"The missing manual" pour le resolver DNS de Linux

La résolution de domaine sous Linux est un processus simple et statique. Il suffit de renseigner le fichier /etc/resolv.conf avec les adresses IP des serveurs DNS à utiliser. Une option "rotate" permet de changer de serveur DNS à chaque requête et de compléter les noms sans domaine avec le domaine souhaité avec l'option search.

Je parle au présent, mais en pratique les distributions Linux modernes utilisent un resolver DNS qui peut être configuré pour utiliser plusieurs serveurs DNS et la configuration de ce resolver est une dynamique résultant de la combinaison de plusieurs outils. J'ai choisi d'uniformiser la configuration du resolver DNS et des adresses IP en utilisant le trio Netplan / systemd-resolved / systemd-networkd.

Modifiez manuellement le fichier /etc/resolv.conf pour ajouter les serveurs DNS à utiliser donc obsolète, voire voué à l'échec.

Si je regarde le contenu du fichier /etc/resolv.conf voilà ce que j'y trouve:

```
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-
resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but
only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in
a
# different way, replace this symlink by a static file or a different
symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes
of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search iutbeziers.fr
```

Le "Do not edit" est clair, l'origine de la configuration aussi c'est le service systemd-resolved. L'adresse IP du serveur DNS la 127.0.0.53 correspond à une IP locale.

Alors quel est le serveur DNS utilisé par mon système ?

La commande `resolvectl status` est plus riche d'enseignements :

```
root@debian:~# resolvectl status
Global
       Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (eth0)
    Current Scopes: DNS LLMNR/IPv4
         Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
Current DNS Server: 10.6.255.106
       DNS Servers: 10.6.255.106
        DNS Domain: iutbeziers.fr

Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 4 (docker0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
```

Il y a des informations intéressantes dans ce retour. Le serveur DNS utilisé courant est le 10.6.255.106 et le domaine de recherche est iutbeziers.fr. On peut noter que le resolver est lié à une interface réseau, ici l'interface eth0. Les + et - devant les protocoles indiquent si le protocole est activé ou non comme "DefaultRoute" qui m'interpelle rapidement : que vient faire la notion de routage ici ? L'IP 10.6.255.106 est celle que j'ai configuré dans le fichier /etc/netplan/01-netcfg.yaml pour l'interface eth0 ainsi que le domaine de recherche iutbeziers.fr.

```
network:
    version: 2
    renderer: networkd
    ethernets:
        eth0:
            dhcp4: true
            link-local: [ ipv4 ]
            dhcp4-overrides:
              use-dns: false
            nameservers:
              addresses:
              - 10.6.255.106
              search:
              - iutbeziers.fr
```

Cette configuration est donc cohérente avec le retour de la commande `resolvectl status`. J'ai refusé le serveur DNS proposé par le serveur DHCP en utilisant `use-dns: false` et j'ai configuré le serveur DNS à utiliser ainsi que le domaine de recherche. Comment systemd-resolved a-t-il récupéré ces informations ?

La commande `networkctl -ls status eth0` me donne des informations sur l'interface eth0 et surtout le fichier de configuration réseau utilisé pour cette interface et généré par Netplan qui place les configurations qu'il génère dans /run/systemd/network.

```
● 2: eth0
                     Link File: /usr/lib/systemd/network/99-default.link
                  Network File: /run/systemd/network/10-netplan-
eth0.network
                         State: routable (configured)
                  Online state: online
                          Type: ether
...
```

Le fichier /run/systemd/network/10-netplan-eth0.network fait le lien entre netplan et systemd-networkd. Il contient la configuration de l'interface eth0.

```
[Match]
Name=eth0

[Network]
DHCP=ipv4
LinkLocalAddressing=ipv4
DNS=10.6.255.106
Domains=iutbeziers.fr

[DHCP]
RouteMetric=100
UseMTU=true
UseDNS=false
```

systemd-resolved récupère donc les informations de configuration DNS de l'interface eth0 via systemd-networkd.

C'est une configuration fonctionnelle que je teste avec ces commandes :

```
root@debian:~# resolvectl query www.iutbeziers.fr
www.iutbeziers.fr: 146.59.209.152              -- link: eth0

-- Information acquired via protocol DNS in 177.9ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: no
-- Data from: network
root@debian:~# resolvectl query www.umontpellier.fr
```

```
www.umontpellier.fr: 193.51.152.74             -- link: eth0

-- Information acquired via protocol DNS in 314.1ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: no
-- Data from: network
```

La requête part bien de l'interface eth0 à destination du serveur DNS 10.6.255.106 et tous les FQDN sont résolus correctement. Il n'y a pas de chiffrement des données, ni d'authentification des données et au vu du temps de réponse, la requête est bien passée par le réseau et n'était pas en cache.

