LaasNetExp : une plateforme expérimentale pour l'émulation et les tests en réseaux

Philippe Owezarski, Yann Labit, Pascal Berthou, David Gauchard

LAAS-CNRS
Université de Toulouse
7 Avenue du Colonel Roche - 31077 Toulouse cedex 4, France
{owe, ylabit, berthou, gauchard}@laas.fr

RÉSUMÉ. *L'expérimentation en réseaux est incontournable pour évaluer et valider de nouvelles technologies, architectures et protocoles de communication. Il s'avère que les résultats obtenus à partir de simulations sont la plupart du temps irréalistes, les simulateurs ne pouvant pas intégrer tous les modèles de comportement des composants réseaux, des systèmes opératoires et des applications des machines d'extrémité. Des travaux autour de l'émulation et de l'expérimentation réelle en réseaux ont donc vu le jour. Ce papier présente les motivations et besoins qui ont conduit à la mise en place d'une telle plate-forme expérimentale au LAAS - LaasNetExp - et décrit ses principes de conception. L'article détaille comment les conditions expérimentales peuvent être contrôlées pour des expériences reproductibles et faciles à analyser. L'article montre aussi comment des conditions expérimentales réalistes (configuration des émulateurs et générateurs de trafics) peuvent être mises en œuvre à partir des résultats de caractérisation et analyse du trafic Internet. Cet article évalue le réalisme des expérimentations ainsi obtenues.*

ABSTRACT. *Network experiments are essential for assessing and validating new networking technologies, architectures and protocols. These assessments have long been performed using network simulators. But it clearly appeared that the results got in simulations are not realistic, simulators being unable to accurately integrate all models of all networking components, end host operating systems and applications. Therefore, some work has been issued for developing real experiment platform and network emulators. This paper addresses the motivations that raised the setting-up of such an experimental platform at LAAS - LaasNetExp - and describes its design. It is detailed how experimental conditions can be fully controlled for reproducible and easy to analyze experiments. This paper also describes how realistic conditions can be set-up in experiments (configuration of emulators and traffic generators) based on the results of Internet traffic characterization and analysis. The realism of such experiments is thus assessed.*

MOTS-CLÉS : *Expérimentation en environnement réel, émulation, expérimentations réalistes, expérimentations reproductibles et contrôlées, générateurs de trafic*

KEYWORDS: *Experiments in real environment, emulation, realistic experiments, reproducible and controlled experiments, traffic generator*

1. Motivations

La recherche en réseaux ne peut pas se dissocier d'expérimentations faites soit à l'aide de simulations, soit par émulation ou enfin par expérimentations en environnements réels. Ces expérimentations sont essentielles pour évaluer et valider a priori des architectures et/ou protocoles de communication en cours de conception.

En simulation, tous les composants du système distribué (i.e. les réseaux interconnectés et toutes les machines d'extrémité) sont représentés par des modèles ([FAL 99]) dont le réalisme peut toujours être discuté. De fait, dès que l'on considère le réseau Internet, les simulations deviennent difficiles [FLO 01] notamment à cause de sa taille (nombre d'utilisateurs et d'équipements), de sa complexité (le nombre de protocoles différents par exemple), des comportements très variables de ses utilisateurs, mais aussi des évolutions rapides de ses technologies et usages... On comprend aisément dans pareilles circonstances pourquoi il est si difficile de réaliser des simulations réalistes de l'Internet. Une des limites des simulateurs est leur extensibilité limitée : simuler ne serait-ce qu'une fraction de l'Internet nécessite des machines aux puissances de calcul et capacités mémoire difficilement accessibles.

Même si la simulation est toujours l'outil le plus utilisé dans la Recherche en réseau (notamment pour son faible coût en termes d'investissements matériel et humain - cf. figure 1.a), il est généralement constaté que les résultats sont très imprécis. Les différences entre les résultats de simulations et ceux obtenus en environnement réel sont généralement significatives. Ainsi, des plates-formes d'expérimentation en environnement réel ont aujourd'hui le vent en poupe (comme Planetlab - cf. les figures 1.a et b). Sur de telles plates-formes, tous les composants, les systèmes opératoires, les protocoles, etc. sont réels. Toutefois, le problème lors de l'utilisation de ces plates-formes vient de l'impossibilité de contrôler les conditions expérimentales - le trafic de fond par exemple - ce qui rend l'analyse des résultats délicate voire impossible. Avec Planetlab, par exemple, qui n'est pas complètement monitorée, il est souvent impossible d'analyser les causes (possiblement externes) des problèmes observés sur les protocoles que l'on teste. De plus, ces expérimentations ne sont pas reproductibles par la non maîtrise des conditions expérimentales.

