini-Infradio*	19
Konstantin Kabassanov (mail: Konstantin.Kabassanov@lip6.fr)	19
Déploiement de mécanismes d'authentification Mini-Infradio*	20
Konstantin Kabassanov (mail: Konstantin.Kabassanov@lip6.fr)	20
Extension en JAVA d'une plateforme pour applications réseaux 802.11	21
Marcelo Dias de Amorim (amorim@rp.lip6.fr), Fehmi Ben Abdesslem (fehmi@rp.lip6.fr), - Luigi Iannone (iannone@rp.lip6.fr)	21
Développement d'une application graphique pour une application 802.11	22
Marcelo Dias de Amorim (amorim@rp.lip6.fr), Fehmi Ben Abdesslem (fehmi@rp.lip6.fr), - Luigi Iannone (iannone@rp.lip6.fr)	22
Implémentation de protocoles de routage réactifs cross-layer sur une plateforme de prototypage	23
Marcelo Dias de Amorim (amorim@rp.lip6.fr), Fehmi Ben Abdesslem (fehmi@rp.lip6.fr), - Luigi Iannone (iannone@rp.lip6.fr)	23

Simulation comparative de la QoS dans WIMAX et UMTS	24
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr) (responsable), Tara Ali-Yahiya (Tara.Ali-Yahiya@rp.lip6.fr)	_ 24
Simulation comparative de la QoS dans WIFI et WIMAX	25
Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr) (responsable), Tara Ali-Yahiya (Tara.Ali-Yahiya@rp.lip6.fr)	_ 25

Optimisation de route dans Mobile IPv6

Encadrante :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)

Nombre d'étudiants demandés:

2-3

Description générale:

Le but de ce projet est de mettre en place une plateforme Mobile IPv6 et d'étudier la fonctionnalité d'optimisation de route dans Mobile IPv6.

Pré-requis :

Mobile IPv6, Linux.

Travail à réaliser :

- Réaliser une étude théorique sur l'architecture et le fonctionnement du protocole Mobile IPv6.
- Définir l'architecture du plateforme et les scénarios de test permettant de montrer l'intérêt de la fonctionnalité d'optimisation de route dans Mobile IPv6.
- Mettre en place la plateforme, réaliser les tests et comparer les résultats obtenus dans les cas sans optimisation de route et avec optimisation de route.

Liens complémentaires :

[1] <http://www.ietf.org/rfc/rfc3775.txt>

[2] <http://www.mipl.mediapoli.com/>

Remarques :

Matériel : 5 portables ou 5 stations équipés des cartes 802.11 donc 2 ont en plus des cartes Ethernet.

Support de l'algorithme d'adaptation de débit dans le simulateur NS-2

Matériel nécessaire : un PC au Laptop par personne de préférence avec système d'exploitation Linux ou BSD.

Encadrant:

Luigi Iannone (luigi.iannone@lip6.fr)

Nombre d'étudiants demandés: 2

Description générale: Le simulateur NS-2 est un des outils le plus répandus dans le domaine de la recherche en réseaux. Il permet de simuler des réseaux (filaire et sans fil) de taille importante. NS-2 doit son succès au fait qu'il est un logiciel libre. Dans les dernières années, il a été utilisé beaucoup pour simuler des réseaux sans fil à grande échelle. Néanmoins, étant donné que les extensions pour la partie sans fil est relativement jeune, la modélisation n'est pas totalement correcte et il manque de détails importants pour réaliser des simulations réalistes.

Pré-requis: Programmation en C/C++ et connaissance de la couche MAC 802.11.

Travail à réaliser: Le simulateur NS-2 ne prévoit pas le support pour l'algorithme d'adaptation du débit de transmission. Un premier module a été développé dans le LIP6, mais il doit être adapté à la dernière version de NS-2 et aussi être amélioré. En particulier l'algorithme d'adaptation doit être capable de différencier chaque voisin d'un nœud et être capable d'adapter le débit de façon indépendante pour chacun d'entre eux. Le groupe de travail doit réaliser cette dernière partie et fournir un « patch » pour réaliser des simulations réalistes.

Liens complémentaires:

<http://www.isi.edu/nsnam/ns/>

Remarques:

Support de l'interférence entre noeuds dans le simulateur NS-2

Matériel nécessaire : un PC au Laptop par personne de préférence avec système d'exploitation Linux ou BSD.

Encadrant:

Luigi Iannone (luigi.iannone@lip6.fr)

Nombre d'étudiants demandés: 2-3

Description générale: Le simulateur NS-2 est un des outils le plus répandus dans le domaine de la recherche en réseaux. Il permet de simuler des réseaux (filaire et sans fil) de taille importante. NS-2 doit son succès au fait qu'il est un logiciel libre. Dans les dernières années, il a été utilisé beaucoup pour simuler des réseaux sans fil à grande échelle. Néanmoins, étant donné que les extensions pour la partie sans fil est relativement jeune, la modélisation n'est pas totalement correcte et il manque de détails importants pour réaliser des simulations réalistes.

Pré-requis: Programmation en C/C++ et connaissance de la couche MAC 802.11.

Travail à réaliser: Le simulateur NS-2 a un support très limité pour ce qui concerne l'interférence entre les nœuds. En particulier NS-2 est seulement capable de comprendre s'il y a une collision entre trame de la couche MAC mais il est pas capable de modéliser la perte des paquets dû au niveau d'interférence excessive produit par les autres nœuds. Le groupe de travail doit implémenter un modèle d'interférence correct et modifier NS-2. NS-2 devra être capable de produire un nouveau type de trace avec l'évolution de l'interférence sur chaque nœud du réseau simulé.

Liens complémentaires: <http://www.isi.edu/nsnam/ns/>

Débit du trafic utilisateur dans 802.11

Encadrante :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)

Nombre d'étudiants demandés : 3

Description générale :

Le but de ce projet est de mettre en évidence les facteurs influant au débit du trafic utilisateur dans un réseau 802.11.

