Projet industriel - Soutenance finale

Comportement d'une architecture Dual-Stack Lite

Isabelle Kraemer, Frédéric Perrin
Encadrants : Tanguy Ropitault, Laurent Toutain
Telecom Bretagne
18 mars 2011

Lignes directrices

Lignes directrices

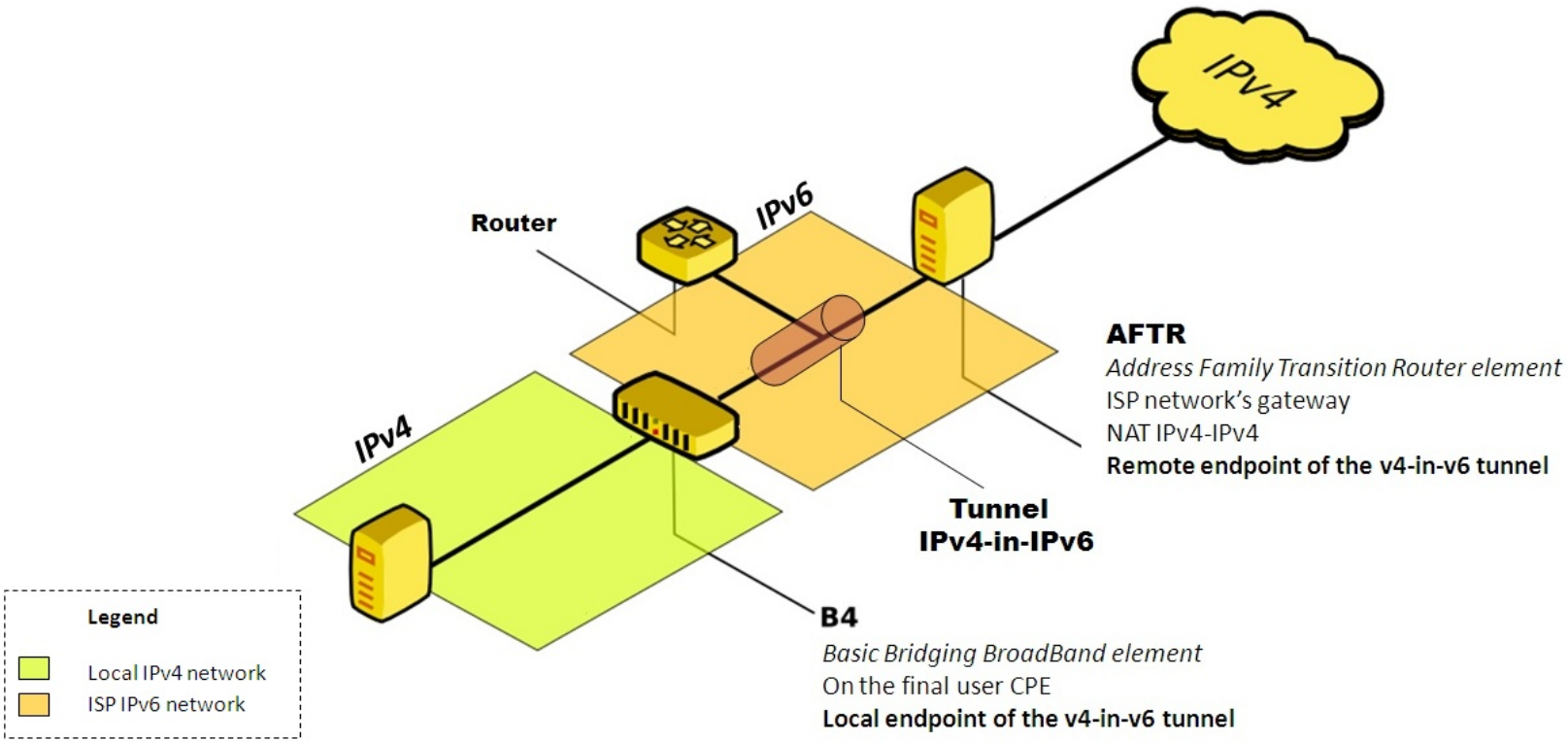
Introduction

- IPv4/IPv6 : Une cohabitation nécessaire
  - Épuisement des adresses IPv4 depuis février 2011
  - Étude de solutions *transitoires* pour une *migration douce*
  - Exemple : L'architecture Dual-Stack Lite
- Le projet
  - Preuve de concept sur quelques machines du ResEI
  - Continuité des travaux de l'année dernière de Florent Fourcot et Bertrand Grelot
  - Étude du comportement en cas d'insuffisance de ports