Le compromis adapté pour réaliser des expérimentations réalistes et exploitables paraît être l'émulation de réseaux. En émulation, les machines d'extrémité, leurs OS, applications et les protocoles de bout en bout sont réels. Seul le comportement réseau est simulé. Cette solution permet des expérimentations dans lesquelles la totalité des conditions expérimentales sont contrôlées, tout en minimisant les coûts d'équipements par rapport aux plates-formes en environnement réel. De plus, un émulateur peut être configuré pour se comporter comme un routeur (émulation fine) ou comme un système autonome complet, ce qui permet d'adapter la granularité d'émulation en fonction des besoins, et donc de résoudre les problèmes d'échelle des scénarios expérimentaux.

Les figures 1.a et b illustrent les capacités des différentes familles d'outils pour l'ingénierie réseau à réaliser des expérimentations réalistes, et ce en fonction du coût induit.

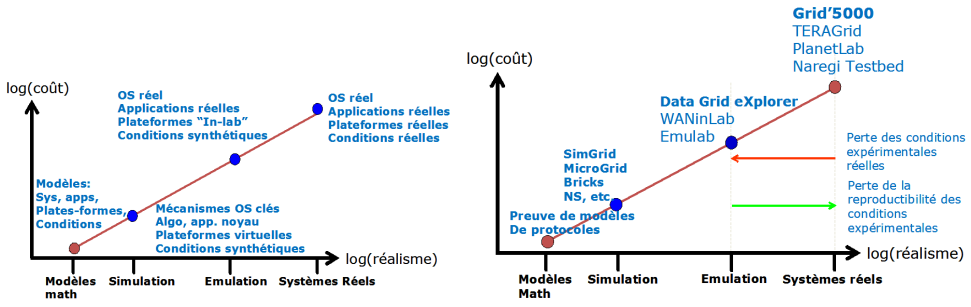

Figure 1. *Les outils d'ingénierie réseau. (a) Niveaux de virtualisation, de réalisme et de coût des différentes familles d'outils. (b) Exemples d'outils dans chacune des familles.*

Dans ce contexte, une plate-forme d'expérimentation a été conçue et installées au LAAS. L'objectif pour cette plate-forme est de répondre au plus grand nombre de demandes, et donc d'être aussi générique que possible. Elle a été conçue pour permettre à la fois de réaliser des expérimentations en émulations réseau, mais aussi en environnement réel. De plus, cette plate-forme permet une utilisation simultanée pour plusieurs expérimentations différentes et reproductibles. Cette plateforme a été baptisée "LaasNetExp" ("LAAS Network Experiments").

Le papier est organisé de la manière suivante : La partie 2 présente en détails les besoins des chercheurs avec cette plateforme expérimentale. La partie 3 décrit comment elle a été conçue et mise en œuvre, et comment elle est gérée. Enfin, dans la partie 4, il est montré comment des conditions expérimentales réalistes (configuration des émulateurs et générateurs de trafics) peuvent être mise en œuvre à partir des résultats de caractérisation et d'analyse du trafic Internet, et évalue le réalisme des expérimentations ainsi obtenues.

2. Besoins expérimentaux

Après avoir présenté les motivations de création de LaasNetExp, cette partie présente les différents atouts de cette dernière. LaasNetExp a été définie pour satisfaire aux quatre besoins principaux :

– **Un contrôle expérimental total** : Dans le but de mener des expérimentations exploitables et reproductibles, il est primordial d'avoir un contrôle total sur les conditions expérimentales, notamment en termes de trafic (charges, propriétés, profils...). Le contrôle total signifie aussi de pouvoir gérer automatiquement des (re-)configurations de la plate-forme sans l'arrêter, surtout lorsque que cette dernière est partagée par plusieurs expérimentations : les expérimentations doivent être indépendantes même quand elles partagent toutes ou parties des ressources.

– **Monitoring et mesure en continu des expérimentations** : LaasNetExp doit fournir toutes les mesures nécessaires sur les expérimentations en vue de leur analyse, évaluation et validation. Le système de monitoring et de mesure doit être transparent, i.e. ne pas influer sur les expérimentations en cours. Il doit de plus être très précis (avec des estampilles de grande précision), fiable (aucun paquet ne doit être manqué par exemple) et capable de stocker et analyser des fichiers de traces de grandes tailles sans perturber le déroulement des expérimentations en cours.

