JCSA

JOURNÉE
DU CONSEIL
SCIENTIFIQUE
Afnic

4 JUILLET 2022

CONSOMMATION ÉNERGÉTIQUE DU NUMÉRIQUE : INFRASTRUCTURES, SERVICES ET USAGES

Tutoriel - Cycle de vie d'une requête DNS. Quel impact sur le réseau ?

Stéphane Bortzmeyer Alexandre Pion

JCSA22

4 juillet 2022

afnic

Le DNS

- Un protocole client-serveur pour obtenir des informations à partir des noms de domaine
- Fournit des identificateurs parlants et stables
- Comment ça marche ?

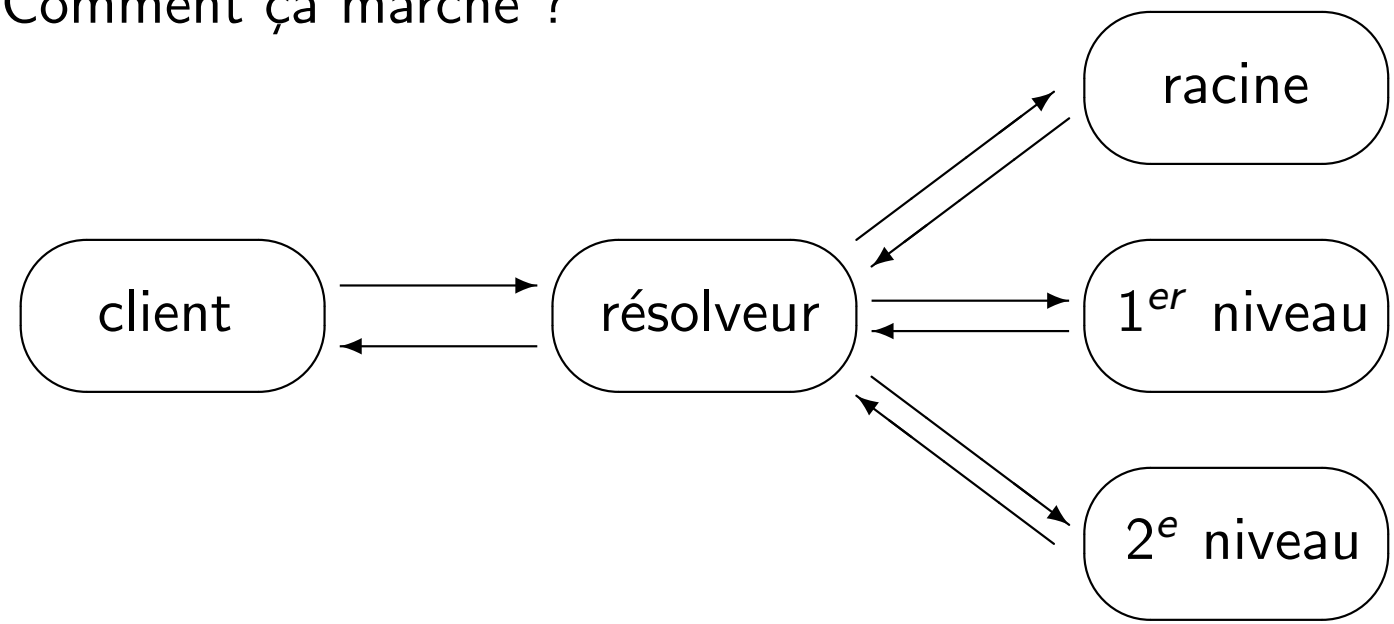


Figure: Schéma Client-Résolveur-{3 serveurs faisant autorité}

Le bailliage

- Trouver l'adresse IP du serveur faisant autorité : un problème d'œuf et de poule
- Première solution : hors-bailliage, on nomme les serveurs de noms en dehors de la zone (`ns1.toto.example` est serveur de `machintruc.example`). Cela nécessite une requête DNS supplémentaire.
- Deuxième solution : dans le bailliage, on nomme les serveurs dans la zone (`ns1.toto.example` est serveur de `toto.example`).

Le DNS et la sécurité

- Dans l'Internet, personne ne vous entend modifier les bits.
- DNSSEC, solution pour authentifier les réponses.
- Signature cryptographique des enregistrements.
- Conséquence : augmentation de la taille des réponses.
- Il existe plusieurs algorithmes, la taille varie.

Le DNS et la vie privée

- Trafic en clair, possibilité de le surveiller
- DoT et DoH pour chiffrer les requêtes (et DoQ !)
- DoT et DoH, comment ça marche ?