- Interférence
- Puissance du signal
- Stations avec les applications gourmandes
- Algorithme de slow-start de TCP
- Nombre d'utilisateurs
- ...

Pré-requis :

802.11, Linux, TCP/IP.

Travail à réaliser :

- Définir en détail les facteurs qui d'après vous ont l'impact sur le débit d'un trafic utilisateur d'une station 802.11.
- Définir les scénarios de test permettant de montrer les impacts de ces facteurs.
- Réaliser les tests définis et rendre le rapport avec les résultats obtenus.

Liens complémentaires :

[1] Standard 802.11 (fourni par l'encadrant).

Remarques :

Matériel : 3 portables et 1 point d'accès

Handover vertical

Encadrante :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)

Nombre d'étudiants demandés : 3

Description générale :

Un handover vertical est un changement de réseaux de différentes technologies (par exemple, entre le réseau GPRS et le réseau WLAN). Un terminal multi-interface est un terminal qui a au moins deux interfaces réseaux de deux technologies différentes. Le but de ce projet est de mettre en place un système simple de gestion du handover vertical dans un réseau avec les terminaux multi-interface. Selon certaines caractéristiques comme la puissance du signal ou la charge du réseau, le système va décider quelle interface à utiliser.

Pré-requis :

802.11, Linux, éventuellement d'autres technologies sans fil.

Travail à réaliser :

- Définir le testbed avec deux types de réseaux à réaliser le handover vertical.
- Définir les critères permettant de prendre la décision de l'interface réseau à utiliser dans le terminal.
- Réaliser un programme permettant de gérer les interfaces réseaux et exécuter la décision de handover vertical.
- Tester le système avec au moins un handover vertical réalisé.

Liens complémentaires :

[1] Standard 802.11 (fourni par l'encadrant).

Remarques :

Matériel : 1 ou 2 portables.

Protocoles de routage Ad Hoc

Encadrante :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)

Nombre d'étudiants demandés : 4

Description générale :

Le but de ce projet est de mettre en évidence deux approches, réactive et proactive, dans les protocoles de routage ad-hoc en analysant deux protocoles de routage ad-hoc AODV et OLSR.

Pré-requis :

Routage ad-hoc, Linux, Ethereal, AODV, OLSR.

Travail à réaliser :

- Le pré rapport doit contenir une analyse des deux approches, réactive et proactive, dans les protocoles de routage ad - hoc, notamment sur deux protocoles AODV et OLSR
- Réaliser un réseau ad-hoc pour tester deux protocoles AODV et OLSR
- Pour chaque protocole, analyser les traces des messages et les informations dans les tables de routage pour mettre en évidence le fonctionnement du protocole et surtout pour montrer l'aspect « réactif » ou « proactif » du protocole.
- Rendre un rapport avec la recherche bibliographique et les analyses effectuées.

Liens complémentaires :

[1] <http://ethereal.com/>

[2] <http://www.ietf.org/rfc/rfc3561.txt>

[3] <http://www.ietf.org/rfc/rfc3626.txt>

[4] http://tuxmobil.org/manet_linux.html

Remarques :

Matériel : 3 ou 4 portables.

Etude du point d'accès surchargé dans 802.11

Encadrante :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr)

Nombre d'étudiants demandés: 3

Description générale:

Un point d'accès est considéré comme surchargé quand le volume de trafic que les stations souhaitent envoyer et recevoir dépasse la capacité d'un point d'accès. Par conséquent, la QoS des applications dans les stations est faible. Une méthode permettant de caractériser l'état de surcharge d'un point d'accès est d'assigner un certain nombre de points à chaque type de trafic d'application. Par exemple, une communication de la voix vaut 3 points, une communication FTP vaut 10 points, une communication de vidéo vaut 15 points, etc. Quand la somme des points des trafics passant par un point d'accès atteint un certain seuil, le point d'accès est considéré comme surchargé. Le but de ce projet est d'étudier l'état de surcharge d'un point d'accès dans l'environnement 802.11 par une simulation sur NS en utilisant la méthode décrite ci-dessus.

Pré-requis :

802.11, NS (Network Simulator).

Travail à réaliser :

- Définir les scénarios de simulation permettant d'étudier l'état de surcharge d'un point d'accès.
- Réaliser les simulations, proposer un plan d'assignement des points aux différents types de trafic et le seuil avec lequel un point d'accès est considéré comme surchargé.
- Donner l'avis sur la méthode de caractérisation de la surcharge de point d'accès par assignement des points. Est-ce que cette méthode est adéquate pour caractériser la surcharge d'un point d'accès ?

Liens complémentaires :

[1] <http://www.isi.edu/nsnam/ns/>

[2] Standard 802.11 (fourni par l'encadrant).

Remarques :

Matériel : 1 ordinateur avec NS

Développement d'une application P2P sur un réseau Wifi

De nombreux logiciels P2P ont été développés pour différents types d'applications, en particulier pour le partage de fichiers (Kazaa, Emule, etc.). Les réseaux P2P constituent un réseau de recouvrement (overlay) au-dessus du réseau IP.

L'objectif de ce projet est de *mettre en œuvre des applications P2P sur des réseaux Wifi*, afin de permettre, afin de permettre, par exemple, les échanges de fichiers à l'intérieur d'un réseau sans fil constitué de terminaux mobiles et de points d'accès sans fil (dans le cadre de ce projet, des ordinateurs portables devront être configurés pour fonctionner comme point d'accès). Un réseau spontané devra être construit à partir des terminaux mobiles et des points d'accès, qui constitueront ainsi une *communauté* pour partager des fichiers.

Une application P2P de partage de fichier devra donc être adaptée aux réseaux sans fil. Après cette première étape, des mesures de performance seront effectuées afin d'évaluer l'impact des communications sans fil sur le service fourni.