– **Intégration avec d'autres plateformes** : un des problèmes essentiels avec les réseaux aujourd'hui a trait à leur taille. Les problèmes d'échelle peuvent se régler facilement en émulation, en jouant sur la granularité des émulations (un émulateur peut émuler le comportement d'un unique composant réseau, ou d'un réseau tout entier). Par contre, en environnement réel, il est nécessaire de pouvoir disposer de plus de machines. Pour ces raisons, il est nécessaire que LaasNetExp puisse s'interconnecter avec d'autres plates-formes expérimentales, tout en gardant à l'esprit que les conditions expérimentales doivent être contrôlées et monitorées.

– **Isolation du monde extérieur ou entre expérimentations différentes** : pour pouvoir contrôler les conditions expérimentales, il faut s'isoler du monde extérieur, de son trafic et plus spécifiquement de ses anomalies. Notre plate-forme représentant une puissance de calcul et de communication importante, il faut également la protéger des intrusions. L'isoler de l'extérieur garantit un certain niveau de sécurité, mais il doit être renforcé par d'autres mécanismes de sécurité (filtrage par exemple, car nous connaissons à l'avance les caractéristiques des trafics expérimentaux venant de l'extérieur).

3. Description de la plateforme LaasNetExp

La figure 2 présente la plateforme LaasNetExp dans sa globalité. Pour remplir les besoins expérimentaux, LaasNetExp est complètement séparée du réseau opérationnel du LAAS afin d'éviter les perturbations mutuelles. LaasNetExp est reliée à l'Internet directement par RENATER, pour pouvoir profiter des nombreux services IP qui sont offerts par GEANT, RENATER et la plupart des réseaux de la recherche européens. LaasNetExp n'est donc pas connectée au réseau régional Midi-Pyrénées Rémip (ce qui aurait dû être le cas), car Rémip est un réseau commuté de niveau 2 qui n'offre donc pas les services IP dont nous avons besoin (cf. suite de cette partie).

Actuellement, LaasNetExp est composée d'un serveur et de 38 machines d'expérimentation (PC Dell PowerEdge 860, processeur Xeon Core 2, à 2.13 GHz, 2 Go de RAM et disque dur de 600 Go) fonctionnant sous différents OS et équipés de quatre interfaces Ethernet. Pour permettre des expérimentations en émulation et en environnement réel, deux réseaux ont été crées dans LaasNetExp : un réseau public réel de 3 domaines (pour les expérimentations multi-domaines) associé à des adresses IP publiques déclarées comme appartenant à 3 réseaux différents, et un réseau d'émulation. Chaque machine a deux interfaces Ethernet (Eth0 et Eth1) associées aux adresses IP publiques (Eth0 dans le domaine1 et eth1 soit dans le domaine2 soit dans le domaine3)