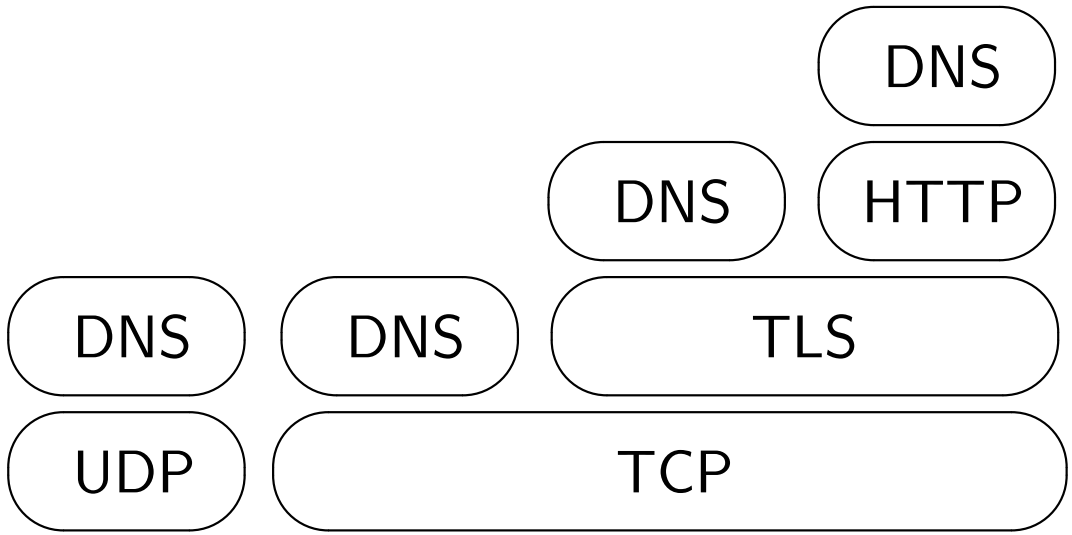


Figure: Les couches des protocoles DNS, DoT et DoH

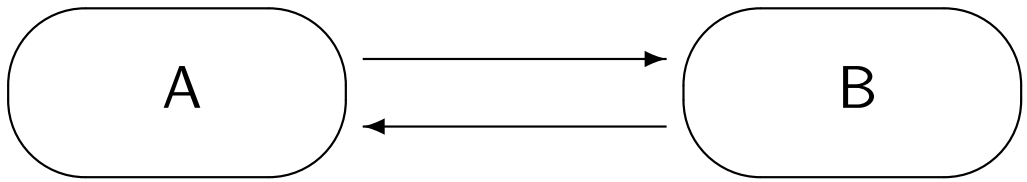
Estimation du nombre de paquets sur le réseau

Pour une question donnée, combien de paquets vont transiter sur le réseau ?

Estimation du nombre de paquets sur le réseau

Pour une question donnée, combien de paquets vont transiter sur le réseau ?

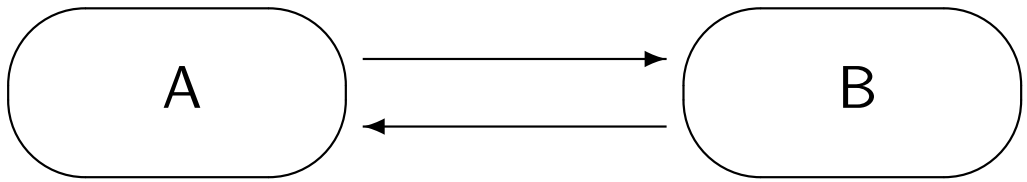
- Décomposer par lien :



Estimation du nombre de paquets sur le réseau

Pour une question donnée, combien de paquets vont transiter sur le réseau ?

- Décomposer par lien :

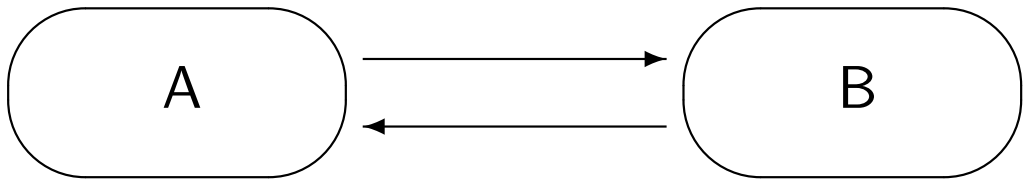


- Recomposer en sommant

Estimation du nombre de paquets sur le réseau

Pour une question donnée, combien de paquets vont transiter sur le réseau ?

- Décomposer par lien :

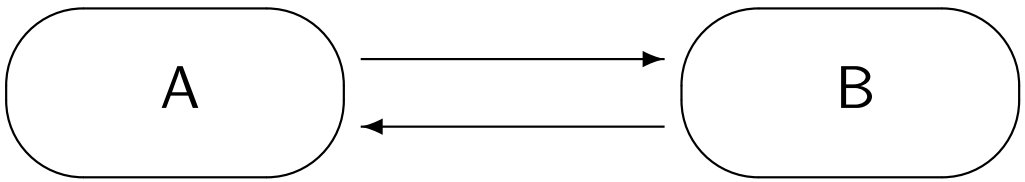


- Recomposer en sommant
- Même pile protocolaire sur un lien

Estimation du nombre de paquets sur le réseau

Pour une question donnée, combien de paquets vont transiter sur le réseau ?

- Décomposer par lien :

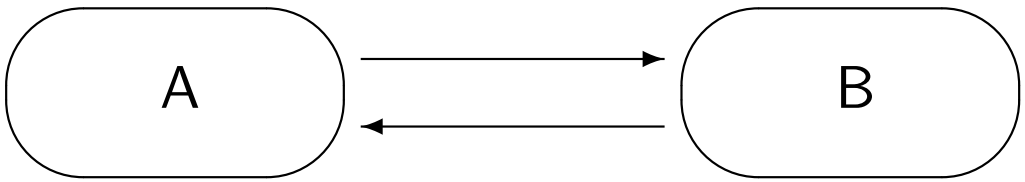


- Recomposer en sommant
- Même pile protocolaire sur un lien
- Taille d'un paquet limitée par la *MTU* (*maximum transmission unit*)

Estimation du nombre de paquets sur le réseau

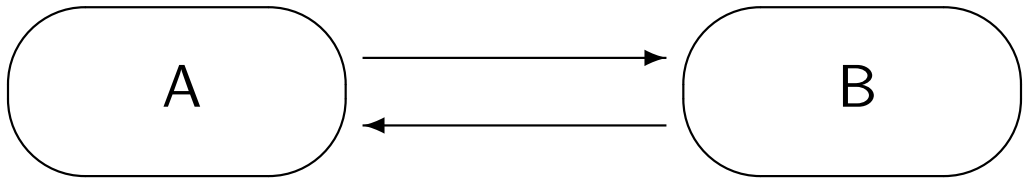
Pour une question donnée, combien de paquets vont transiter sur le réseau ?