Pré-rapport :

- Étudier les logiciels de P2P disponibles en open source et sélectionner celui ou ceux qui seront mis en œuvre et adaptés pour l'utilisation dans un environnement ad hoc sans fil.
- Spécifier les étapes pour construire le réseau Wifi (configuration des points d'accès, découverte des points d'accès, construction du réseau, choix du protocole de routage etc.).
- Décrire les critères de performance qui seront évalués lors de la phase de mesure (par ex temps de téléchargement d'un fichier en fonction du nombre de pairs et de leur localisation, etc.)

Pour ce projet, les étudiants devront disposer de PC portables munis d'une carte Wifi.

Développement Java sous linux (de préférence).

Groupe de 3 étudiants (pour pouvoir tester la mobilité et le multi-saut), si possible équipés de PC portables avec carte Wifi.

Encadrement :

Bénédicte Le Grand (Benedicte.Le-Grand@lip6.fr), Panayotis Antoniadis, Johanna Chouffane

Extension d'applications P2P avec des réseaux sociaux

De nombreuses applications P2P ont été développées, en particulier pour le partage de fichiers (Kazaa, Emule, etc.). Par ailleurs, on observe également la croissance des applications « sociales » (*social software*), qui permettent aux internautes de créer des communautés virtuelles (MySpace, flickr, etc.). Les réseaux F2F (Friend to Friend) sont des réseaux P2P dans lesquels les utilisateurs ne créent des connexions directes qu'avec des personnes en qui ils ont confiance. Ainsi, seuls les amis d'un utilisateur, les amis de ses amis (etc.) peuvent télécharger ses ressources.

L'objectif de ce projet est d'intégrer les fonctionnalités des logiciels sociaux dans des applications P2P, afin de pouvoir :

- créer un groupe, c'est-à-dire un réseau P2P composé uniquement de membres d'un réseau social existant (c'est-à-dire que l'on veut faire en sorte que les terminaux qui appartiennent à une même communauté puissent communiquer à travers ce réseau). Les communications à l'intérieur d'un groupe seront *chiffrées* à l'aide de clés publiques.
- créer un réseau social à partir de pairs impliqués dans une application P2P, en sélectionnant certains participants selon des critères à définir (à l'inverse, on veut créer des communautés de terminaux dans des réseaux qui existent déjà)

Les utilisateurs devront pouvoir modifier leur réseau social de manière dynamique : ils pourront par exemple décider d'éliminer certains membres du réseau P2P si ceux-ci ne partagent pas leurs ressources (free-riders). La suppression des utilisateurs non coopératifs pourra même être effectuée de manière automatique. Pour cela, des mécanismes de surveillance seront mis en œuvre afin de contrôler la participation (ou non) des utilisateurs, en terme de partage de fichiers. Ces mécanismes devront être mis en œuvre au niveau de chaque pair (système *distribué*).

Chaque utilisateur possèdera un profil qui regroupe des informations le concernant (type de contenu favori par exemple pour le cas du partage de fichiers) ainsi que la liste de ses contacts et des groupes qu'il a constitué à partir de ceux-ci. A la différence de la plupart des systèmes existants, ces informations doivent rester *chez l'utilisateur*, c'est-à-dire que les informations concernant les utilisateurs ne sont *pas centralisées*.

Pré-rapport :

- Etudier les logiciels de P2P disponibles en open source et sélectionner celui ou ceux qui seront mis en œuvre dans le projet.
- Etudier les logiciels « sociaux » (*social software*) disponibles en open source et sélectionner celui ou ceux qui seront mis en œuvre dans le projet.
- Proposer des mécanismes permettant d'évaluer la participation des pairs en terme de mise à disposition de fichiers (afin d'éliminer les free riders des réseaux sociaux).
- Proposer des solutions pour éliminer les free-riders des réseaux sociaux et, au contraire, pour y inclure les membres qui coopèrent.

Les étudiants devront disposer de PC portables munis d'une carte Wifi.

Développement Java sous linux (de préférence).

Groupe de 2 étudiants.

Encadrement :

Bénédicte Le Grand (Benedicte.Le-Grand@lip6.fr), Panayotis Antoniadis, Johanna Chouffane

Créer un sujet de TP mettant en évidence La configuration et le fonctionnement de réseaux Wifi

BUT :

Le but de votre projet est de composer un sujet de TP de 4h permettant :

- la configuration de machines dans un mini réseau local wifi/filaire avec 2 points d'accès et une station d'accueil.
- la compréhension de ce qu'est un pont versus un routeur.
- la compréhension des mécanismes du Wifi.

Vous allez créer un CD de distribution qui puisse être utilisé dans la salle machine. Elle devra donc inclure les bons drivers (e.g. Madwifi pour le sans fil).

ARCHITECTURE DU TP : 3 PC reliés par deux cartes Ethernet et possédant tous une interface 802.11b/g DLINK DWL-G520.

Encadrante :

Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)

TACHES :

Architecture du TP :

- Les étudiants travaillent par groupes de 3 binômes et disposent donc de 3 pc.
- Faire en sorte qu'il y ait deux points d'accès et que cela soit transparent pour la troisième machine qui se connecte indifféremment à l'un ou à l'autre.
- Possibilité de configurer les stations soit en point d'accès/routeur soit en point d'accès simple ou encore en simple station wifi (pas point d'accès).
 - Attention à permettre de faire varier ce que l'on peut configurer (opérer avec RTS/CTS, fragmentation, choix du canal de fréquence, cryptage,...)

Vous devrez d'abord réfléchir :

- Aux différentes fonctionnalités qui pourront être testées et au moyen de mettre certains aspects en évidence.
Par exemple :
 - Comment faire en sorte que la station puisse se connecter à deux point d'accès différent de manière transparente (SSID commun)
 - Différence entre un *point d'accès/routeur* ou un *point d'accès simple* : proposer une architecture et un moyen de faire des traces permettant d'analyser cela.