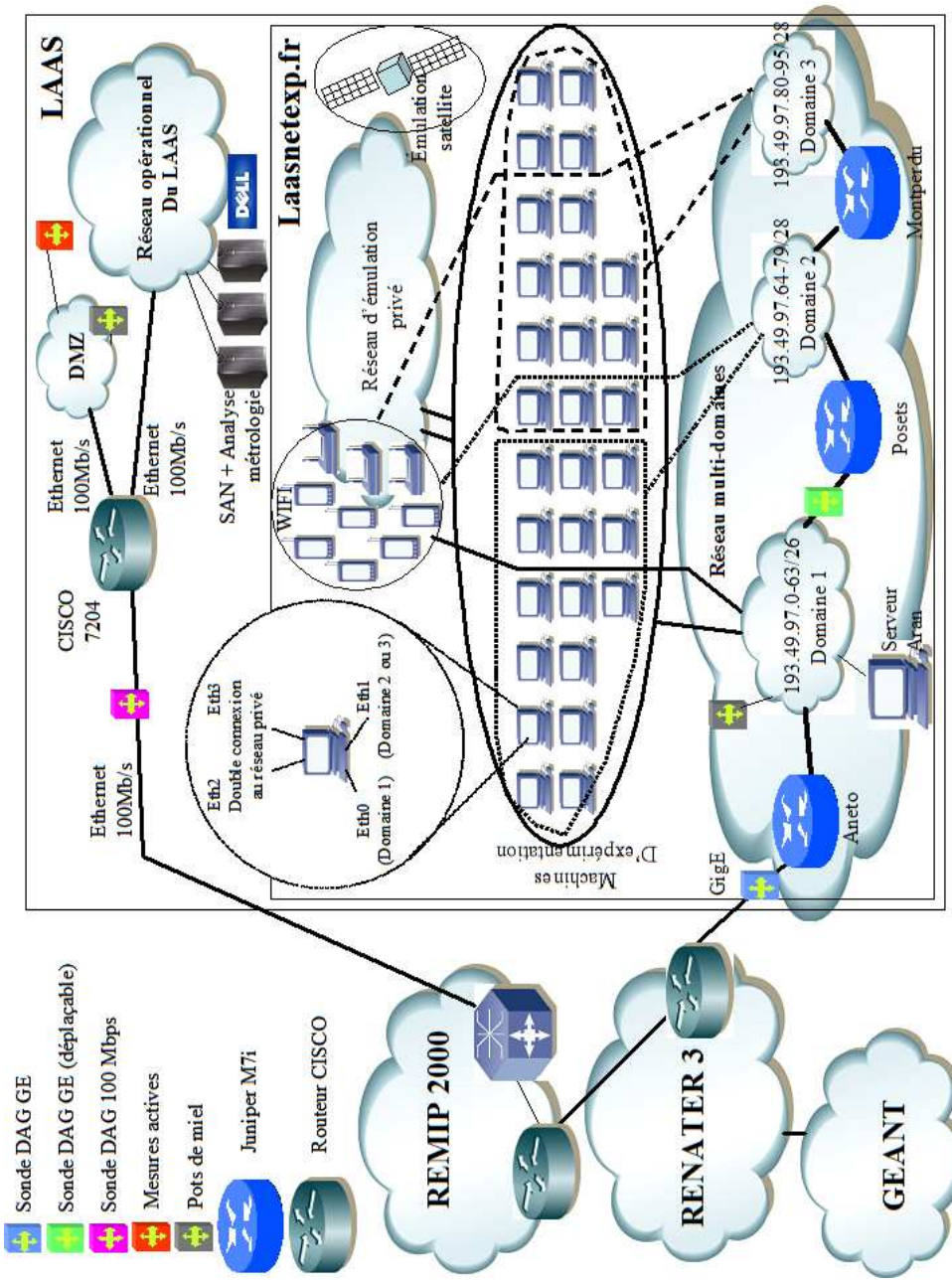

Figure 2. *La plate-forme LaasNetExp*

et les deux autres interfaces Ethernet (Eth2 et Eth3) liées à deux adresses privées dans le réseau d'émulation.

3.1. *Le réseau d'expérimentation réel*

Concernant le réseau réel multi-domaines de LaasNetExp, chaque domaine est implémenté par un commutateur Ethernet Cisco Catalyst 4948-10GE avec 24 ports Gigabits. Les trois domaines sont interconnectés par des routeurs Juniper M7i, nommés *aneto*, *posets* et *montperdu* (cf figure 2). *Aneto* est le routeur d'accès à RENATER, ce qui signifie qu'il est un point sensible et ne subira pas de conditions expérimentales risquées. La suite de cette partie détaille et justifie les choix faits pour LaasNetExp pour satisfaire aux besoins expérimentaux.

Reproductibilité des expérimentations grâce au contrôle du trafic et de la charge des machines. LaasNetExp est isolée du réseau du LAAS. Il ne passe donc sur ce réseau que le trafic généré par nos expérimentations. De plus, on peut renforcer cette propriété grâce au routeur *aneto* - en contact avec l'extérieur - en lui faisant filtrer le trafic entrant potentiel qui ne serait pas lié aux expérimentations en cours. De plus, avec un planning strict des machines, on évite d'avoir des machines utilisées simultanément dans deux expériences. Par conséquent, charge des machines et trafic sur le réseau sont totalement maîtrisés, assurant ainsi la reproductibilité des expérimentations.

Mesures et monitoring. Le système de mesure et monitoring mis en place s'appuie sur la carte DAG d'ENDACE [CLE 00]. Située derrière un splitter électrique ou optique qui laisse passer 80% de la puissance sur le lien normal, et en récupère 20% pour la carte DAG en dérivation, le trafic n'est absolument pas perturbé. Ce système de monitoring est complètement *transparent*.