- Décomposer par lien :



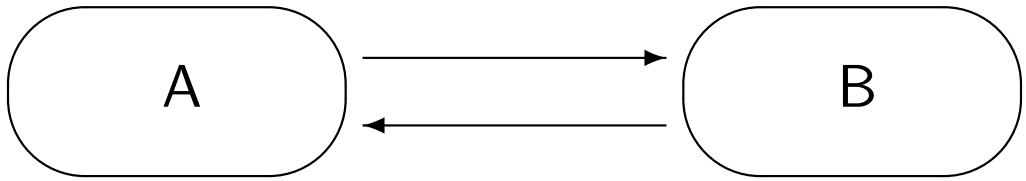
- Recomposer en sommant
- Même pile protocolaire sur un lien
- Taille d'un paquet limitée par la *MTU*
- Coefficient de rejeu : $\alpha \geq 0$ (lié aux perturbations du réseau)
- On pose $\varepsilon = 1 + \alpha$
  $\varepsilon = 1 \Leftrightarrow$ pas de perturbations

Estimation du nombre de paquets sur un lien



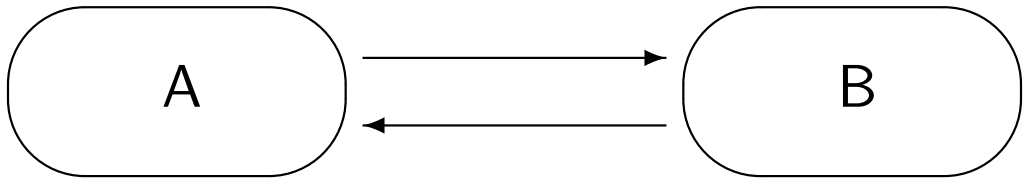
- en UDP : $N = 2\varepsilon$

Estimation du nombre de paquets sur un lien

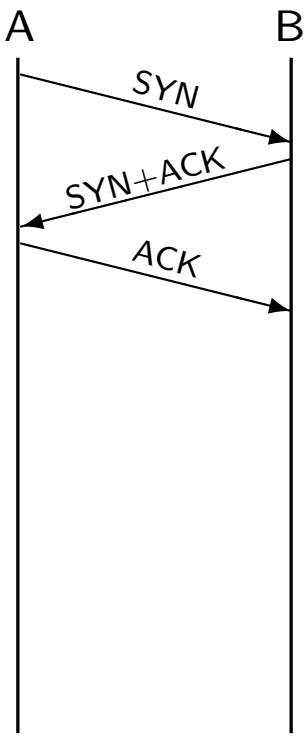


- en UDP : $N = 2\varepsilon$
- en TCP :

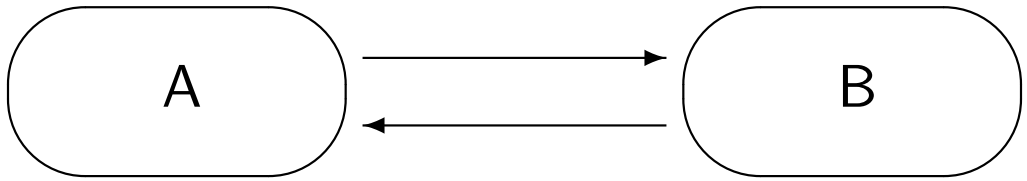
Estimation du nombre de paquets sur un lien



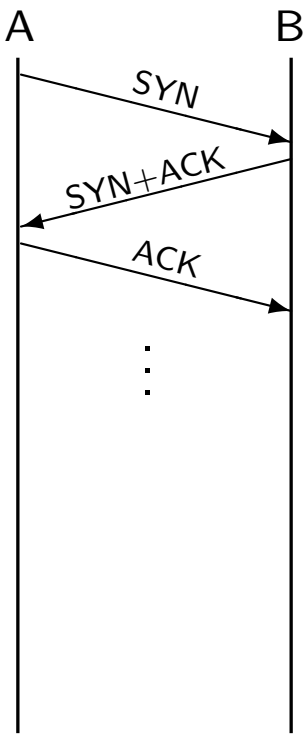
- en UDP : $N = 2\varepsilon$
- en TCP :



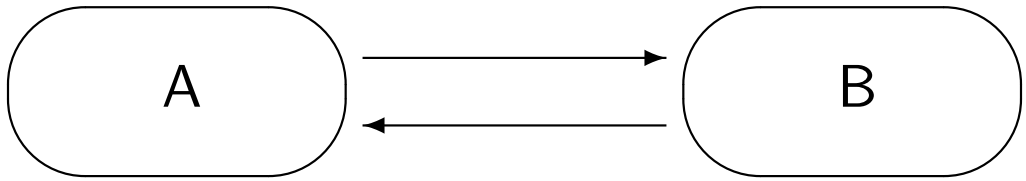
Estimation du nombre de paquets sur un lien



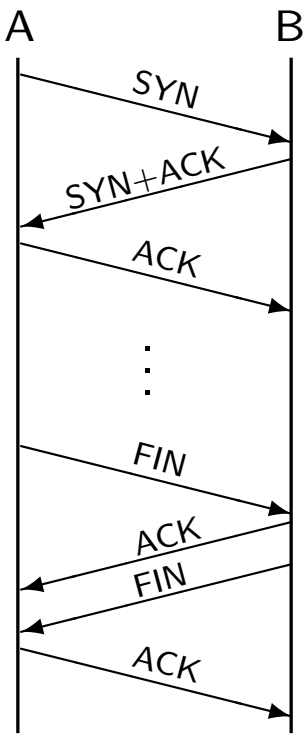
- en UDP : $N = 2\varepsilon$
- en TCP :



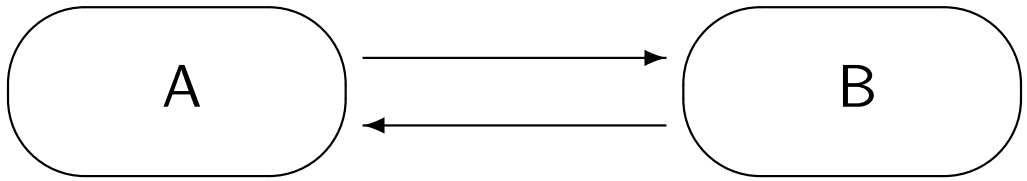
Estimation du nombre de paquets sur un lien