Lignes directrices

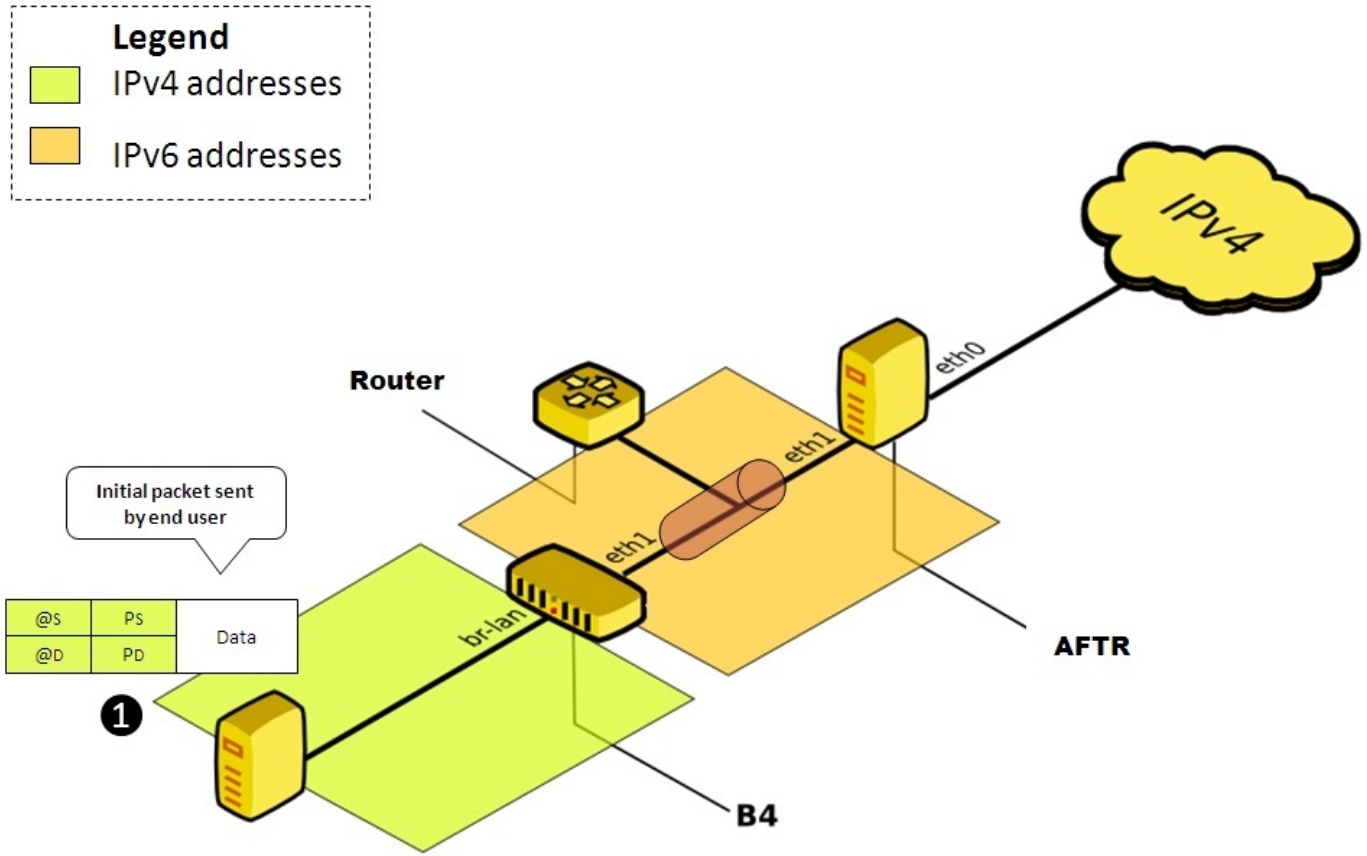
L'architecture Dual Stack Lite

- Réseau de l'opérateur **IPv6**, réseau local **IPv4**
- Un **tunnel IPv4-in-IPv6** pour assurer la connectivité IPv4
- Partage d'une adresse IPv4 par plusieurs utilisateurs ⇒ **NAT**
- Des **équipements spécifiques**