La carte DAG extrait ensuite en temps réel l'entête de tous les paquets circulant sur le lien, y ajoute une estampille GPS de haute précision (64 bits) et stocke le tout dans un fichier sur le disque dur local. La machine est un serveur Dell PowerEdge 1950, avec un processeur Xeon à 1,6 GHz, 2 Go de RAM et 1To de disques durs. Elle dispose aussi d'un bus PCI étendu de 64 bits de large et de fréquence supérieure à 66 MHz. Avec une telle configuration, il est garanti qu'aucun paquet ne sera manqué. De plus, l'horloge GPS assure une synchronisation de toutes les sondes de capture avec une précision inférieure à 2 μs sur le temps de référence universel. La solution DAG est donc *fiable* et *précise*.

Enfin, un SAN (Storage Area Network) de 4,2 To et 5 serveurs de calcul (Dell PowerEdge 6850, avec 2 processeurs Xeon à 3,2 GHz, 32 Go de RAM et 1,5 To de disques durs) ont été installés sur le réseau opérationnel du LAAS et permettent d'analyser les traces collectées. Naturellement, les transferts entre machines DAG et machines d'analyse se font lorsqu'il n'y a aucune expérimentation en cours.

Un système DAG a également été installé à la sortie du réseau opérationnel du LAAS pour nous permettre de capturer des traces de trafic réel.

Intégration avec d'autres plates-formes. Par rapport aux problèmes d'échelle des expérimentations, il est possible d'interconnecter LaasNetExp avec d'autres plates-formes similaires. Toutefois, pour conserver le contrôle des conditions expérimentales, l'interconnexion se fait par des tunnels PIP (Premium IP) offerts par GEANT et la plupart des réseaux nationaux de la recherche qui garantissent que les paquets PIP seront routés en priorité. Grâce à la bande passante réservée et ce niveau de priorité maximal, il n'y a pas de perte ni de gigue observable sur ce service. PIP donne l'impression d'être seul sur un réseau réservé, et offre donc des conditions reproductibles.

Finalement, PIP cache la complexité des interconnexions de GEANT et des réseaux nationaux de la recherche qui apparaissent comme un unique domaine libre. La structure multi-domaines du réseau expérimental est donc celle de LaasNetExp qui est complètement contrôlée et permet des expérimentations reproductibles.

Isolation des différentes expérimentations entre elles. Pour isoler plusieurs expérimentations en cours simultanément, il faut s'assurer qu'aucune machine n'intervient dans les deux expérimentations : cela se fait par un système de planning de réservation. Ensuite, il ne faut pas que les trafics des 2 expérimentations interagissent. Pour cela, nous utilisons des VLAN qui permettent de créer différentes routes statiques et disjointes. Naturellement, ceci est possible et aisé grâce aux capacités largement surdimensionnées de la plate-forme.

Gestion. Toutes les fonctions de gestion du réseau public de LaasNetExp sont concentrées sur le serveur *aran* : DNS, FTP, Web, comptes des développeurs, etc.

Aran héberge également un serveur PXE. PXE utilise une base de données dans laquelle sont stockées les images des différents OS utilisés par les différentes expérimentations, et que les utilisateurs de LaasNetExp peuvent télécharger et installer automatiquement sur les machines qui leur ont été attribuées pour une expérimentation. Ainsi, chaque fois qu'une nouvelle version ou configuration d'OS est nécessaire pour une nouvelle expérimentation, elle est stockée sur le serveur et donc facilement ré-installable. Grâce à ce serveur, il est possible de facilement mutualiser la plate-forme pour de nombreuses expérimentations tout en ne perdant pas beaucoup de temps en reconfiguration des OS (et des topologies de VLAN au niveau des commutateurs Ethernet).

3.2. *Le réseau d'émulation*

Le réseau d'émulation de LaasNetExp est un réseau privé avec des adresses privées non routables. Il consiste en un routeur CISCO Catalyst 6504 avec 96 ports GigaEthernet, et une matrice de commutation non bloquante. La suite, présente les différents choix faits pour ce réseau d'émulation.

Contrôle expérimental de bout en bout. Par construction, le réseau est privé et son trafic totalement contrôlé. On se réfère également au planning de réservation des différentes machines pour éviter qu'une des machines soit impliquées dans plusieurs expériences. La charge des machines est donc contrôlée aussi. Toutes les conditions expérimentales sont donc sous contrôle, et les expérimentations, par conséquent, reproductibles.

Mesures et monitoring. La même solution DAG que celle décrite précédemment est utilisée.

Intégration avec d'autres plates-formes. Pour l'instant, cet aspect ne s'applique pas à la plate-forme d'émulation qui est entièrement privée. Toutefois, il serait facile d'utiliser une des machines comme un routeur entre les espaces privé et public, et d'utiliser des tunnels PIP pour interconnecter la plate-forme d'émulation avec d'autres plates-formes similaires.