Proposer des mini configurations qui permettent de faire réfléchir aux étudiants au moyen de mettre en évidence cette différence.

Dans un deuxième temps, vous devrez réfléchir au moyen de configurer les interfaces réseau de vos machines sans avoir accès au disque. En effet, ce CD devra être sécurisé, c'est-à-dire que l'écriture sur le disque devra être impossible. Pour rendre possible ce TP, certains fichiers ou paramètres de configuration pourront/devront être accessibles grâce à une clé USB. Il vous faudra les identifier et créer le CD en conséquence.

MODALITES :

Avant de débiter les projets vous devrez présenter un pré-rapport définissant la trame du sujet de TP en vous étant assuré de la cohérence de ceux-ci par rapport aux contraintes du CD de distribution.

Ce pré-rapport devra être validé avant la semaine de tests et d'implémentations. Vous vous baserez sur le modèle de TP donné en annexe.

A la fin de l'implémentation devront être rendus une version électronique :

- d'un sujet de TP (fichier doc ou tex) comprenant l'explication du sujet et la démarche qui doit être suivie par les étudiants. Une réponse en texte masqué des résultats aux commandes (route, ifconfig,...) et à la configuration des fichiers.
- Le CD de la distribution qui aura été créée.
- Une soutenance est prévue afin de montrer la validité du travail.

Développement d'un outil de détection d'attaques

BUT : Développer un outil de détection d'attaques sur un réseau local

Encadrante :

Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)

Lieu du projet : Campus de Jussieu

Architecture : Vous disposerez de 4 machines sur un réseau local.

TACHES :

Prérapport :

- Lister les attaques réseau possibles sur un réseau (exemple : déni de service, recherches de vulnérabilités)
- Lister les applications/outils disponibles pour effectuer ces attaques (Linux)
- Définir une architecture permettant de mettre en évidence ce type d'attaques
- Définir les attaques que vous souhaiteriez détecter

Développement

- Implémenter une application qui affichera des alertes lorsque les attaques ou des comportements anormaux sur le réseau seront détectés.
- Rendre le rapport contenant une analyse de la plateforme, des difficultés rencontrées et des traces

MODALITES

Avant de débiter les projets vous devrez présenter un pré-rapport précisant les traces que vous voudrez identifier pour mettre en évidence le fonctionnement de l'ensemble de ces services. Ce pré rapport devra être validé avant la semaine de tests et d'implémentations.

A la fin de la semaine d'implémentation devront être rendus :

- le rapport contenant les scénarios pour mettre en évidence le bon fonctionnement des différents services et les fonctions intéressantes de chacun (pré-rapport).
- une version électronique des traces, qui pourront être visualisées avec Ethereal ainsi qu'un rapport sur ce quelles mettent en évidence.

Ce sujet est proposé pour 1 groupe de 3 étudiants

Protocole de transport en environnement hétérogène

Encadrante :

Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)

Nombre d'étudiants demandés: 3

Description générale:

Les protocoles usuels de l'Internet (TCP/IP) doivent être adaptés dans le cadre d'environnements contraints tels que les réseaux mobiles ad hoc (Mobile Ad hoc NETWORKS – MANETS). Dans le cadre de communications réseaux traversant ces environnements, la mise en place de proxy TCP ou de *protocol boosters* est souvent nécessaire.

Le but de ce projet est de recenser les implémentations libres (open source) de ces proxy, et de mettre en évidence leur utilité au travers de la mise en œuvre de scénarios de test sur une plateforme WiFi ad hoc Linux.

Pré requis:

- Être familier avec Linux.
- 4 portables.

Travail à réaliser:

- Identification des proxy (ou booster) libres
- Définition des scénarios permettant de montrer l'utilité
- Étude technique (outils de mesure, installations nécessaires, ...)
- Mise en œuvre des scénarios sur un réseau ad hoc.

Analyse des protocoles multicast pour les réseaux mobile ad hoc.

Encadrante :

Anne Fladenmuller (Anne.Fladenmuller@upmc.fr)

Nombre d'étudiants demandés: 3

Description générale:

Les communications de groupe constituent un service de premier ordre dans de nombreux secteurs d'activité (e.g. forces de sécurité civile, diffusion video). Le multicast dans les réseaux ad hoc s'avère être une problématique intéressante qui permettrait d'apporter assez de flexibilité pour optimiser les moyens de communication dans ces environnements très contraints.

Le but de ce projet est de recenser les implémentations libres (open source) de diverses solutions multicast (e.g. MOLSR, MAODV, BMF) et de mettre en évidence leur pertinence au travers de la mise en œuvre de scénarios de test sur une plate-forme WiFi ad hoc Linux en tenant compte de la mobilité des nœuds.

Pré requis:

- Etre familier avec Linux.
- 4 portables.

Travail à réaliser:

- Étude des protocoles multicast implémentés sous Linux pour les MANETs ;
- Définition des scénarios de tests
- Étude technique (outils de mesure, installations nécessaires, ...)

Mise en place d'un démonstrateur intégrant un service multimédia (VoIP ou video) sur la plate-forme ad hoc.

Création d'une application autonome gérant le contrôle, l'accessibilité et le transfert de données topologique vers un serveur central.

Encadrants: Timur Friedman, Benoit Donnet, Thomas Bourgeau

[Timur Friedman \(timur.friedman@lip6.fr\)](mailto:timur.friedman@lip6.fr), copie à [Benoit Donnet \(donnet@info.ucl.ac.be\)](mailto:donnet@info.ucl.ac.be), copie à [Thomas Bourgeau \(thomas.bourgeau@lip6.fr\)](mailto:thomas.bourgeau@lip6.fr), en ligne de sujet : "[PRES LOOK]"

Nombre d'étudiants: 4

Description générale:

Afin de contribuer au développement de systèmes de mesures et de découverte de la topologie d'Internet, nous avons développé une application simple, coopérative et largement distribuée.

