Open DynDNS

Projet innovante RICM4 2013-2014

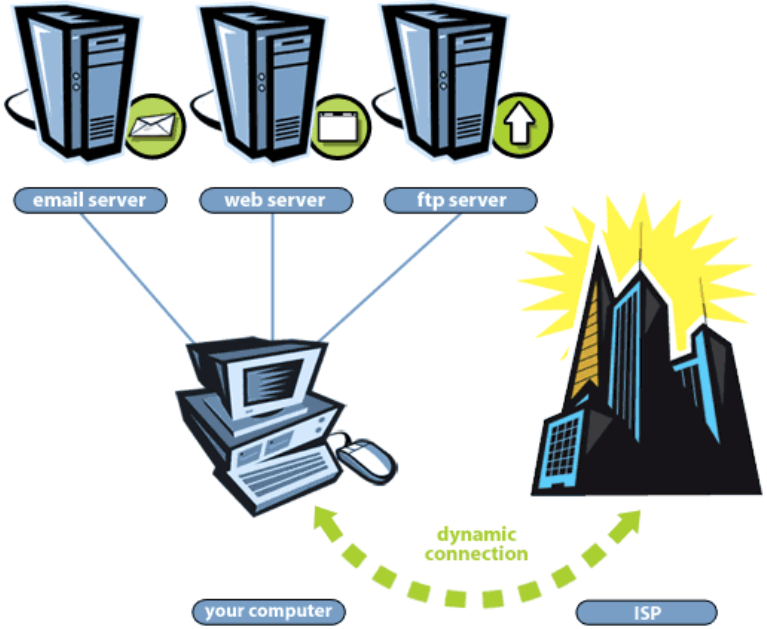
BOEY Lionel
GUO Tianming

Plan

1. Contexte
2. Introduction projet OpenDynDNS
3. Version Publique
4. Version Locale
5. Demonstration
6. Pour aller plus loin....
7. Conclusion

1. Contexte

- DNS
- Dynamique DNS

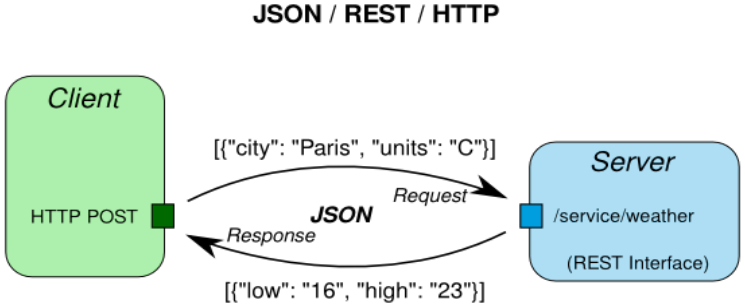


2. Introduction projet OpenDynDNS

- Solution DNS dynamique dans des environnements dynamiques et différents
- Convergence rapide du réseau
- Sécurité
- Flexibilité et extensibilité

2. Introduction projet OpenDynDNS

- Flask
  - web application framework
  - cookies, HTTP authentification, SSL etc
- REST
  - architecture de communication via HTTP/HTTPS



2. Introduction projet OpenDynDNS

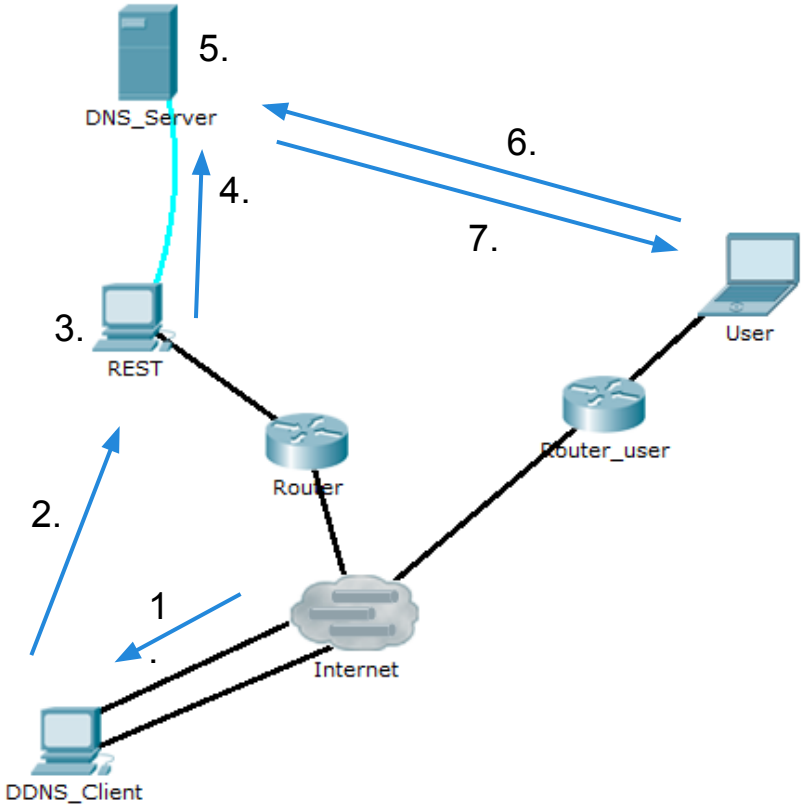
- Zeroconf (mDNS)
  - ensemble de technologies utilisé pour créer rapidement un réseau utilisable
  - librarie pyZeroconf
- Kivy
  - librairies Python pour développer des application mobiles