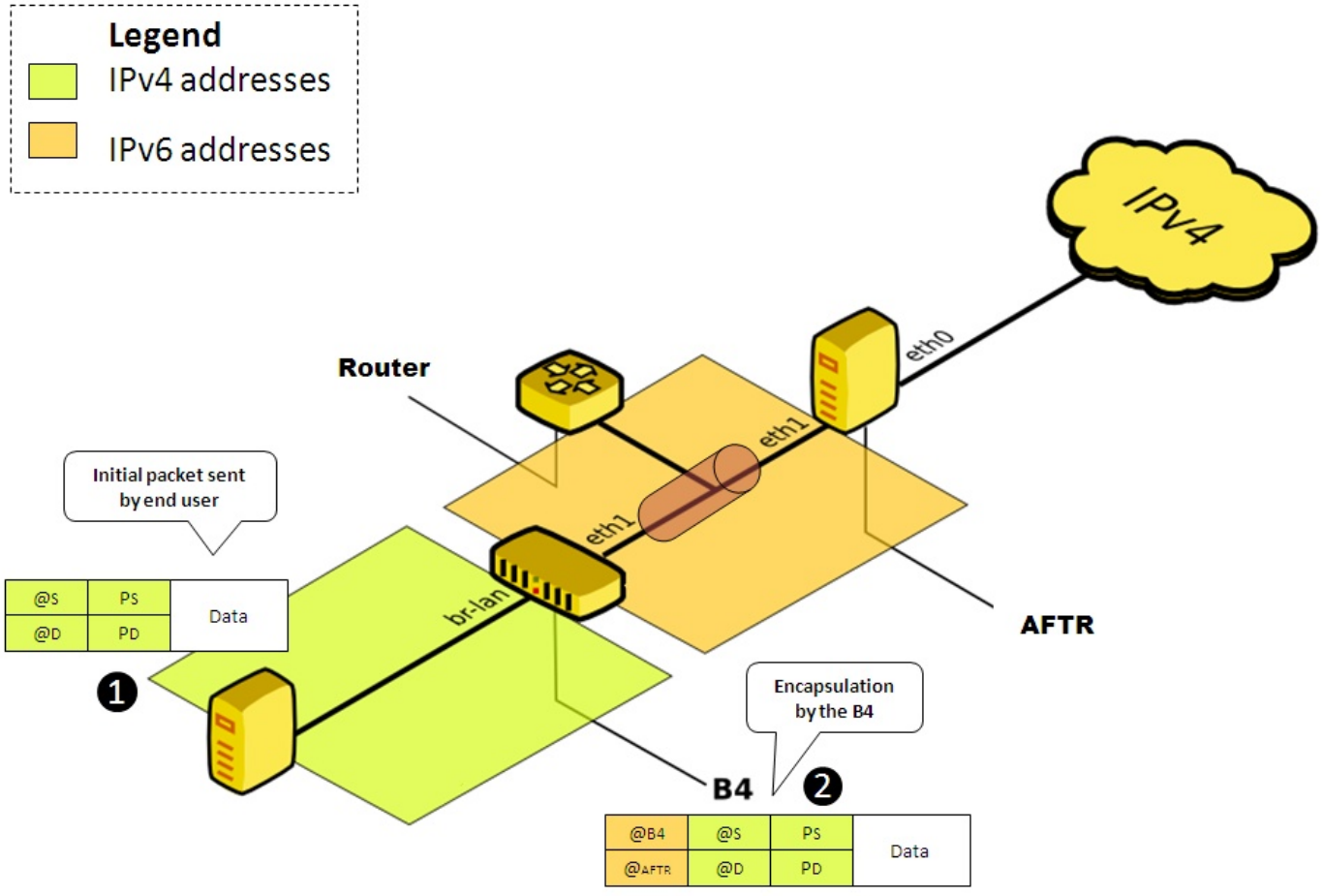


TELECOM Bretagne

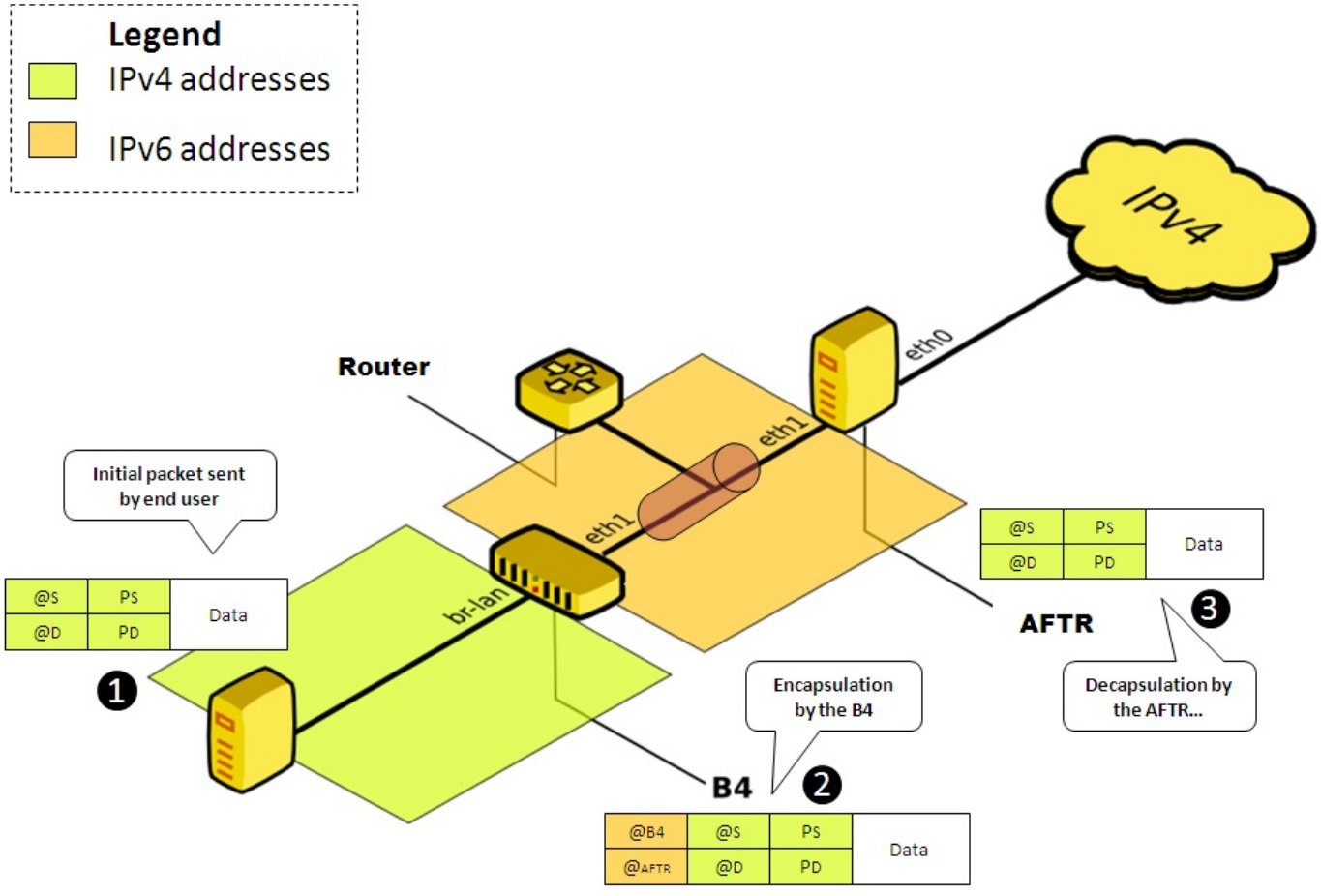
L'architecture Dual Stack Lite



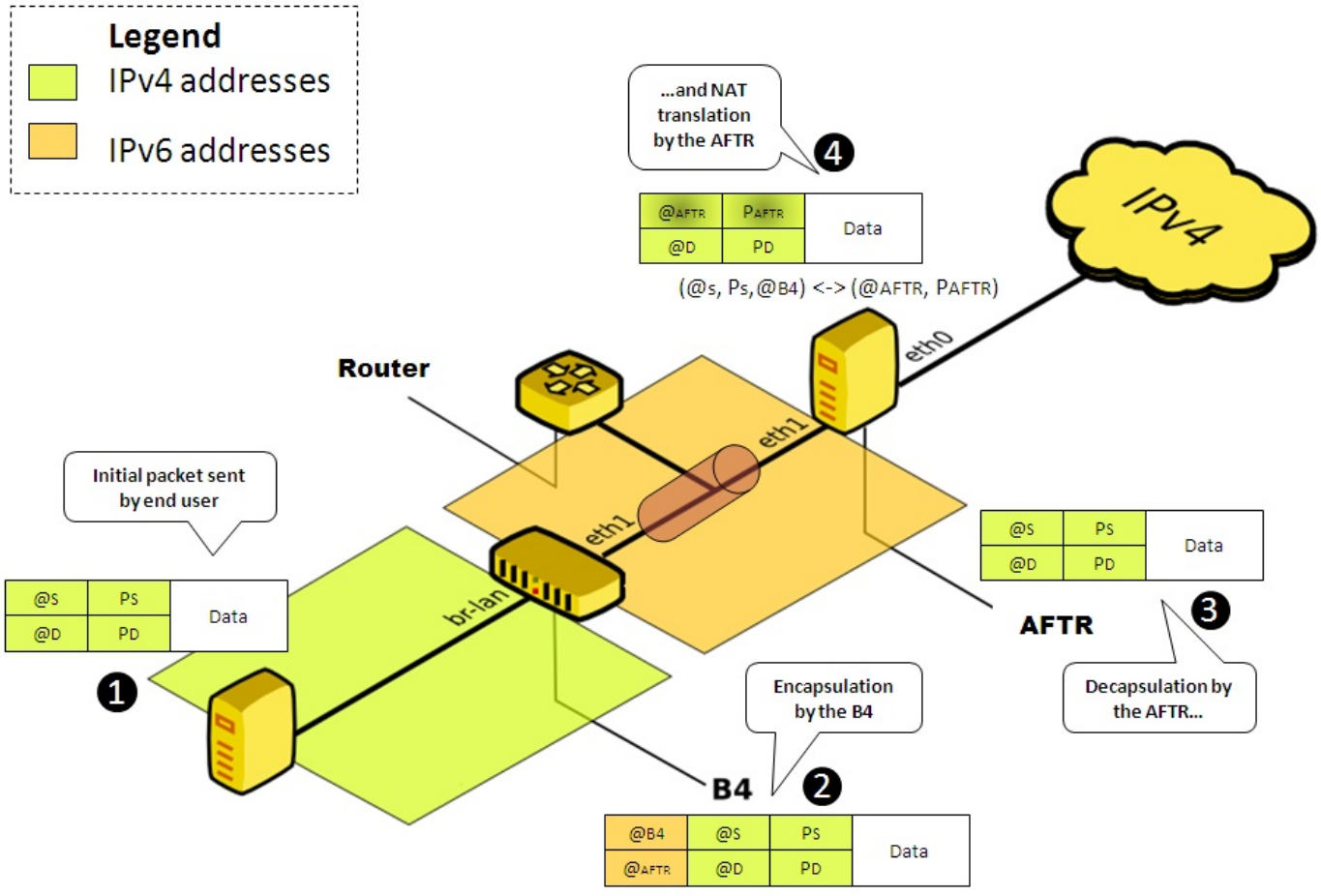
L'architecture Dual Stack Lite



L'architecture Dual Stack Lite



L'architecture Dual Stack Lite

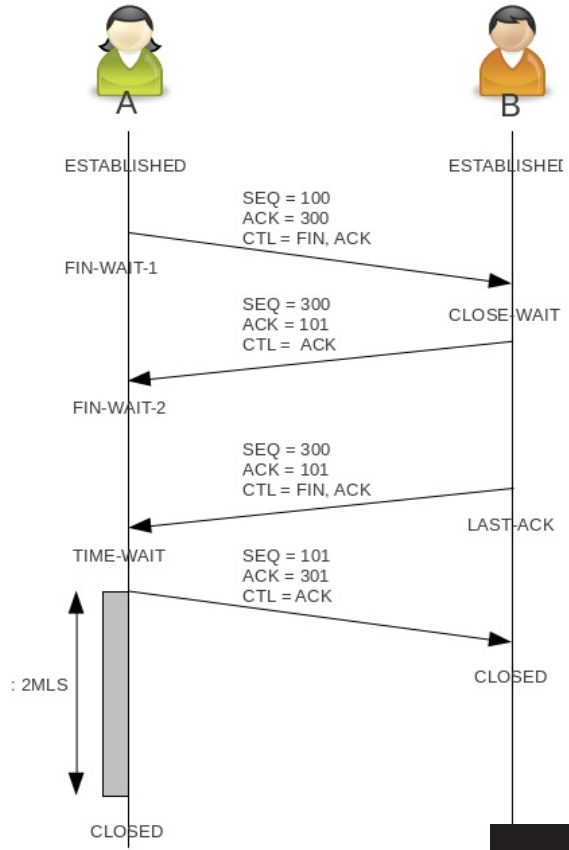


TELECOM Bretagne

Consommation de ports

Exemple : un téléchargement d'image (300 ms)

- Cas des connexions TCP
  - Fermeture simplex
  - Malgré la fermeture de la connexion, il peut y avoir des segments encore en transit
  - L'acquittement du FIN n'est pas acquitté : on n'est pas sûr qu'il arrive
  - Une fois la connexion fermée, on ne peut pas réutiliser le port → état `TIME_WAIT`
  - (Linux : `TIME_WAIT` = 120 s)



