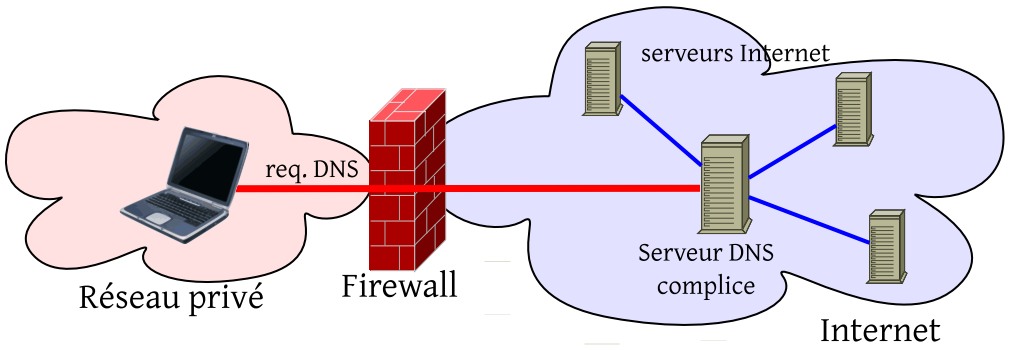
Prototype de Canal Caché dans le DNS

Objectifs

- comprendre le fonctionnement des canaux cachés
- évaluer les implémentations existantes
- implémenter un prototype et l'évaluer

Principe



Implantations existantes

TUN over DNS

Utilisent TUN/TAP, fournissent une connectivité IP

NSTX :
- encodage base64 (enregistrements non-conforme à DNS - utilisation de "_")
- utilise des enregistrements TXT
- découpe un paquet IP en plusieurs paquets DNS
- bugs (segfaults, memleaks)

Iodine :
- encodage base32 ou base64
- utilise des enregistrements NULL et EDNS0
- découpe un paquet IP en plusieurs paquets DNS

TCP over DNS

Encapsulent un/des flux TCP dans des paquets DNS
Utilisations plus complexes avec SSH + proxy SOCKS

Nécessitent d'implémenter une machine à états pour réordonner ou retransmettre les paquets

OzymanDNS :
- utilise des enregistrements TXT et EDNS0
- plantages très fréquents

dns2tcp :
- utilise des enregistrements TXT
- encodage base64 non conforme ("/")

Exemple

1 ping initié côté serveur, avec TUNS

```
client -> serveur DNS Standard query CNAME d668.send.domain
serveur -> client DNS Standard query response CNAME IUAAAVAAABA[...]
  EJBGFAVCYLRQGI2DMOB2HQ7EAQSEIZEEUTCOKBJFIVSYLJOF4YDCM.RTG
  Q2TMNY0.send.domain
client -> serveur DNS Standard query CNAME IUAAAVBNP4AAAQABM[...]
  AAAAAAQCEJBGFAVCYLRQGI2DMOB2HQ7EAQSEIZEEUTCOKBJFIVSYL
  JOF4YDCM.RTGQ2TMNY0.recv.domain
serveur -> client DNS Standard query response CNAME length0.recv.domain
```

http://www-id.imag.fr/~nussbaum/tuns.php

TUNS : notre prototype

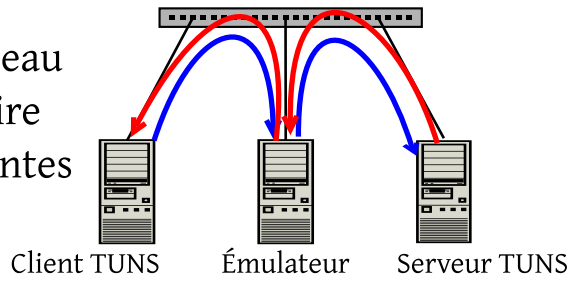
Encapsule des paquets IP dans des requêtes DNS
Utilise uniquement des enregistrements CNAME
Utilise un encodage Base32 pour l'émission et la réception
Ne découpe pas les paquets IP. réduit le MTU. (à 140 octets pour un nom de domaine court)
Écrit en Ruby
Conforme à la RFC 1035 (DNS)

Envoi de données (client -> serveur) :
Dès que des données arrivent : encapsulation + envoi