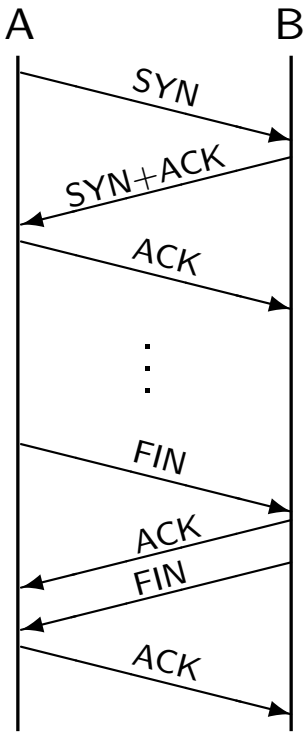
- en UDP : $N = 2\varepsilon$
- en TCP :



Estimation du nombre de paquets sur un lien

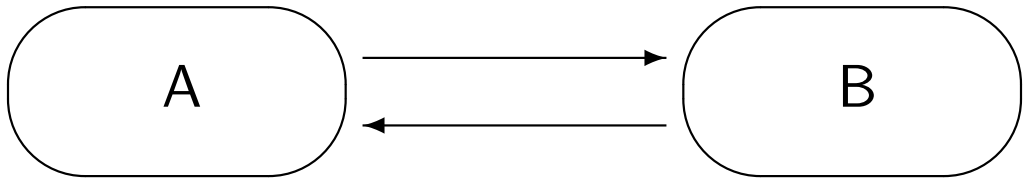


- en UDP : $N = 2\varepsilon$
- en TCP :

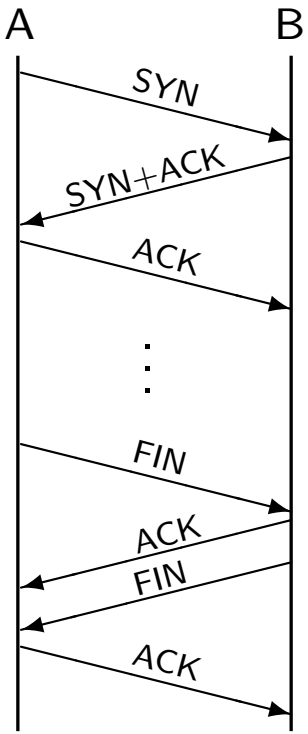


$$N = \varepsilon \left(3 + 4 + 2 \left\lceil \frac{\mathcal{L}(\mathcal{P}_Q)}{MSS} \right\rceil + 2 \left\lceil \frac{\mathcal{L}(\mathcal{P}_R)}{MSS} \right\rceil \right)$$

Estimation du nombre de paquets sur un lien



- en UDP : $N = 2\varepsilon$
- en TCP :



$$N = \varepsilon \left(3 + 4 + 2 \left\lceil \frac{\mathcal{L}(\mathcal{P}_Q)}{MSS} \right\rceil + 2 \left\lceil \frac{\mathcal{L}(\mathcal{P}_R)}{MSS} \right\rceil \right)$$

facteur 2 : ACK non *piggybacké*
$f(x) = \lceil x \rceil$: fonction partie entière supérieure
$\mathcal{L}(\mathcal{P}_Q)$: charge utile de la question
$\mathcal{L}(\mathcal{P}_R)$: charge utile de la réponse
$MSS = MTU - H_{IP} - H_{TCP}$: *maximum segment size*
MTU : *maximum transmission unit*
H_{IP} : taille de l'en-tête IP
H_{TCP} : taille de l'en-tête TCP

Utilisation de DNSSEC

- Des requêtes en plus : DS (sauf pour la racine) et DNSKEY
- Présence de signatures : RRSIG
- RSA vs ECC
- Augmentation du nombre d'octets à envoyer
- Dans la majorité des cas, la taille des signature n'augmente pas le nombre de paquets

Utilisation de DNSSEC

- Des requêtes en plus : DS (sauf pour la racine) et DNSKEY
- Présence de signatures : RRSIG
- RSA vs ECC
- Augmentation du nombre d'octets à envoyer
- Dans la majorité des cas, la taille des signature n'augmente pas le nombre de paquets

- $\Rightarrow 6 + 2(1 + 2 + 2) = 16$ paquets (cache froid)