Isolation entre expérimentations simultanées. Pour isoler des émulations différentes (qui utilisent des ensembles disjoints de machines), l'utilisation de VLAN appropriés et les capacités de communication surdimensionnées assurent que les deux expériences ne vont pas nécessiter des ressources communes, et n'ont donc aucune relation de dépendance l'une envers l'autre.

Emulation avec des technologies réseaux spécifiques. Par rapport à nos travaux sur les réseaux satellites, la plate-forme d'émulation est utilisée pour émuler des interconnexions particulières entre des réseaux satellites et terrestres. De même, la plate-forme intègre des périphériques sans fils (mini PC, PDA) et un accès WIFI.

Gestion. Comme pour la partie publique de la plate-forme d'expérimentation, il est possible de dynamiquement configurer les machines et le réseau impliqué dans une expérimentation particulière. Pour l'OS, le serveur PXE est utilisé comme déjà indiqué. Pour la configuration réseau, on utilise des VLAN. Chaque expérimentation définit son ensemble de VLAN pour émuler artificiellement la topologie sur le commutateur CISCO Catalyst 6504. Le script permettant de configurer la topologie (l'ensemble de VLAN) permet alors de facilement et rapidement restaurer l'environnement spécifique d'une expérimentation donnée. Cela permet ainsi d'utiliser simultanément la plate-forme pour plusieurs expérimentations sans perdre beaucoup de temps en reconfiguration.

4. Utilisation de LaasNetExp pour des expérimentations réalistes et reproductibles

Par construction, LaasNetExp offre un environnement dans lequel les conditions expérimentales sont complètement contrôlées et reproductibles. Il ne reste donc plus qu'à faire en sorte que ces conditions soient réalistes, ce qui est essentiel lors du test de n'importe quel protocole ou architecture de communication. De même, il est souvent intéressant de les tester dans des conditions extrêmes pour évaluer leurs limites.

A partir des traces de trafic capturées sur le réseau opérationnel du LAAS ainsi qu'à partir d'autres traces publiques, il est possible d'extraire les distributions caractéristiques pour les délais, les pertes, etc. dans l'Internet. Il est alors possible de configurer les émulateurs pour qu'ils reproduisent ces distributions.

4.1. *Génération de trafic réaliste*

Il reste cependant à générer du trafic réaliste sur le réseau expérimental, et notamment le trafic de fond. Par trafic réaliste, nous voulons dire du trafic qui possède toutes les caractéristiques statistiques du trafic réel, à savoir une variabilité forte, de la dépendance longue (LRD), des structures de corrélations spécifiques, etc. [PAR 00] [WIL 98]. Il a en effet été démontré que ces propriétés avaient un impact négatif sur la qualité de service des réseaux [PAR 97], les pics ou les variations fortes du trafic n'étant pas aisés à gérer pour les protocoles ou les mécanismes réseaux actuels. Ce sont ces "mauvaises" propriétés du trafic qu'il faudra également exagérer pour des évaluations aux limites.

Pour générer un tel trafic, nous avons défini un modèle Gamma-Farima capable de décrire les propriétés de variabilité, de corrélation et de LRD du trafic normal, mais aussi du trafic contenant tous types d'anomalies. De nombreux modèles existaient déjà, mais la plupart utilisent des hypothèses gaussiennes (irréalistes pour le trafic Internet), et les autres sont bien trop complexes pour être facilement utilisables. Notre modèle Gamma-Farima a été présenté dans [SCH 06]. Son originalité et sa force sont sa capacité à modéliser facilement le trafic Internet qui est non gaussien et à dépendance longue. Il se base sur l'utilisation combinée d'une loi Gamma pour modéliser les distributions des lois marginales du trafic et d'un processus Farima pour caractériser sa fonction de corrélation et de LRD. Au final, toute la complexité du trafic peut s'exprimer à l'aide de seulement 5 paramètres :

– α est le paramètre de forme (loi Gamma) du trafic

– β est le paramètre d'amplitude (loi Gamma) du trafic

– d exprime la LRD (Farima)

– ϕ et θ expriment la dépendance courte (Farima)

Pour les lecteurs intéressés, le modèle $Gamma(\alpha, \beta) - Farima(\phi, d, \theta)$ et sa validation sur de nombreuses traces de trafic sont présentés dans [SCH 06].