Cette approche s'appuie sur une application classique appelée *traceroute* [1] et dont le principe de fonctionnement consiste à découvrir le chemin qu'un paquet de données prend pour aller d'une machine S (la source ou moniteur) vers une machine D (la destination). Cependant cette technique résiste mal au facteur d'échelle et induit de la redondance dans les résultats.

Afin de résoudre ces problèmes et apporter une solution plus robuste, le prototype d'un algorithme coopératif de découverte de topologie appelé Doubletree [2] fut proposé. Une implémentation de Doubletree a été réalisée en Java ; nous l'avons appelée *traceroute@home* [3]. Il s'agit d'une implémentation totalement distribuée dont la coopération entre les différents moniteurs Doubletree se fait au moyen de *tables de hachage distribuées* (DHTs) [4]. Le déploiement de *traceroute@home* sur le réseau PlanetLab [5] nous a donné des résultats convaincant mais pour le moment les données liées à la topologie mesurée résident dans la mémoire de chaque moniteur ayant réalisé des mesures avec *traceroute@home*.

Ce travail, qui s'inscrit dans le cadre d'un projet européen appelé [OneLab](#), a pour but de réaliser une application autonome permettant de récupérer les données de topologie réseau, au format XML, sur le disque de chaque moniteur exécutant l'application *traceroute@home* et de les envoyer sur un serveur central qui sera géré en [PostgreSQL](#). Le système devra s'assurer d'avoir rapatrié les données avant d'effacer la mémoire d'un moniteur. Il faudra aussi qu'il n'y ait pas de conflits entre l'écriture des données par *traceroute@home* et le rapatriement des données vers le serveur. Une petite interface sera réalisée afin d'interroger la Base de Données. On pourra par exemple demander le(s) chemin(s) entre la source S et la destination D, la variation du/des délai(s) entre S et D.

Afin de contrôler et piloter toutes les applications à partir d'un moniteur, nous utilisons un programme appelé Application Manager [6]. Il serait intéressant d'utiliser ce programme pour optimiser l'application développée lors du projet. Ce programme open-source sera réalisé en C et devra fonctionner en tâche de fond et récupérera les données selon une période déterminée. La problématique abordée pendant ce projet est un thème de recherche qui intéresse aussi l'industrie.

Prérequis : Programmation C, XML, DTD, Bases de données (PostgreSQL).

Travail à réaliser:

- Phase d'analyse sur l'architecture du programme à réaliser : contrôle, accessibilité et transfert de données.
- Structure client/serveur.
- Implémentation du programme en C pour Linux, FreeBSD, Darwin et Windows.
- Création d'une petite interface pour interroger la Base de Données.
- Déploiement du programme en interaction avec *traceroute@home*.
- Tester la résistance de l'application au facteur d'échelle.

Références:

[1]: <http://traceroute.org/>

[2]: B. Donnet, P. Raoult, T. Friedman, M. Crovella. "Efficient algorithms for large-scale topology discovery", in Proc. ACM SIGMETRICS, Jun. 2005.

[3]: D. Camara, B. Donnet, F. Picconi, T. Friedman, P. Sens. "Towards a Highly Distributed Topology Measurement System", under review, Jan. 2007. Java code available:

<http://gforge.info.ucl.ac.be/projects/trhome/>

[4]: <http://trhome.sourceforge.net/>

[5]: <http://www.planet-lab.org>

[6]: <http://appmanager.berkeley.intel-research.net/>

Développement d'un système efficace et distribué de découverte de la topologie d'Internet

Encadrants: Timur Friedman, Benoit Donnet, Thomas Bourgeau

Timur Friedman (timur.friedman@lip6.fr), copie à Benoit Donnet (donnet@info.ucl.ac.be), copie à Thomas Bourgeau (thomas.bourgeau@lip6.fr), en ligne de sujet : "[PRES LOOK]"

Nombre d'étudiants: 4

Description générale:

Le développement et la fulgurante mutation des réseaux au niveau mondial a suscité depuis plusieurs années un intérêt grandissant pour la création de systèmes de mesures et de découverte de la topologie Internet sous forme d'applications simples, coopératives et largement distribuées.

La communauté réseau s'intéresse de près à cette problématique et contribue fortement à une réelle avancée dans ce domaine (Skitter [1], Dimes [2], Scriptroute [3], ...)

Une approche classique permet de découvrir la topologie d'Internet au niveau IP en se basant sur l'outil réseau *traceroute* [4] dont le principe de fonctionnement consiste à découvrir le chemin qu'un paquet de données prend pour aller d'une machine S (la source ou moniteur) vers une machine D (la destination).

Cependant cette technique résiste mal au facteur d'échelle et induit de la redondance dans les résultats. Afin de résoudre ces problèmes et apporter une solution plus robuste, le prototype d'un algorithme coopératif de découverte de topologie appelé *Doubletree* [5] fut proposé. Une implémentation de Doubletree a été réalisée en Java ; nous avons appelé le prototype *traceroute@home* [6]. Il s'agit d'une implémentation totalement distribuée dont la coopération entre les différents moniteurs Doubletree se fait au moyen de *tables de hachage distribuées* (DHTs) [7]. Le déploiement de *traceroute@home* sur le réseau PlanetLab [8] nous a donné des résultats convaincant sur un nombre de moniteurs limité.

Ce travail, qui s'inscrit dans le cadre d'un projet européen appelé [OneLab](#), a pour but de réaliser une version de *traceroute@home* la plus efficace possible; l'application réalisée devra tourner sous Windows et sous Linux. Afin de réaliser des mesures de topologie sur un maximum de systèmes, nous vous proposons de réaliser un programme en C++ suivant le modèle de *traceroute@home*. Ce programme open-source devra fonctionner en tâche de fond et pourrait être intégré par la suite à un écran de veille afin de réaliser des mesures en continu sur PlanetLab. La problématique abordée pendant ce projet est un thème de recherche qui intéresse aussi l'industrie.