Répartition de la longueur des réponses

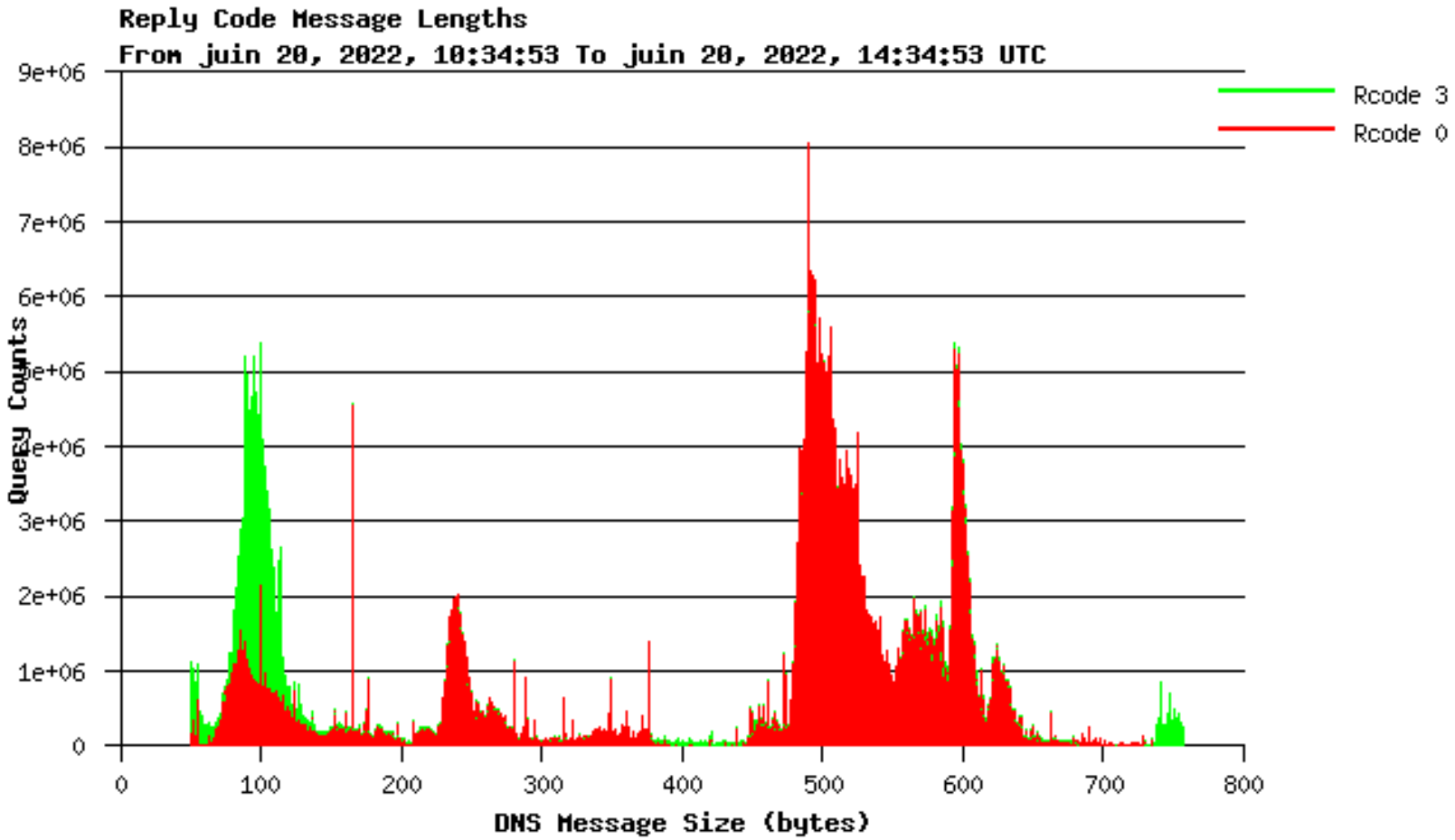


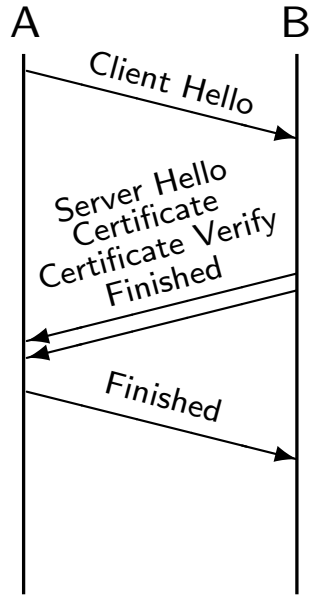
Figure: Nombre de requêtes en fonction de la longueur de la réponse en octets pour `d.nic.fr`

Utilisation de DoT ou DoH

- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets

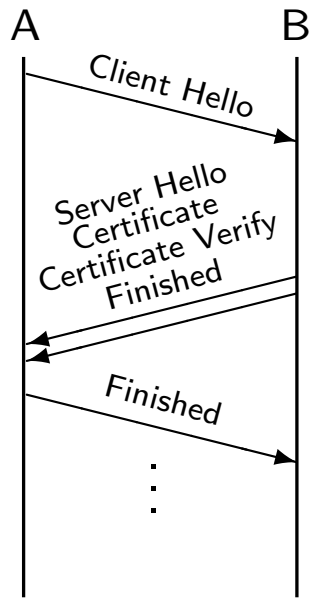
Utilisation de DoT ou DoH

- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets



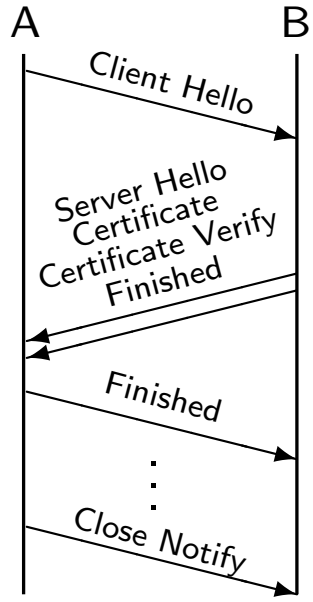
Utilisation de DoT ou DoH

- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets



Utilisation de DoT ou DoH

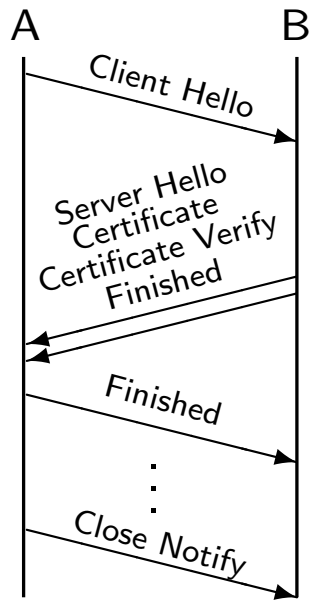
- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets



Utilisation de DoT ou DoH

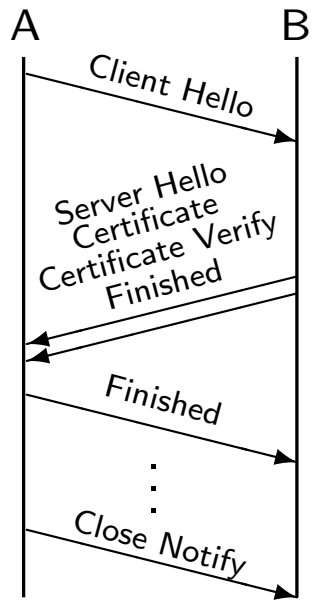
- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets

  - $\Rightarrow 1 + 2 + 1 + 1 = 5$ paquets



Utilisation de DoT ou DoH

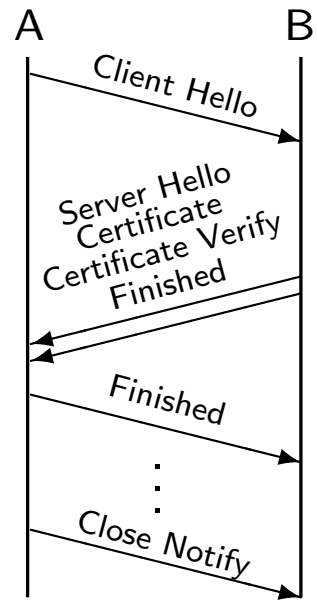
- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets
  - $\Rightarrow 1 + 2 + 1 + 1 = 5$ paquets