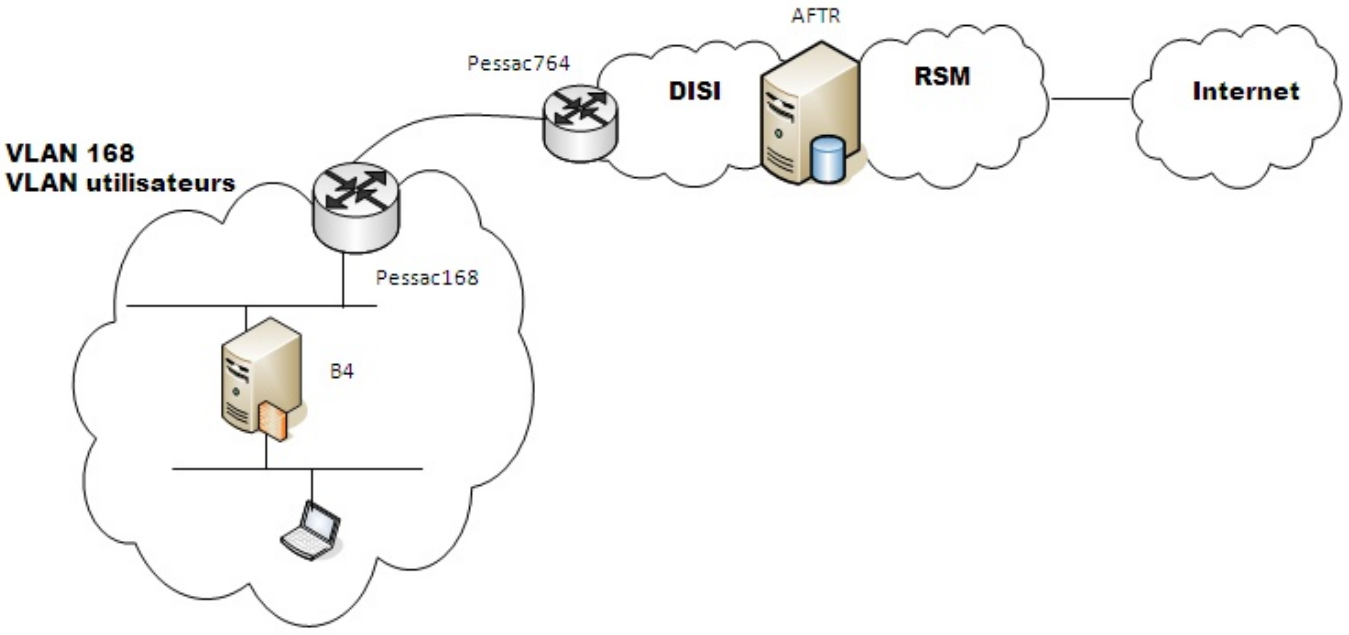
Consommation de ports

- NAT
  - *Problème* : router correctement les paquets en retard
  - *Solution* : Lorsque le client établit la connexion, création d'un contexte : @IPv4+port client, **@IPv6 B4**, @IPv4+port AFTR, @IPv4+port site distant
  - Durée du contexte
    - 120 s si TCP est utilisé
    - 300 s si UDP est utilisé
  - `default_hold_lifetime` = 120 s
  - Allocation des ports par paquet (bucket) pour pouvoir grouper les entrées dans le journal (trace) et en diminuer la taille
- **Conclusion**
  - **Ports non immédiatement réutilisables**
  - **Problématique du dimensionnement du nombre de ports par utilisateurs**
  - **Importance de la gestion de leur attribution**