Un générateur de trafic GFtg (pour Gamma-Farima traffic generator) a été mis au point à partir de ce modèle. Il se compose de deux outils :

– Le générateur de séries Gamma-Farina, qui à partir des 5 paramètres du modèle, génère une série temporelle indiquant le nombre d'octets ou de paquets qui doivent être générés par intervalles de temps δ ;

– L'injecteur qui injecte dans le réseau le trafic comme indiqué dans la série temporelle produite par l'outil précédent. Pour respecter les temps entre deux paquets

consécutifs générés par l'émetteur et éviter les interactions avec l'environnement réseau, le trafic est généré en utilisant UDP.

4.2. *Validation du générateur de trafic Gamma-Farima*

Méthodologie

GFtg doit générer du trafic réaliste en point à point. Il est donc seulement nécessaire de valider GFtg localement, i.e. vérifier que le trafic en sortie de la carte réseau respecte les caractéristiques souhaitées.

La validation de GFtg a été réalisée sur LaasNetExp. Un PC standard a été utilisé pour générer le trafic, ainsi qu'un système de capture de trafic DAG, connecté sur le lien entre le PC et le commutateur Ethernet.

GFtg a été validé sur plusieurs traces : des traces publiques (Auckland, NLANR, etc.) et d'autres collectées dans le cadre du projet MétroSec (ces traces sont des traces de trafic normal, mais également des traces contenant des anomalies parfois légitimes - comme des foules subites - ou illégitimes comme des attaques DoS). Le tableau 1 liste les traces utilisées dans la validation de GFtg. Il indique également le type d'anomalies générées dans le cadre de MétroSec [SCH 07].

Foules subites. Dans le cadre de MétroSec, des expériences de foules subites sur un serveur Web ont été conduites. Pour les réaliser tout en ayant une activité la plus réaliste possible, la foule subite n'a pas été générée par des automates, mais avec de vrais utilisateurs volontaires. La cible était le serveur web du LAAS qui a été consulté de façon intensive pendant une certaine période de temps. Toutefois, la façon de naviguer de chaque participant relevait de sa propre initiative.

Attaques DoS. Les attaques de DDoS ont été réalisées en utilisant soit Iperf soit Trinoo (sur des machines Linux) pour générer des flux UDP avec des débits différents. Par rapport à Iperf, Trinoo utilise un démon installé sur chaque site participant à l'attaque et permet de créer des attaques plus complexes et réalistes. Les scénarios mis en œuvre comportent 4 sites d'attaque (Mont de Marsan, Lyon, Nice et Paris) qui "bombardaient" une machine du LAAS à Toulouse, où le trafic a été collecté. Le trafic lié à cette attaque a été transmis par l'intermédiaire du réseau RENATER. L'utilisation de plusieurs outils a permis d'évoluer d'attaques de DDoS simples avec Iperf (qui n'est pas un logiciel d'attaque) vers des scénarios plus complexes dans lesquels le type des paquets, le débit, la durée, l'intensité des flots d'attaque, la taille des paquets, le débit d'émission et bien d'autres paramètres ont pu être configurés selon nos souhaits.

Pour plus d'informations sur la production de ces traces, le lecteur pourra se rapporter à [BOR 08].

Pour valider GFtg, et pour chaque trace :

– les 5 paramètres Gamma-Farina ont été calculés sur la trace ;

Données	T (s)	# Pkts	IAT (ms)	Répertoire
PAUG	2620	1	2.6	ita.ee.lbl.gov/index.html
LBL-TCP-3	7200	1.7	4	ita.ee.lbl.gov/index.html
AUCK-IV	10800	9	1.2	wand.cs.waikato.ac.nz/wand/wits
CAIDA	600	65	0.01	www.caida.org/analysis/workload/oc48/
UNC	3600	4.6	0.8	www-dirt.cs.unc.edu/ts/
METROSEC-ref	5000	3.9	1.5	www.laas.fr/METROSEC/
METROSEC-DDoS	9000	6.9	1.3	www.laas.fr/METROSEC/
METROSEC-FC	1800	3.7	0.48	www.laas.fr/METROSEC/

Tableau 1. Exemples de traces utilisées pour valider GFtg. *Paramètres généraux des traces étudiées. T est la durée de la trace, en secondes. # Pkts (10^6) est le nombre de paquets dans la trace, en millions. IAT est le temps d'inter-arrivées moyen, en ms.*

– GFtg a été utilisé pour générer le trafic correspondant à ces 5 paramètres, puis ce trafic a été capturé par le système DAG, produisant ainsi la trace de trafic rejoué ;

– Les 5 paramètres Gamma-Farina ont été calculés sur la trace rejouée et comparés à ceux de la trace originale.