- HTTP/2 : échange de *frames* d'initialisation
  - configuration minimale
  - préface HTTP/2 + SETTINGS *frames*
  - réponse et options du serveur

Utilisation de DoT ou DoH

- DoT et DoH : ajout de TLS (TLS 1.3)
- TLS 1.3 : échange de certificats puis chiffrement
  - Certificats X.509
  - RSA $\sim$ 1500 octets
  - ECDSA $\sim$ 800 octets
  - $\Rightarrow 1 + 2 + 1 + 1 = 5$ paquets



- HTTP/2 : échange de *frames* d'initialisation
  - configuration minimale
  - préface HTTP/2 + SETTINGS *frames*
  - réponse et options du serveur
  - $\Rightarrow 1 + 1 = 2$ paquets

Application

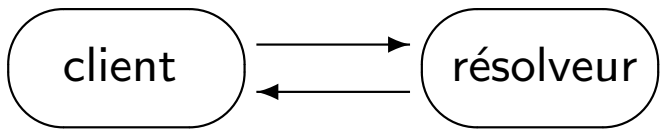
- Hypothèses :
  - dans le bailliage (colle envoyée)
  - 2 frontières de zone
  - pas de fragmentation IP

Application

- Hypothèses :
  - dans le bailliage (colle envoyée)
  - 2 frontières de zone
  - pas de fragmentation IP

- Hypothèses variables :
  - cache chaud vs cache froid
  - avec/sans DNSSEC
  - DoT, DoH ou sans
  - réutilisation des connexions TCP ou non

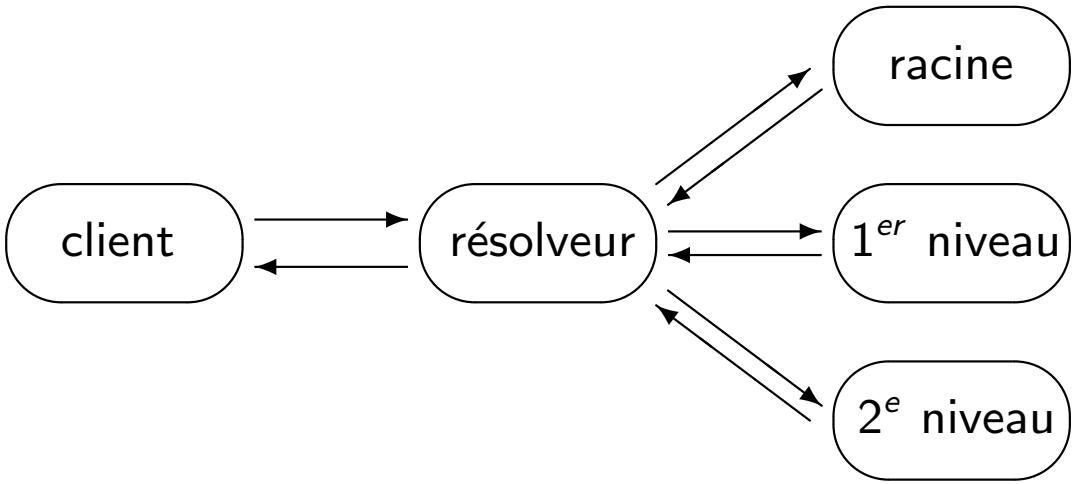
Exemples applicatifs - 1



- UDP et cache chaud :

$$N = 2\varepsilon$$

Exemples applicatifs - 1



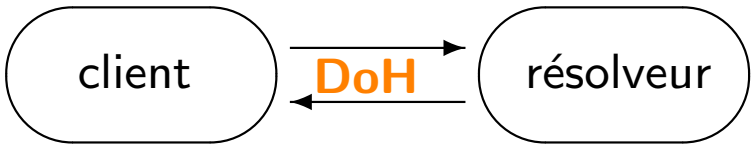
- UDP et cache chaud :

$$N = 2\varepsilon$$

- UDP et cache froid :

$$N = 2\varepsilon + 3 * 2\varepsilon = 8\varepsilon$$

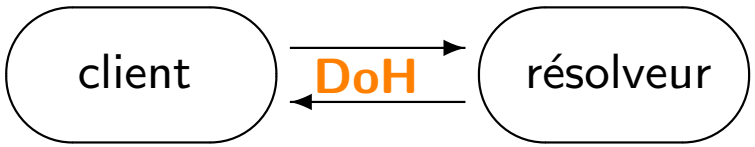
Exemples applicatifs - 2



- DoH entre le client et le résolveur (TCP + TLS + HTTP/2)
- sans réutilisation des connexions TCP

$$N = (3 + 4)\varepsilon + (5 + 2 + 2)\varepsilon$$

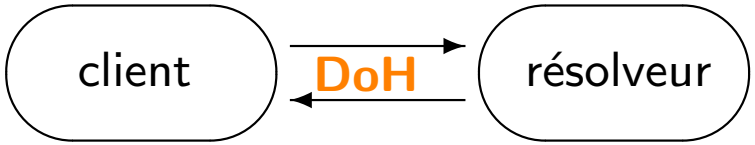
Exemples applicatifs - 2



- DoH entre le client et le résolveur (TCP + TLS + HTTP/2)
- sans réutilisation des connexions TCP
- ACK non *piggybacké*
- cache chaud