3. Version publique

- Concept
  - Un serveur DNS est monté derrière un box.
  - Un hôte de ce domaine se déplace dans un réseau différent donc IP publique différent (ex: Starbucks, chez un ami).
  - Les utilisateurs doit être capable de savoir ou est l'hôte.

- Audience
  - Fournisseur des services(web,mail etc) alors que l'IP des hôtes changes regulièrement.

3. Version publique (mise en place)



1. DDNS_client : get IP publique a.b.c.d

2. DDNS_client : envoie une mise à jour vers REST

3. REST : vérification

4. REST à DNS_server : <ddns_client @ a.b.c.d>

5. DNS_server : mise à jour fichier de zone

6. User à DNS_server : Où est ddns.testopendyn.com?

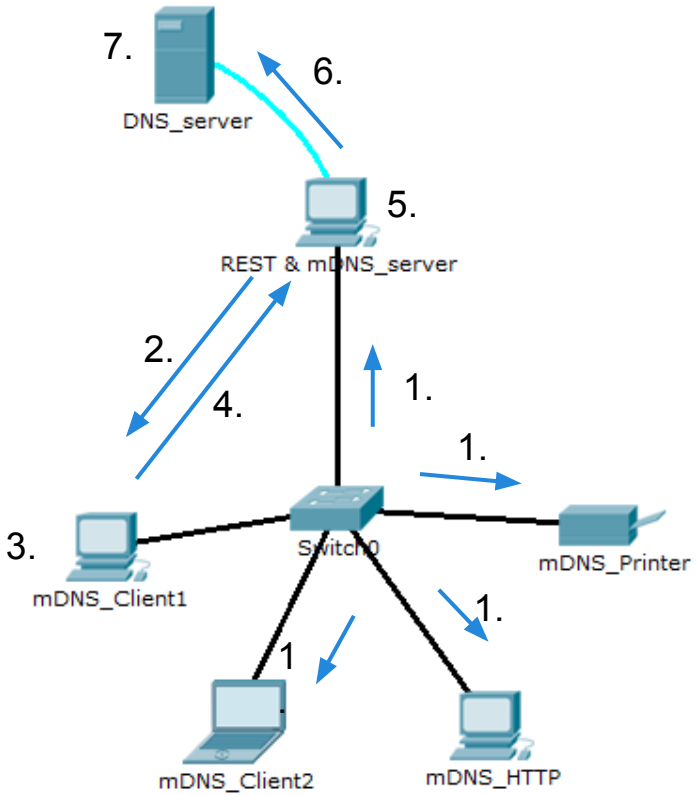
7. DNS_Server à User : <ddns_client @ a.b.c.d>

4. Version locale

- Concept
  - Client doit trouver le vrai service DNS dans un réseau local par mDNS.
  - Mettre à jour le service DNS avec la présence du client.
  - Découvrir les hôtes/services disponibles et les accéder.