Lignes directrices

Caractéristiques de notre déploiement

- Adaptation de l'architecture DS Lite à notre environnement
  - Client final et B4 au ResEI
  - AFTR au laboratoire RSM
  - Routage du trafic entre l'AFTR et le B4
  - Mise en place localement des infrastructures nécessaires



Lignes directrices

Protocole de test

- **Constat** : dégradation nette de l'expérience utilisateur en situation de manque de ports.
- **But** : évaluer la vitesse de dégradation en fonction de la famine de ports. Le temps d'affichage d'une page est un paramètre pertinent.
- **Outil** : script Perl simulant une navigation aléatoire
  1. Choix d'une adresse de départ (input) et initialisation d'un pool d'adresses vide
  2. Téléchargement des pages et des éléments `img`, `script`, `link`
  3. Ajout des liens rencontrés au pool d'adresses
  4. Attente pendant un certain temps ($15\text{s} \times \log(1/\text{rand})$)
  5. Sélection d'une adresse du pool
  6. Retour au point 2

Protocole de test

- Caractéristiques du script
  - Nombre de threads parallèles : 15 (paramètre par défaut de `firefox`)
  - Durée : 1 heure
  - Seul le temps d'établissement de la connexion est pertinent
    - Téléchargement de seulement un octet de chaque élément
    - Requêtes DNS effectuées auparavant
- URLs de départ
  - `planet.debian.org`, `twitter.com` et `del.icio.us`
  - Web 2.0 : beaucoup de liens externes
- Reproductibilité
  - Enregistrement des pages visitées et des temps de pause la première fois
  - Rejeu des enregistrements
  - Mesures effectuées avec trois clients simultanés

Lignes directrices

Caractéristiques des parcours suivis

- Client directement relié à Internet

Point de départ	Planet Debian	Twitter	del.icio.us
Nombre de pages visitées	196	218	207
Nombre total d'objets HTTP récupérés	3967	7412	2472
Nombre moyen d'éléments récupérés	20.2	34.0	11.9
Nombre typique d'éléments par page	$\emptyset$	11 et 57	10
Erreurs 404	16	18	7
Timeout serveur et DNS non resolvable	15	2	0
Autres erreurs	28	10	13
Noms de domaine différents parcourus	202	143	38
Nombre de pages sous le nom de domaine initial	13 (dont `*.debian.org`)	123	187 (dont `delicious.com`)
Requêtes HTTPS	190	65	155