$$N = (3 + 4)\varepsilon + 2 * (5 + 2 + 2)\varepsilon$$

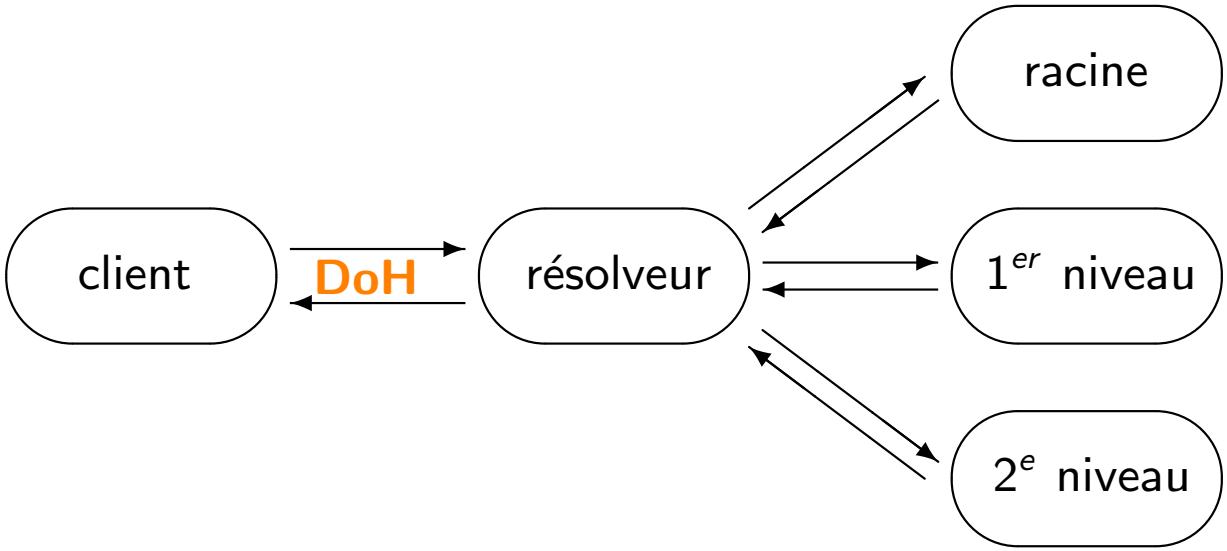
Exemples applicatifs - 2



- DoH entre le client et le résolveur (TCP + TLS + HTTP/2)
- sans réutilisation des connexions TCP
- ACK non *piggybacké*
- cache chaud

$$N = (3 + 4)\varepsilon + 2 * (5 + 2 + 2)\varepsilon$$
$$N = 25\varepsilon$$

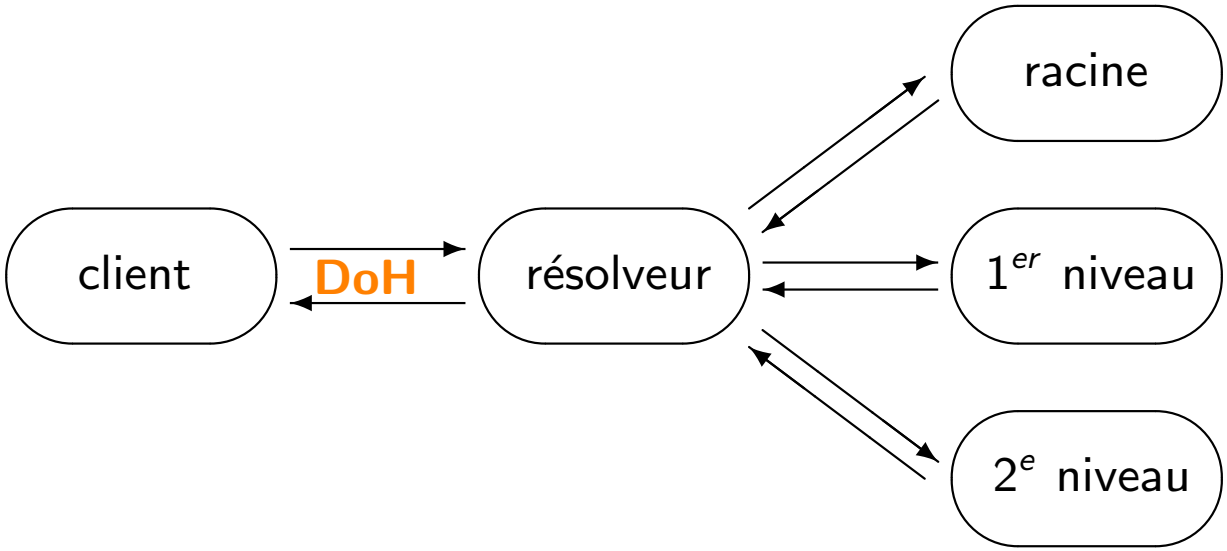
Exemples applicatifs - 2



- DoH entre le client et le résolveur (TCP + TLS + HTTP/2)
- sans réutilisation des connexions TCP
- ACK non *piggybacké*
- cache ~~chaud~~ froid
- UDP avec les serveurs faisant autorité

$$N = (3 + 4)\varepsilon + 2 * (5 + 2 + 2)\varepsilon + 3 * 2\varepsilon$$
$$N = 31\varepsilon$$

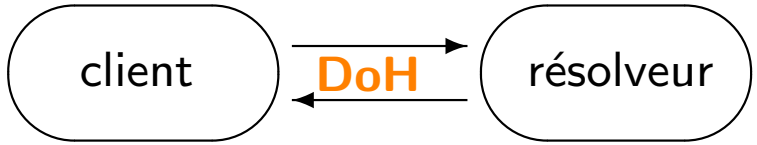
Exemples applicatifs - 2



- DoH entre le client et le résolveur (TCP + TLS + HTTP/2)
- ~~sans réutilisation des connexions TCP~~ connexion réutilisée
- ACK non *piggybacké*
- cache ~~chaud~~ froid
- UDP avec les serveurs faisant autorité

$$N = \ \cancel{(3+4)}\varepsilon \ + 2 * (\ \cancel{5+2}\ + 2)\varepsilon + 3 * 2\varepsilon$$
$$N = 10\varepsilon$$

Exemples applicatifs - 2



- DoH entre le client et le résolveur (TCP + TLS + HTTP/2)
- ~~sans réutilisation des connexions TCP~~ connexion réutilisée
- ACK non *piggybacké*
- cache chaud
- UDP avec les serveurs faisant autorité

$$N = \ \cancel{(3+4)}\varepsilon \ + 2 * (\ \cancel{5+2}\ + 2)\varepsilon + \ \cancel{3*2\varepsilon}$$
$$N = 4\varepsilon$$