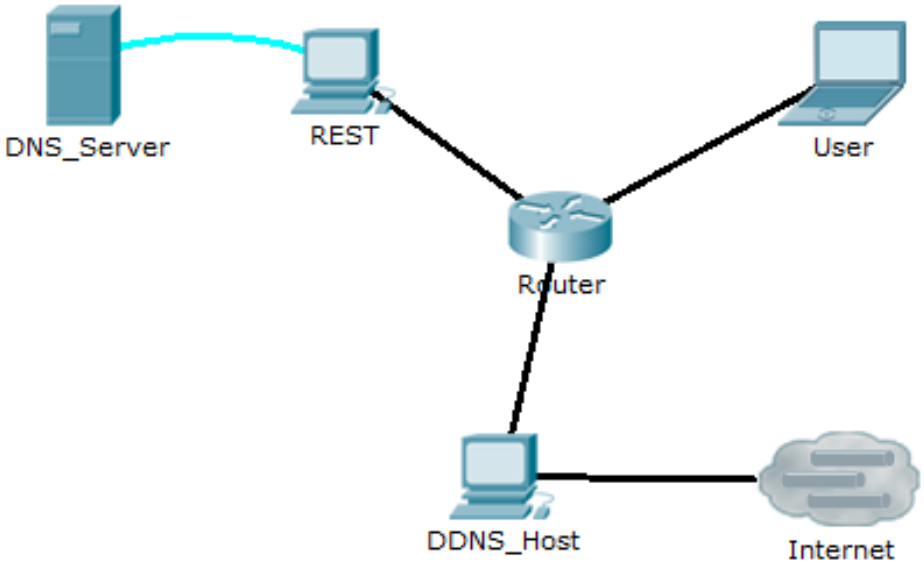
- Audience
  - PME, établissement scolaire, labos de recherche
  - locaux ou il y a des clients qui viennent de qui repartent

4. Version locale (mise en place)

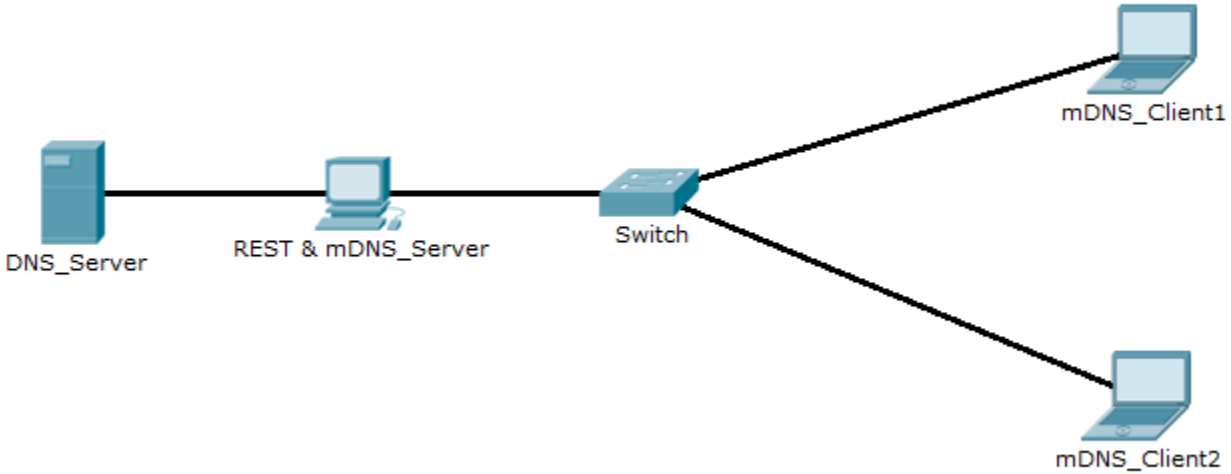


1. client1 au groupe mDNS : Je suis client1. Qui est DNS?
2. mdns_server au groupe mDNS : Je suis DNS
3. client1 : update resolv.conf
4. client1 au REST : Voilà mon adresse IP local a.b.c.d
5. REST : verification
6. REST au DNS_server : <client1 @ a.b.c.d>
7. DNS_server : mise à jour fichier de zone

5. Demo version publique



5. Demo version Locale



5. Pour aller plus loin….

- Implementation UPnP plus elabore
  - un des technologies de Zeroconf
  - hotes version Publique seront accessible quelquesoit leurs IP publique
  - hotes version Locale seront accessible par nom de l’exterieur

- IPV6
  - bind9 supported deja IPv6

- Python 3.X
  - translation 2.x vers 3.x

Conclusion

- Avantages :
  - Multi-plateforme (python)
  - Facilité d’usage (zeroconf)
  - Flexible et extensible (flask)
    - intégrer UPnP
    - ajouter des applications web

- Difficultes rencontres :
  - UPnP n’est pas très développé