TABLE: Caractéristiques d'une session de navigation

Distribution du nombre d'éléments par page

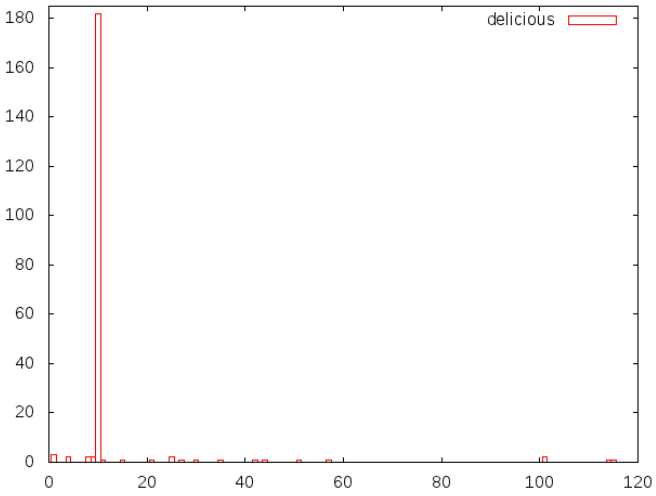


FIGURE: del.icio.us

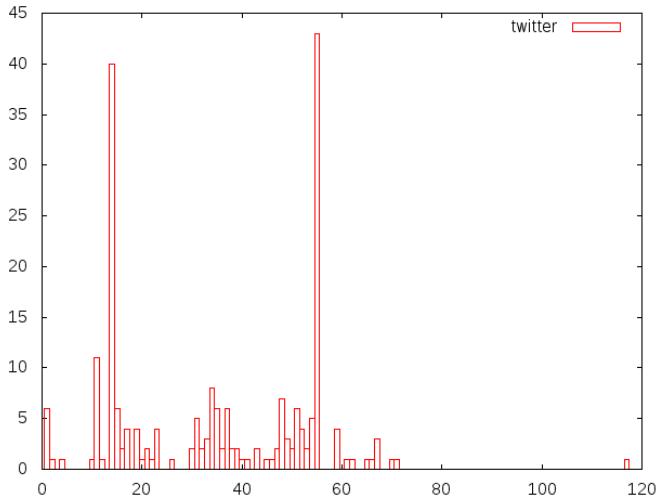


FIGURE: twitter.com

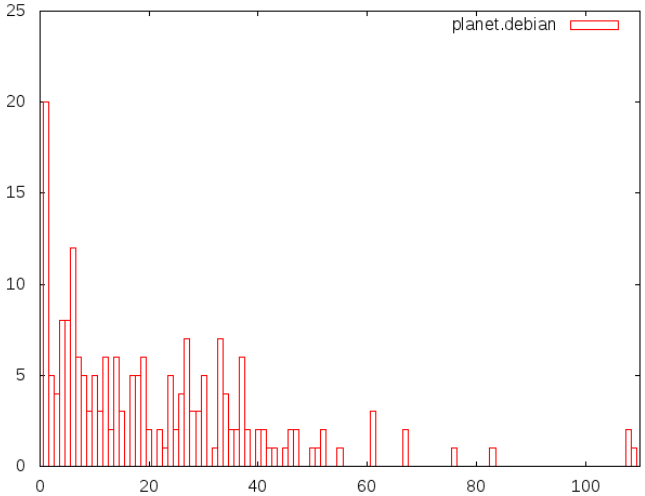


FIGURE: planet.debian.org

Temps d'affichage en situation normale

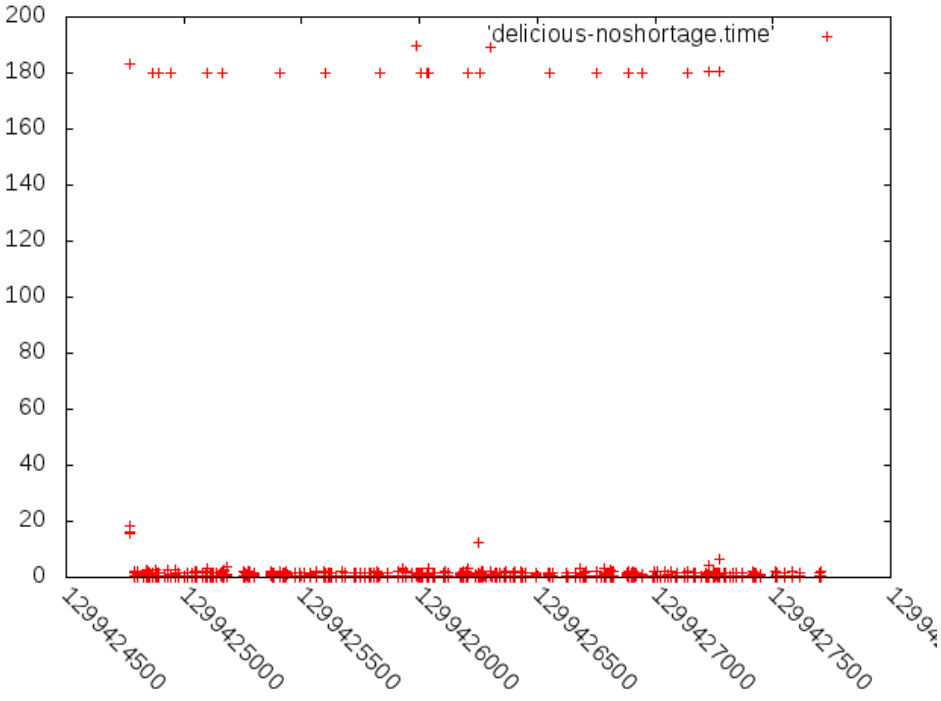


FIGURE: Surf sur del.icio.us

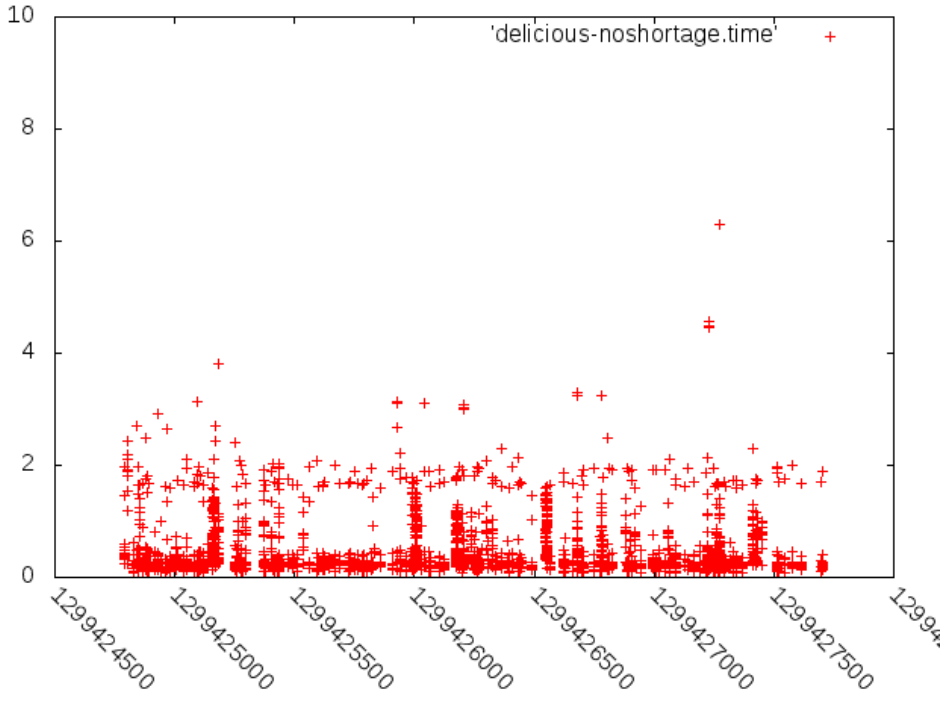
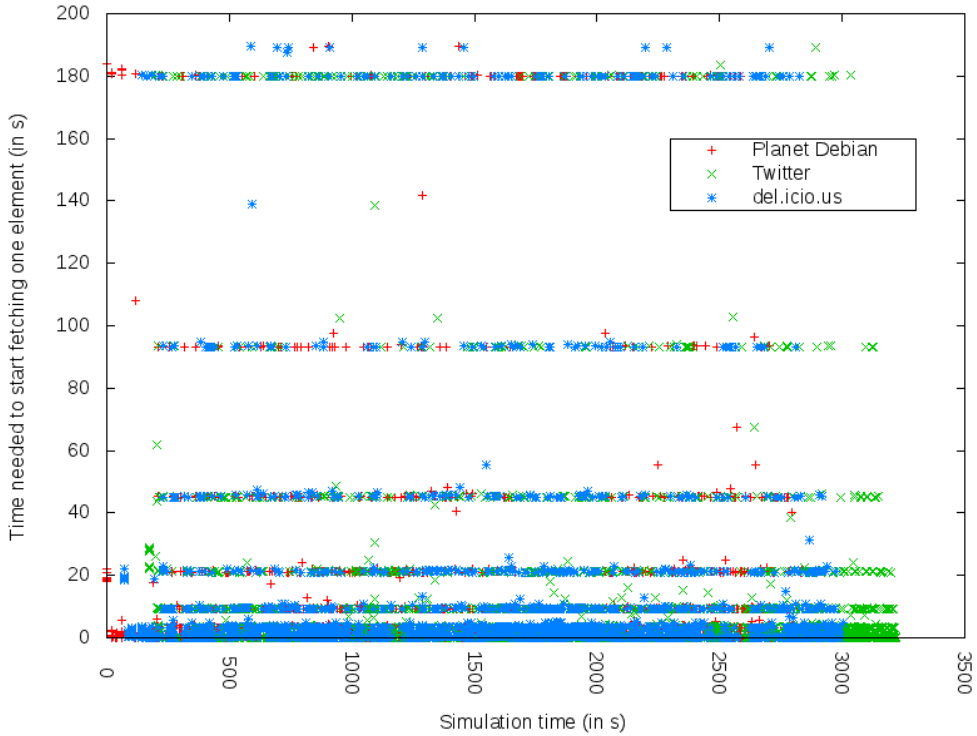


FIGURE: Agrandissement : $y \in [0,10]$s

- Temps moyen de récupération des éléments inférieur à 2s
- Limite à 180 s : timeout de notre UserAgent `LWP::UserAgent`