Prérequis : Programmation réseaux (langage Java et C++), XML, rigueur.

Travail à réaliser:

Un véritable travail d'équipe devra être mis en œuvre pour la réussite de ce projet : ceci est un réel exercice à de préparation à la vie professionnelle.

Phase d'analyse sur le principe et la structure du programme *traceroute@home*.

- 1. Proposition d'une structure de programme en C++.
- 1. Implémentation du programme en C++.
- 1. Déploiement de l'application sur le réseau PlanetLab sur un maximum de systèmes différent (Windows/Linux) afin de tester l'efficacité du programme.

Informations complémentaires:

[1]: <http://www.caida.org/tools/measurement/skitter/>

[2]: <http://www.netdimes.org/new/>

[3]: <http://traceroute.org/>

[4]: B. Donnet, P. Raoult, T. Friedman, M. Crovella. "Efficient algorithms for large-scale topology discovery", in Proc. ACM SIGMETRICS, Jun. 2005.

[5]: D. Camara, B. Donnet, F. Picconi, T. Friedman, P. Sens. "Towards a Highly Distributed Topology Measurement System", under review, Jan. 2007. Java code available: <http://gforge.info.ucl.ac.be/projects/trhome/>

[6]: <http://www.planet-lab.org>

Création d'un portail captif IPv6

BUT : Fournir un portail captif IPv6 utilisable.

- Chercher et comparer des solutions de portails captifs IPv4 existants.
- Vérifier l'existence de portails captifs IPv6.
- En se basant sur les fonctionnalités du portail captif NoCat (ou un autre qui vous semble meilleur), développer un portail captif IPv6.

Encadrant :

Konstantin Kabassanov (mail: Konstantin.Kabassanov@lip6.fr)

Tâches :

- Ecrire un cahier des charges.
- Découper le travail en sous-tâches disjointes.
- Définir l'architecture logique.
- Choisir et installer le système d'exploitation.
- Se répartir le reste du travail.
- Développement.
- Définir des scénarios de test.
- Validation de la solution proposée.

ARCHITECTURE

2 ordinateurs, dont 1 avec 2 interfaces réseaux pour le portail (1 interface en ipv6 uniquement pour les clients et une autre vers l'extérieur).

MODALITES

Avant de débiter les projets vous devrez présenter au plus tard la semaine qui précède votre semaine de développement un pré-rapport spécifiant :

- Le cahier de charge.
- Le découpage du projet en sous-projets et la répartition des personnes.
- La conception de l'architecture et de la solution logicielle proposée.

Ce pré rapport devra être validé avant le début du déploiement et des tests.

A la fin de la semaine du projet devront être rendus :

- Le pré-rapport.
- Un schéma complet de l'architecture.
- Les jeux de tests effectués et leurs résultats.

Ce sujet est proposé pour 1 groupe de 3 ou 4 étudiants

Le travail sera effectué dans les locaux du LIP6, situés avenue du Président Kennedy (Paris 16).

Déploiement d'une architecture Mini-Infradio*

*Ce sujet est étroitement lié avec le sujet « Déploiement de mécanismes d'authentification Mini-Infradio »

BUT : Déployer une maquette de taille réduite de la plate-forme Infradio.

- Créer un réseau d'accès exclusivement IPv6.
- Déployer un pare-feu.
- Déployer un serveur DNS IPv6 avec une interconnexion avec un serveur double pile (IPv4/IPv6).
- Déployer un serveur proxy IPv6 – IPv4.
- Déployer un serveur de mail IPv6 communiquant avec le reste du monde en IPv6 et IPv4, ainsi qu'un serveur web uniquement IPv6.

Encadrant :

Konstantin Kabassanov (mail: Konstantin.Kabassanov@lip6.fr)

Tâches :

- Ecrire un cahier des charges.
- Découper le travail en sous-tâches disjointes.
- Définir l'architecture logique.
- Choisir et installer le système d'exploitation.
- Se répartir le reste du travail.
- Déploiement.
- Définir des scénarios de test.
- Validation de la solution proposée.

ARCHITECTURE

3 ou 4 ordinateurs, dont 2 ou 3 avec 2 interfaces réseaux pour le routeur, le serveur et/ou serveur proxy (1 interface en ipv6 uniquement et une autre double pile) et le quatrième en tant que client standard ipv6 uniquement.

MODALITES

Avant de débuter les projets vous devrez présenter au plus tard la semaine qui précède votre semaine de déploiement un pré-rapport spécifiant :

- Le cahier de charge.
- Le découpage du projet en sous-projets et la répartition des personnes.
- La conception de l'architecture et de la solution logicielle proposée.

Ce pré rapport devra être validé avant le début du déploiement et des tests.

A la fin de la semaine du projet devront être rendus :

- Le pré-rapport.
- Un schéma complet de l'architecture.
- Les jeux de tests effectués et leurs résultats.

Ce sujet est proposé pour 1 groupe de 4 étudiants

Le travail sera effectué dans les locaux du LIP6, situés avenue du Président Kennedy (Paris 16).

Déploiement de mécanismes d'authentification Mini-Infradio*

Ce sujet est étroitement lié avec le sujet « Déploiement d'une architecture Mini-Infradio»

BUT : Déployer une maquette de taille réduite des mécanismes d'authentification Infradio.

- Déployer une base LDAP.
- Déployer un serveur RADIUS.
- Déployer un ensemble de certificats.
- Définir l'interaction avec le pare-feu d'une architecture existante et la mettre en pratique si possible.
- Eventuellement, déploiement d'un portail captif IPv6.

Encadrant :

Konstantin Kabassanov (mail: Konstantin.Kabassanov@lip6.fr)

Tâches :

- Ecrire un cahier des charges.
- Découper le travail en sous-tâches disjointes.
- Définir l'architecture logique.
- Choisir et installer le système d'exploitation.
- Se répartir le reste du travail.
- Déploiement
- Définir des scénarios de test.
- Validation de la solution proposée.