Voyons ce qu'il se passe si on supprime l'option +DefaultRoute de l'interface eth0.

```
resolvectl default-route eth0 no
root@debian:~# resolvectl status
Global
       Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (eth0)
Current Scopes: DNS LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
   DNS Servers: 10.6.255.106
    DNS Domain: iutbeziers.fr

Link 3 (eth1)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 4 (docker0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
root@debian:~# resolvectl query www.iutbeziers.fr
www.iutbeziers.fr: 146.59.209.152              -- link: eth0

-- Information acquired via protocol DNS in 176.8ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: no
-- Data from: network
root@debian:~# resolvectl query www.umontpellier.fr
*www.umontpellier.fr: resolve call failed: No appropriate name servers or
networks for name found
```

On peut mieux comprendre l'option +DefaultRoute se comporte comme une route par défaut pour le resolver DNS. Si aucune résolution n'est possible on se replie sur un resolver DNS par défaut. La résolution de www.umontpellier.fr n'est pas possible car il n'y a plus de route par défaut.

On sait maintenant comment netplan configure systemd-networkd qui permet de configurer systemd-resolved pour la résolution DNS.

Eliminons maintenant netplan de l'équation avec cette configuration qui nous permet d'avoir une adresse IP mais de rejeter la configuration DNS envoyé par le serveur DHCP. Au passage l'option link-local: [ipv4] permet de ne pas avoir d'adresse IPv6.

```
cat << EOF > /etc/netplan/01-netcfg.yaml
network:
    version: 2
    renderer: networkd
    ethernets:
                eth0:
                    dhcp4: true
                    dhcp4-overrides:
                      use-dns: false
                    link-local: [ ipv4 ]
                eth1:
                    dhcp4: true
                    dhcp4-overrides:
                      use-dns: false
                    link-local: [ ipv4 ]
EOF
netplan apply
```

On confirme qu'il n'y a plus de DNS configuré et que la résolution de nom ne fonctionne plus:

```
root@debian:~# resolvectl status
Global
       Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (eth0)
Current Scopes: LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 3 (eth1)
Current Scopes: LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 4 (docker0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
root@debian:~# resolvectl query www.umontpellier.fr
www.umontpellier.fr: resolve call failed: No appropriate name servers or
networks for name found
root@debian:~# resolvectl query www.iutbeziers.fr
```

```
www.iutbeziers.fr: resolve call failed: No appropriate name servers or
networks for name found
```

Ca serait bien que par défaut j'ai une configuration DNS de repli quand aucun DNS n'est configuré. Le fichier de configuration de systemd-resolved /etc/systemd/resolved.conf contient une option FallbackDNS qui semble correspondre à ce que je cherche.

```
cat << EOF > /etc/systemd/resolved.conf
[Resolve]
FallbackDNS=1.1.1.1
EOF
systemctl restart systemd-resolved
```

En effet, la résolution de nom fonctionne à nouveau avec le serveur DNS 1.1.1.1

```
root@debian:~# resolvectl status
Global
          Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
   resolv.conf mode: stub
Fallback DNS Servers 1.1.1.1

Link 2 (eth0)
Current Scopes: LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 3 (eth1)
Current Scopes: LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 4 (docker0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
root@debian:~# resolvectl query --cache no www.umontpellier.fr
www.umontpellier.fr: 193.51.152.74             -- link: eth0

-- Information acquired via protocol DNS in 51.1ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: no
-- Data from: network
root@debian:~# resolvectl query --cache no www.iutbeziers.fr
*www.iutbeziers.fr: 146.59.209.152              -- link: eth0
                    2001:41d0:301::31           -- link: eth0

-- Information acquired via protocol DNS in 102.8ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
```

```
transport: no
-- Data from: network
```

Donc si il n'y a pas de DNS configuré, systemd-resolved utilise le serveur DNS de repli configuré dans /etc/systemd/resolved.conf grâce à l'option FallbackDNS.

Est-ce que le fallback DNS est utilisé si un DNS est configuré mais ne répond pas ? Je me trompe volontairement de serveur DNS pour voir si le fallback DNS est utilisé.

```
cat << EOF > /etc/systemd/resolved.conf
[Resolve]
DNS=10.10.10.10
FallbackDNS=1.1.1.1
EOF
systemctl restart systemd-resolved
resolvectl query --cache no www.umontpellier.fr
```

L'attente est longue et la résolution de nom échoue. Le fallback DNS n'est pas utilisé comme solution de repli si un DNS est configuré mais ne répond pas.