Première évaluation de l'expérience utilisateur

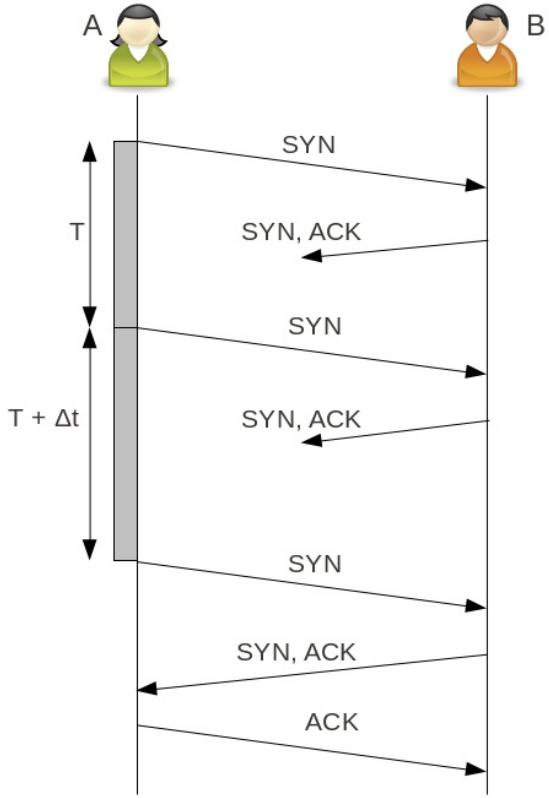
- Réduction artificielle du nombre de ports disponibles
- Temps d'affichage plus long
- Observation de paliers :
  - 3 s
  - 9 s
  - 21 s
  - 45 s
  - 93 s
  - 180 s (timeout)



Temps nécessaire pour récupérer les éléments
500 ports disponibles, 3 clients

Interprétation

- Ouverture d'une connexion TCP
- Three ways handshake
  - Émission du SYN
  - Armement d'un timer
  - Expiration du timer
  - Renvoi du SYN
  - Armement d'un timer de durée supérieure
  - Etc ...
- Les connexions s'établissent donc à chaque expiration de timer. Sous Linux :
  - net/tcp_timer.c :retransmits_timed_out()
  - Timer du n*ième* paquet : $T_n = (2^n - 1) \times 3$ secondes
  - Correspond aux valeurs observées



Vitesse de dégradation de l'expérience utilisateur

Plus la famine de ports est prononcée, plus le nombre de retransmissions nécessaires est grand.

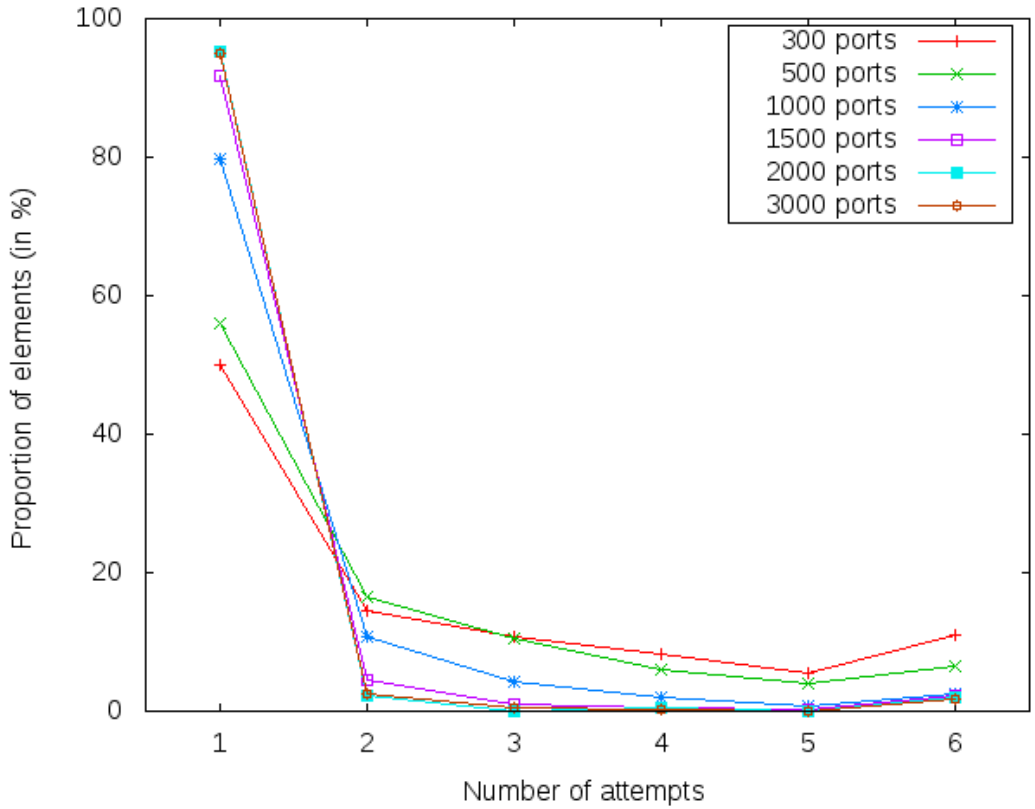


FIGURE: Planet.debian.org

Vitesse de dégradation de l'expérience utilisateur

- Tolérance de l'utilisateur à 5s
- $\Rightarrow$ Abaque aidant au dimensionnement du système