ARCHITECTURE

2 ou 3 ordinateurs, dont 1 ou 2 avec 2 interfaces réseaux pour le serveur LDAP/RADIUS et éventuellement le portail captif IPv6 et un autre ordinateur en tant que client standard ipv6 uniquement.

MODALITES

Avant de débiter les projets vous devrez présenter au plus tard la semaine qui précède votre semaine de déploiement un pré-rapport spécifiant :

- Le cahier de charge.
- Le découpage du projet en sous-projets et la répartition des personnes.
- La conception de l'architecture et de la solution logicielle proposée.

Ce pré rapport devra être validé avant le début du déploiement et des tests.

A la fin de la semaine du projet devront être rendus :

- Le pré-rapport.
- Un schéma complet de l'architecture.
- Les jeux de tests effectués et leurs résultats.

Ce sujet est proposé pour 1 groupe de 4 étudiants.

Le travail sera effectué dans les locaux du LIP6, situés avenue du Président Kennedy (Paris 16).

Extension en JAVA d'une plateforme pour applications réseaux 802.11

Encadrants : -

Marcelo Dias de Amorim (amorim@rp.lip6.fr), Fehmi Ben Abdesslem (fehmi@rp.lip6.fr), - Luigi Iannone (iannone@rp.lip6.fr)

Nombre d'étudiants : 2 étudiants

Contexte :

L'implémentation et le déploiement de nouveaux protocoles pour réseaux sans-fil 802.11 sur des plateformes réelles d'expérimentation s'avèrent souvent laborieux et requièrent des connaissances techniques spécifiques (implémentation des pilotes, connaissance des chipsets, des requêtes vers le noyau, etc). Afin de faciliter la conception et l'évaluation de ces nouveaux protocoles sans passer par cette implémentation, le laboratoire d'informatique de l'université Pierre et Marie Curie a conçu une plateforme de prototypage d'algorithmes de communication sans-fil, en collaboration avec l'université de Californie, à Santa Cruz (UCSC). Cette plateforme s'inscrit dans le projet européen WIP, qui a pour but de trouver de nouveaux protocoles et architectures pour les réseaux 802.11.

Description du travail à réaliser :

Une librairie intégrée à la plateforme de prototypage est déjà disponible en langage C, et en langage Perl. Cette librairie devra être portée en Java. Il s'agit d'une librairie utilisant les interfaces réseaux, sous linux. Des travaux sur le noyau de la plateforme (implémenté en C), et une mise à niveau des autres librairies (en C et en Perl) sont également à envisager.

Matériel mis à disposition :

Les étudiants auront à leur disposition tout le matériel nécessaire aux travaux d'implémentation, notamment des Nokia N770, des portables DELL X1, et des MiniPC sous Linux, tous équipés d'une interface réseau 802.11.

Pré-requis :

- Langage de programmation Java. Une maîtrise du langage C et Perl serait appréciée.
- Implémentation d'applications réseaux (établissement d'une communication, sockets, etc)

Références :

- La plateforme de prototypage Prawn <http://prawn.lip6.fr>
- Nokia N770 <http://europe.nokia.com/770>

Développement d'une application graphique pour une application 802.11

Encadrants : -

Marcelo Dias de Amorim (amorim@rp.lip6.fr), Fehmi Ben Abdesslem (fehmi@rp.lip6.fr), - Luigi Iannone (iannone@rp.lip6.fr)

Nombre d'étudiants : 2 à 4 étudiants

Contexte :

Le laboratoire d'informatique de l'université Pierre et Marie Curie a conçu une plateforme de prototypage d'algorithmes de communication sans-fil, en collaboration avec l'université de Californie, à Santa Cruz (UCSC). Cette plateforme s'inscrit dans le projet européen WIP, qui a pour but de trouver de nouveaux protocoles et architectures pour les réseaux 802.11. De nombreuses applications simples pour réseaux sans-fil peuvent être développées à l'aide de cette plateforme, afin de réaliser une démonstration pratique de l'utilisation de la plateforme.

Description du travail à réaliser :

Les étudiants devront développer une application graphique capable de s'exécuter sur différentes plateformes, et permettant, à l'aide d'une plateforme de prototypage, d'envoyer un message à un utilisateur du réseau en le sélectionnant dans la représentation graphique dynamique du réseau. Le langage utilisé sera au choix le langage C, ou Perl. L'application utilisera les primitives de la plateforme de prototypage conçue par le laboratoire d'informatique de Paris 6.

Matériel mis à disposition :

Les étudiants auront à leur disposition tout le matériel nécessaire aux travaux d'implémentation, notamment des Nokia N770, des portables DELL X1, et des MiniPC sous Linux, tous équipés d'interface réseau 802.11.

Pré-requis :

- Langage de programmation C et Perl.
- Maîtrise préalable des bibliothèques graphiques (comme GTK+) appréciée.
- Implémentation d'applications réseaux (établissement d'une communication, sockets, etc)

Références :

- La plateforme de prototypage Prawn <http://prawn.lip6.fr>
- Nokia N770 <http://europe.nokia.com/770>
- La bibliothèque GTK+ <http://www.gtk.org>

Implémentation de protocoles de routage réactifs cross-layer sur une plateforme de prototypage

Encadrants : -

Marcelo Dias de Amorim (amorim@rp.lip6.fr), Fehmi Ben Abdesslem (fehmi@rp.lip6.fr), - Luigi Iannone (iannone@rp.lip6.fr)

Nombre d'étudiants : 2 à 4 étudiants

Contexte :

Le laboratoire d'informatique de l'université Pierre et Marie Curie a conçu une plateforme de prototypage d'algorithmes de communication sans-fil, en collaboration avec l'université de Californie, à Santa Cruz (UCSC). Cette plateforme s'inscrit dans le projet européen WIP, qui a pour but de trouver de nouveaux protocoles et architectures pour les réseaux 802.11. De nombreuses applications simples pour réseaux sans-fil peuvent être développées à l'aide de cette plateforme, afin de réaliser une démonstration pratique de l'utilisation de la plateforme.