Résultats

- Domaine dans le bailliage
- 2 coupures de zone
- Requêtes en UDP entre le résolveur et les serveurs faisant autorité
- Le réseau est supposé sans perte $\Rightarrow \varepsilon = 1$

	DNS UDP	DNS TCP	DoT		DoH	
cache chaud	2	11	21	4*	25	4*
cache froid	8	17	27	10*	31	10*
DNSSEC†	18	27	37	20*	41	20*

*Réutilisation des connexions TCP †Cache froid

Figure: Nombre de paquets pour une requête DNS sous certaines hypothèses

Démo

No.	Time	Source	Destination	Protocol	Length	Info
3	2.384325154	192.168.122.53	192.168.122.1	DNS	73	Standard query 0x501b A doh.rd.nic.fr
4	2.384707925	192.168.122.1	192.168.122.53	DNS	89	Standard query response 0x501b A doh.rd.nic.fr A 5.39.72.86
5	2.384773536	192.168.122.53	192.168.122.1	DNS	73	Standard query 0xb40d AAAA doh.rd.nic.fr
6	2.385425379	192.168.122.1	192.168.122.53	DNS	101	Standard query response 0xb40d AAAA doh.rd.nic.fr AAAA 2001:41d0…
7	2.388044004	192.168.122.53	5.39.72.86	TCP	74	46263 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSv…
8	2.398221763	5.39.72.86	192.168.122.53	TCP	74	443 → 46263 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK…
9	2.398269131	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3505095693 T…
10	2.399413581	192.168.122.53	5.39.72.86	TLSv1.3	470	Client Hello
11	2.412022867	5.39.72.86	192.168.122.53	TCP	66	443 → 46263 [ACK] Seq=1 Ack=405 Win=64768 Len=0 TSval=1328343125…
12	2.412023003	5.39.72.86	192.168.122.53	TLSv1.3	4162	Server Hello, Change Cipher Spec, Encrypted Extensions
13	2.412070712	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=405 Ack=4097 Win=62592 Len=0 TSval=3505095…
14	2.413225762	192.168.122.53	5.39.72.86	TLSv1.3	72	Change Cipher Spec
15	2.415049028	5.39.72.86	192.168.122.53	TLSv1.3	600	Certificate, Certificate Verify, Finished
16	2.415073854	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=411 Ack=4631 Win=64128 Len=0 TSval=3505095…
17	2.465369412	5.39.72.86	192.168.122.53	TCP	66	443 → 46263 [ACK] Seq=4631 Ack=411 Win=64768 Len=0 TSval=1328343…
18	2.465442852	192.168.122.53	5.39.72.86	DoH	447	Standard query 0x0000 A afnic.fr OPT
19	2.475322982	5.39.72.86	192.168.122.53	TCP	66	443 → 46263 [ACK] Seq=4631 Ack=792 Win=64512 Len=0 TSval=1328343…
20	2.475323326	5.39.72.86	192.168.122.53	TLSv1.3	145	New Session Ticket
21	2.475323534	5.39.72.86	192.168.122.53	TLSv1.3	145	New Session Ticket
22	2.475323748	5.39.72.86	192.168.122.53	HTTP2	150	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0], WINDOW_UPDATE[1]
23	2.475405367	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=792 Ack=4710 Win=64128 Len=0 TSval=3505095…
24	2.475718711	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=792 Ack=4789 Win=64128 Len=0 TSval=3505095…
25	2.475860014	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=792 Ack=4873 Win=64128 Len=0 TSval=3505095…
26	2.492695164	5.39.72.86	192.168.122.53	DoH	237	Standard query response 0x0000 A afnic.fr A 192.134.5.37 OPT
27	2.492750561	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=792 Ack=5044 Win=64128 Len=0 TSval=3505095…
28	2.499317690	192.168.122.53	5.39.72.86	TLSv1.3	90	Alert (Level: Warning, Description: Close Notify)
29	2.500258173	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [FIN, ACK] Seq=816 Ack=5044 Win=64128 Len=0 TSval=35…
30	2.508596171	5.39.72.86	192.168.122.53	TCP	66	443 → 46263 [FIN, ACK] Seq=5044 Ack=816 Win=64512 Len=0 TSval=13…
31	2.508596513	5.39.72.86	192.168.122.53	TCP	66	443 → 46263 [ACK] Seq=5045 Ack=817 Win=64512 Len=0 TSval=1328343…
32	2.508670536	192.168.122.53	5.39.72.86	TCP	66	46263 → 443 [ACK] Seq=817 Ack=5045 Win=64128 Len=0 TSval=3505095…

Figure: Trafic entre le client et le résolveur pour une requête DoH pour `afnic.fr`

Réalités architecturales et limites

- Critères autres qu'environnementaux :
  robustesse ⇒ surdimensionnement, Anycast
  sécurité, vie privée
- La consommation n'est pas forcément proportionnelle au trafic
- Consommation réelle des serveurs pour l'Afnic :

serveur	qps	watt
AMS	1350	112
AUB	1750	154
BRU	550	182
FRA	2200	154
LON	650	140

serveur	qps	watt
LYN	550	182
MRS	300	140
NYC	1350	118
TH2	1550	182

Figure: Puissance électrique des serveurs de `d.nic.fr`

Conclusion

- De nombreux critères
- DNS central et peu d'octets pour 1 requête (sauf que beaucoup de requêtes)
- But : expliciter la complexité (beaucoup de cas de figure)
- Pas de données énergétiques utilisée
- Ouvert à faire correspondre des mesures avec des données de consommation
- Beaucoup d'études sur l'impact du web, mais pas à notre connaissance sur le DNS $\Rightarrow$ 1ère brique ici
- On parle ici d'une partie du fonctionnement du DNS (et pas la chaine d'enregistrement, EPP, transferts de zones, employé·es)

Merci pour votre attention

bortzmeyer@afnic.fr pion@afnic.fr