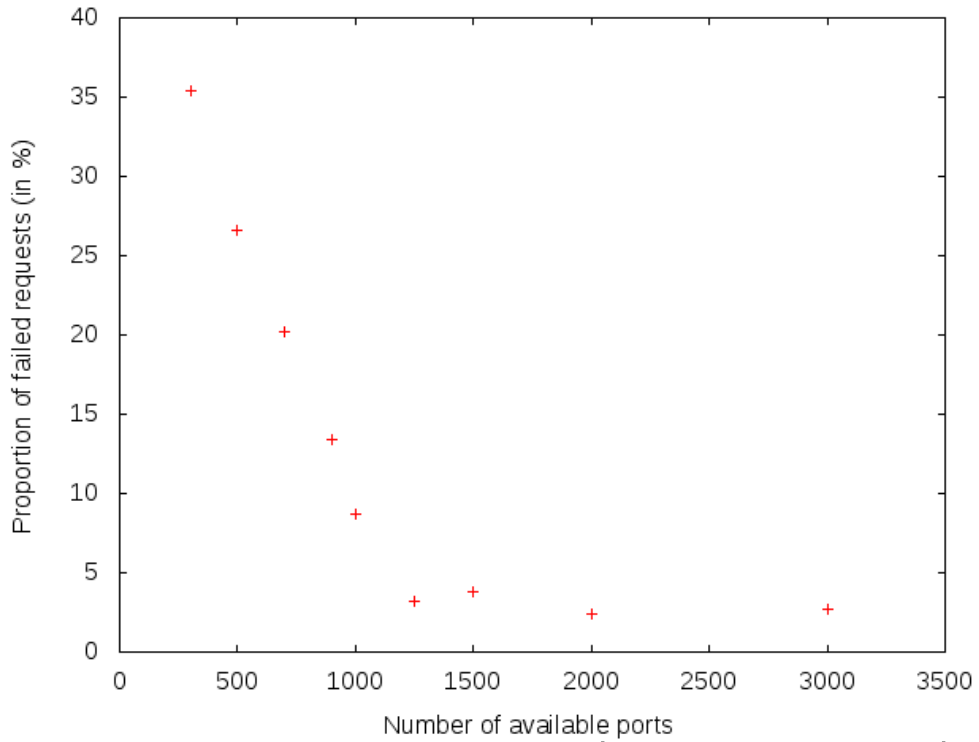


FIGURE: Proportion d'échecs ($\tau = 5$ seconds)

Vitesse de dégradation de l'expérience utilisateur

- Comparaison entre différents seuils acceptables

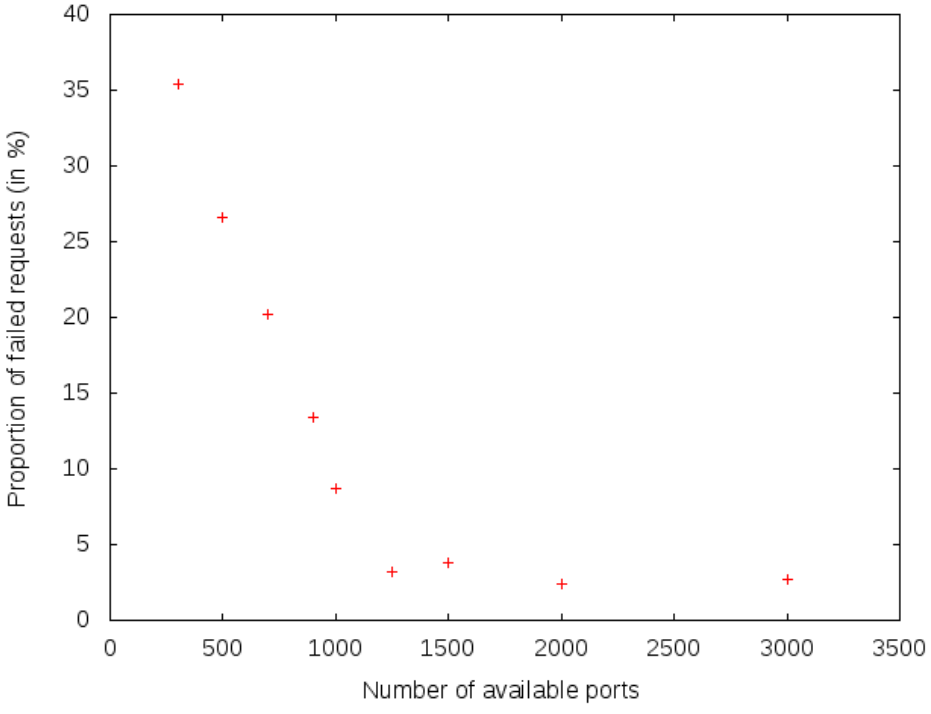


FIGURE: $\tau = 5$ seconds

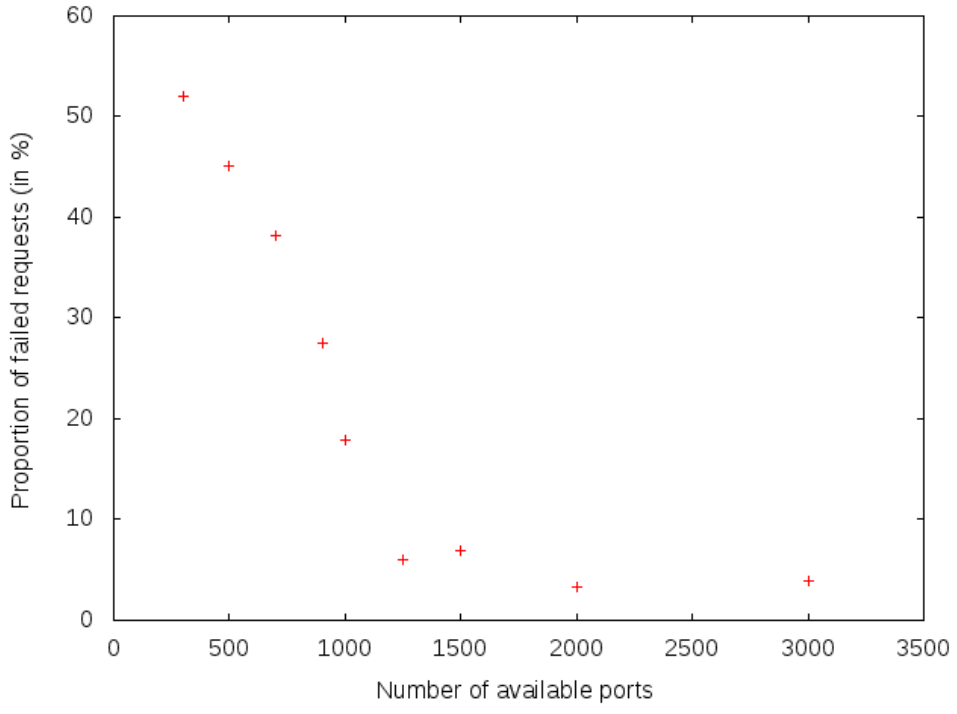


FIGURE: $\tau = 2{,}9$ seconds

Lignes directrices

Conclusion

- Comment évolue le temps de réponse en fonction du nombre de ports disponibles ?
  - Moins il y a de ports disponibles, plus le temps de réponse est élevé
  - On observe des paliers expliqués par les timers TCP
- Cette étude donne une première idée d'un dimensionnement
  - Au moins 2000 ports pour 3 clients
  - Au moins 700 ports par client
- DS-Lite gère difficilement le manque de ports
  - Des ports libres mais inutilisables (réservés)
  - Augmentation du timer entre chaque essai
- L'adoption d'IPv6 permettrait de s'affranchir de ces contraintes

Questions ?

Merci pour votre attention !