Description du travail à réaliser :

Les étudiants devront concevoir et développer un protocole de routage réactif (se basant par exemple sur AODV), et utilisant des métriques cross-layer, c'est-à-dire prenant en compte les informations retournées par les couches plus basses comme le taux de perte ou la puissance de réception du signal. Le protocole devra s'exécuter sur un réseau 802.11 composé de nœuds mobiles. Un travail de documentation sur les drivers pour carte sans-fil et une légère amélioration de la plateforme de prototypage pourraient être nécessaires.

Matériel mis à disposition :

Les étudiants auront à leur disposition tout le matériel nécessaire aux travaux d'implémentation, notamment des Nokia N770, des portables DELL X1, et des MiniPC sous Linux, tous équipés d'interface réseau 802.11.

Pré-requis :

- Maîtrise du langage C. La maîtrise du langage Perl serait appréciée.
- Implémentation d'applications réseaux (établissement d'une communication, sockets, etc)
- Bonnes connaissances des protocoles de routage comme AODV.

Références :

- La plateforme de prototypage Prawn <http://prawn.lip6.fr>
- Le protocole AODV <http://moment.cs.ucsb.edu/AODV/aodv.html>

Simulation comparative de la QoS dans WIMAX et UMTS

Encadrantes :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr) (responsable), Tara Ali-Yahiya (Tara.Ali-Yahiya@rp.lip6.fr)

Nombre d'étudiants demandés : 3-4

Description générale :

Le but de ce projet est de réaliser une simulation comparative en terme de QoS de deux technologies WiMax et UMTS.

Selon le WiMax (IEEE 802.16), les applications sont bien déterminées en ensemble de classes de service, chacune a des paramètres de QoS différentes de l'autre. Ces classes sont appelées « service flows » qui se composent de : UGS (voix), rtPS (video), nrtPS (FTP) et BE. Conformément à l'UMTS, on dispose aussi quatre classes de service destinées aux même applications que le WiMax mais avec des QoS différentes : conversationnel (voix), streaming (video), interactive (non temps réel) et background (BE).

La comparaison est faite au niveau des classes de service et leurs paramètres de QoS pour comparer la QoS que ces deux technologies fournissent aux applications.

Pré-requis :

WiMax, UMTS, OPNET.

Travail à réaliser :

- Réaliser une recherche bibliographique sur les classes de service dans WiMax et dans l'UMTS
- Pour chaque type de réseau (WiMax et UMTS), définir des scénarios pour simuler une cellule avec les stations mobiles dans lesquelles se lancent différents types d'application comme voix sur IP, vidéo en format MPEG-4, téléchargement des fichiers via FTP, et des applications de type best-effort comme la messagerie.
- Lancer les simulations avec les scénarios définis pour mesurer la QoS obtenue pour chaque classe de service dans chaque type de réseau en terme de délai, gigue,...
- En se basant sur les résultats obtenus, répondre aux questions telles que : Quelles sont les applications qui se déroulent mieux en UMTS qu'en WiMax et vice versa ? Est-ce que les applications similaires ont les mêmes paramètres de QoS pour ces deux types de réseau ? Quels sont les avantages et les inconvénients des classifications des services dans chaque type de réseau ? Est-ce que la technologie d'accès influence à la QoS de chaque application dans ces réseaux ?

Remarques :

Matériel : 1 ordinateur avec OPNET et Visual C++ (logiciels fournis par l'encadrante).

Simulation comparative de la QoS dans WIFI et WIMAX

Encadrantes :

Nguyen Thi Mai Trang (Thi-Mai-Trang.Nguyen@lip6.fr) (responsable), Tara Ali-Yahiya (Tara.Ali-Yahiya@rp.lip6.fr)

Nombre d'étudiants demandés : 3-4

Description générale :

Le but de ce projet est de réaliser une simulation comparative en terme de QoS de deux technologies WiFi et WiMax.

Selon le WiMax (IEEE 802.16), les applications sont bien déterminées en ensemble de classes de service, chacune a des paramètres de QoS différentes de l'autre. Ces classes sont appelées « service flows » qui se composent de : UGS (voix), rtPS (video), nrtPS (FTP) et BE. Conformément à WiFi (IEEE 802.11e), on dispose aussi quatre classes de service destinées aux mêmes applications que WiMax mais avec des QoS différentes : AC1 (voix), AC2 (video), AC3 (non temps réel) et AC4 (BE).

La comparaison est faite au niveau des classes de service et leurs paramètres de QoS pour comparer la QoS que ces deux technologies fournissent aux applications.

Pré-requis :

802.11e, WiMax, OPNET.

Travail à réaliser :

- Réaliser une recherche bibliographique sur les classes de service dans 802.11e et dans WiMax
- Pour chaque type de réseau (WiFi et WiMax), définir des scénarios pour simuler une cellule avec les stations mobiles dans lesquelles se lancent différents types d'application comme voix sur IP, vidéo en format MPEG-4, téléchargement des fichiers via FTP, et des applications de type best-effort comme la messagerie.
- Lancer les simulations avec les scénarios définis pour mesurer la QoS obtenue pour chaque classe de service dans chaque type de réseau en terme de délai, gigue,...
- En se basant sur les résultats obtenus, répondre aux questions telles que : Quelles sont les applications qui se déroulent mieux en WiFi qu'en WiMax et vice versa ? Est-ce que les applications similaires ont les mêmes paramètres de QoS pour ces deux types de réseau ? Quels sont les avantages et les inconvénients des classifications des services dans chaque type de réseau ? Est-ce que la technologie d'accès influence à la QoS de chaque application dans ces réseaux ?

Remarques :

Matériel : 1 ordinateur avec OPNET et Visual C++ (logiciels fournis par l'encadrante).