On réinitialise systemd-resolved pour explorer maintenant la configuration de DNS au travers de systemd-networkd.

```
cat << EOF > /etc/systemd/resolved.conf
[Resolve]
DNS=
FallbackDNS=
EOF
systemctl restart systemd-resolved
```

Créons un fichier de configuration pour l'interface eth0 dans /etc/systemd/network/05-eth0.network pour configurer le serveur DNS à utiliser et le domaine de recherche.

```
cat << EOF > /etc/systemd/network/05-eth0.network
[Match]
Name=eth0
[Network]
DHCP=yes
DNS=8.8.8.8 1.1.1.1
Domains=iutbeziers.fr
EOF
systemctl restart systemd-networkd
```

Vérifions

```
root@debian:~# resolvectl status
Global
       Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (eth0)
    Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6
         Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
Current DNS Server: 8.8.8.8
       DNS Servers: 8.8.8.8 1.1.1.1 10.0.2.3
        DNS Domain: iutbeziers.fr
```

On constate que systemd-resolved a bien récupéré la configuration DNS de l'interface eth0. Cette configuration est plus prioritaire que celle généré par netplan: Les configurations dans /etc/systemd sont prioritaires sur celles dans /run/systemd. C'est discutable l'administrateur système est à l'origine des deux configurations. Il vaut mieux éviter de mélanger les deux.

On peut remarquer qu'il y a un DNS de plus qui correspond à l'adresse IP du serveur DNS fourni par le DHCP de virtualbox.

Entre nos trois serveurs quel est celui utilisé pour la résolution de nom ? Un tcpdump sur l'interface eth0 m'indique que c'est bien le serveur DNS 8.8.8.8 qui est toujours accédé. Si je bloque l'accès à 8.8.8.8 le second serveur DNS 1.1.1.1 est utilisé:

```
root@debian:~# tcpdump -i any port  53
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length
262144 bytes
23:33:28.733402 eth0  Out IP debian.36927 > dns.google.domain: 1738+ [1au]
AAAA? www.ac-montpellier.fr. (50)
23:33:28.733564 eth0  Out IP debian.46633 > dns.google.domain: 52892+ [1au]
A? www.ac-montpellier.fr. (50)
23:33:28.760795 eth0  In  IP dns.google.domain > debian.46633: 52892 2/0/1
CNAME cs1730.wpc.alphacdn.net., A 192.229.220.59 (103)
23:33:28.789931 eth0  In  IP dns.google.domain > debian.36927: 1738 1/1/1
CNAME cs1730.wpc.alphacdn.net. (143)
23:33:28.790870 eth0  Out IP debian.46498 > dns.google.domain: 51932+ [1au]
AAAA? cs1730.wpc.alphacdn.net. (52)
23:33:28.791082 eth0  Out IP debian.33480 > dns.google.domain: 33880+ [1au]
A? cs1730.wpc.alphacdn.net. (52)
23:33:28.814588 eth0  In  IP dns.google.domain > debian.33480: 33880 1/0/1
A 192.229.220.59 (68)
23:33:28.820610 eth0  In  IP dns.google.domain > debian.46498: 51932 0/1/1
(108)
23:33:28.825853 eth0  Out IP debian.51680 > dns.google.domain: 18520+ [1au]
PTR? 8.8.8.8.in-addr.arpa. (49)
23:33:28.839389 eth0  In  IP dns.google.domain > debian.51680: 18520 1/0/1
PTR dns.google. (73)
```

```
23:33:28.840917 eth0  Out IP debian.42232 > dns.google.domain: 3196+ [1au]
PTR? 15.2.0.10.in-addr.arpa. (51)
23:33:28.854545 eth0  In  IP dns.google.domain > debian.42232: 3196
NXDomain 0/0/1 (51)
23:33:28.854990 eth0  Out IP debian.42232 > dns.google.domain: 3196+ PTR?
15.2.0.10.in-addr.arpa. (40)
23:33:28.867253 eth0  In  IP dns.google.domain > debian.42232: 3196
NXDomain 0/0/0 (40)
23:33:45.154366 eth0  Out IP debian.48309 > one.one.one.one.domain: 23502+
[1au] A? www.ac-montpellier.fr. (50)
23:33:45.154578 eth0  Out IP debian.41779 > one.one.one.one.domain: 52082+
[1au] AAAA? www.ac-montpellier.fr. (50)
23:33:45.172676 eth0  In  IP one.one.one.one.domain > debian.48309: 23502
2/0/1 CNAME cs1730.wpc.alphacdn.net., A 192.229.220.59 (103)
23:33:45.198304 eth0  In  IP one.one.one.one.domain > debian.41779: 52082
1/1/1 CNAME cs1730.wpc.alphacdn.net. (143)
23:33:45.199448 eth0  Out IP debian.33815 > one.one.one.one.domain: 3132+
[1au] A? cs1730.wpc.alphacdn.net. (52)
23:33:45.199647 eth0  Out IP debian.35301 > one.one.one.one.domain: 38792+
[1au] AAAA? cs1730.wpc.alphacdn.net. (52)
23:33:45.213637 eth0  In  IP one.one.one.one.domain > debian.33815: 3132
1/0/1 A 192.229.220.59 (68)
23:33:45.213637 eth0  In  IP one.one.one.one.domain > debian.35301: 38792
0/1/1 (108)
23:33:45.254991 eth0  Out IP debian.59207 > one.one.one.one.domain: 15207+
[1au] PTR? 1.1.1.1.in-addr.arpa. (49)
23:33:45.269579 eth0  In  IP one.one.one.one.domain > debian.59207: 15207
1/0/1 PTR one.one.one.one. (78)
```