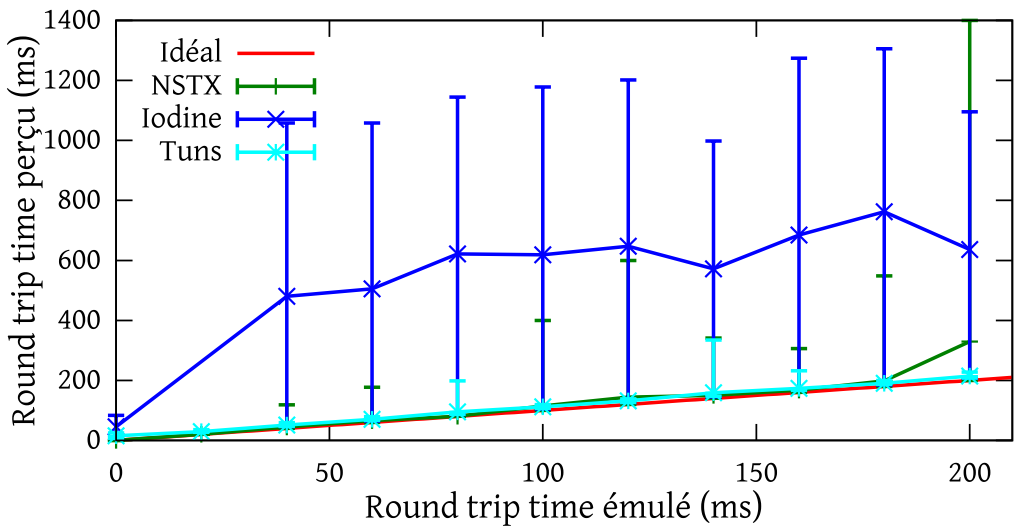
Réception de données (serveur -> client) :
- Sondage régulier par le client
- Le serveur ne répond pas immédiatement s'il n'a rien à envoyer

Évaluations

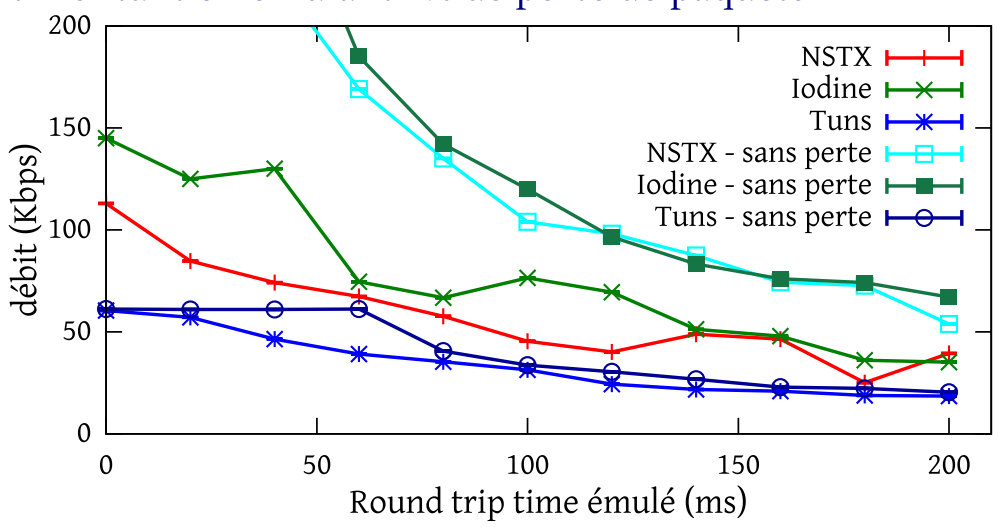
Utilisation d'un émulateur réseau (Linux TC+netem) pour produire des conditions réseaux différentes sur 3 machines de Grid'5000



Latence lors d'un ping initié par le serveur



Débit montant en émulant 2% de perte de paquets



Conclusion

IP over DNS, ça marche !

TUNS :
- implémentation simple (700 LOC) et efficace
- conforme au protocole DNS

NSTX et Iodine : meilleures performances
- facteur limitant pour TUNS : bibliothèque Ruby pour DNS
- mais découper les paquets IP est moins efficace sur un réseau imparfait (avec des pertes de paquets)

Lucas Nussbaum
Olivier Richard
Équipe MESCAL
Laboratoire d'Informatique de Grenoble