Résultats de validation

Les tableaux 2 et 3 montrent pour deux exemples de traces prises parmi celles décrite dans le tableau 1 (ces deux traces ont été choisies car elles ont des débits de paquets très différents) les différences entre les 5 paramètres Gamma-Farima mesurés sur les traces originales et rejouées.

Paramètres	α	β	d	ϕ	θ
trace originale	2.56	2.40	0.222	0.407	0.172
trace rejouée	2.60	2.36	0.225	0.405	0.160

Tableau 2. *Comparaison des paramètres Gamma-Farima entre une trace originale et cette même trace rejouée avec GFtg. Le débit est de 6 paquets/ms*

Paramètres	α	β	d	ϕ	θ
trace originale	26.69	1.36	0.263	0.015	0.274
trace rejouée	29.52	1.23	0.283	0.072	0.343

Tableau 3. *Comparaison des paramètres Gamma-Farima entre une trace originale et cette même trace rejouée avec GFtg. Le débit est de 26 paquets/ms*

Il apparaît dans ces deux tableaux que la précision de l'injecteur est très bonne. Ceci a été confirmé sur toutes les traces que nous avons rejouées jusqu'à maintenant avec GFtg.

5. Conclusion

Cet article a décrit la conception et la mise en œuvre d'une plate-forme expérimentale pour conduire des recherches en réseaux. Les objectifs en termes de contrôle des conditions expérimentales pour des expérimentations reproductibles et faciles à analyser ont été atteints. L'originalité de cette plate-forme réside dans sa capacité à offrir simultanément un réseau émulé et un environnement réseau réel.

Cet article propose également des recettes méthodologiques et des outils pour des scénarios expérimentaux réalistes. Cette méthodologie exploite les résultats de caractérisation et de modélisation du trafic Internet pour configurer les émulateurs, mais aussi pour générer du trafic réaliste (des générateurs de trafic ont été spécifiquement développés). La validation de cette plate-forme et des méthodologies associées ont confirmé le réalisme des conditions expérimentales, ce qui fait de LaasNetExp un outil vraiment adapté aux expérimentations réseaux.

6. Bibliographie

[BOR 08] BORGNAT P., ABRY P., DEWAELE G., SCHERRER A., LARRIEU N., OWEZARSKI P., LABIT Y., GALLON L. AUSSIBAL J., « Une caractérisation non gaussienne et à longue mémoire du trafic Internet et de ses anomalies : validation expérimentale et application à la détection d'attaque de DDoS », *Annales des télécommunications*, 2008.

[CLE 00] CLEARY J., DONNELLY S., GRAHAM I., MCGREGOR A., PEARSON M., « Design principles for accurate passive measurement », *Proceedings of PAM'2000*, Hamilton, New Zealand, avril 2000.

[FAL 99] FALL K., « Network emulation in the Vint/NS simulator », *proceedings of ISCC'99*, July 1999.

[FLO 01] FLOYD S., PAXSON V., « Difficulties in Simulating the Internet », *IEEE/ACM Transactions on Networking*, vol. 9 (4), 2001, p. 392-403.

[PAR 97] PARK K., KIM G., CROVELLA M., « On the Effect of Traffic Self-similarity on Network Performance », *SPIE International Conference on Performance and Control of Network Systems*, novembre 1997.

[PAR 00] PARK K., WILLINGER W., « Self-similar network traffic : an overview », *Self-similar network traffic and performance evaluation, Park K. and Willinger W. eds., J. Wiley and Sons*, 2000.

[SCH 06] SCHERRER A., LARRIEU N., BORGNAT P., OWEZARSKI P., ABRY P., « Non Gaussian and Long Memory Statistical Modeling of Internet Traffic », *4th International Workshop on Internet Performance, Simulation, Monitoring and Measurements (IPS-MoMe'2006)*, Salzburg, Austria, février 2006.

[SCH 07] SCHERRER A., LARRIEU N., OWEZARSKI P., BORGNAT P., ABRY P., « Non Gaussian and long memory statistical characterization for Internet traffic with anomalies », *IEEE Transaction on Dependable and Secure Computing*, vol. 4(1), January-March 2007.

[WIL 98] WILLINGER W., PAXSON V., TAQQU M., « Self-Similarity and Heavy Tails : Structural Modeling of Network traffic », *A Practical Guide To Heavy Tails : Statistical Techniques and Applications, ISBN 0-8176-3951-9*, 1998.