Il n'y a donc pas d'équilibrage de charge entre les serveurs DNS configurés.

Supprimons maintenant la configuration faite pour systemd-networkd pour voir ce que l'on peut faire avec systemd-resolved.

```
rm -f /etc/systemd/network/05-eth0.network
systemctl restart systemd-networkd
systemctl restart systemd-resolved
```

```
cat << EOF > /etc/systemd/resolved.conf
[Resolve]
DNS=1.1.1.1
FallbackDNS=
EOF
systemctl restart systemd-resolved
```

Comme on peut le voir, cette configuration génére une configuration DNS globale et non plus par interface réseau comme avec systemd-networkd:

```
root@debian:~# resolvectl status
Global
       Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub
      DNS Servers 1.1.1.1

Link 2 (eth0)
Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6
     Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
   DNS Servers: 8.8.8.8 1.1.1.1 10.0.2.3
    DNS Domain: iutbeziers.fr

Link 3 (eth1)
Current Scopes: LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported

Link 4 (docker0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
```

Quels sont les priorités de résolution entre une interface globale et locale ? Si un domaine est configuré pour une interface réseau, il est prioritaire sur le domaine global:

```
root@debian:~# resolvectl query --cache no www.umontpellier.fr
www.umontpellier.fr: 193.51.152.74             -- link: eth1

-- Information acquired via protocol DNS in 56.1ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: no
-- Data from: network
root@debian:~# resolvectl query --cache no www.iutbeziers.fr
www.iutbeziers.fr: 146.59.209.152              -- link: eth0
                   2001:41d0:301::31           -- link: eth0

-- Information acquired via protocol DNS in 10.3542s.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: no
-- Data from: network
```

Alors quels sont les avantages de cette configuration dynamique du resolver DNS par rapport à la configuration statique de /etc/resolv.conf ?

Certains noms dns sont identifiés comme des noms de services et sont résolus par systemd-resolved. Par exemple, le nom _gateway est résolu par systemd-resolved et correspond aux passerelles des interfaces réseau. _outbound est un autre nom de service résolu par systemd-resolved et correspond aux adresses IP des interfaces réseau.

```
root@debian:~# resolvectl query  _gateway
_gateway: 192.168.1.1                          -- link: eth1
          10.0.2.2                             -- link: eth0

-- Information acquired via protocol DNS in 672us.
-- Data is authenticated: yes; Data was acquired via local or encrypted
transport: yes
-- Data from: synthetic

root@debian:~# resolvectl query  _outbound
_outbound: 10.0.2.15                           -- link: eth0
           192.168.1.32                        -- link: eth1

-- Information acquired via protocol DNS in 920us.
-- Data is authenticated: yes; Data was acquired via local or encrypted
transport: yes
-- Data from: synthetic
```

Quelques commandes sont intéressantes :

```
root@debian:~# resolvectl dns
Global:
Link 2 (eth0): 8.8.8.8 1.1.1.1 10.0.2.3
Link 3 (eth1):
Link 4 (docker0):
root@debian:~# resolvectl domain
Global:
Link 2 (eth0): iutbeziers.fr
Link 3 (eth1):
Link 4 (docker0):
root@debian:~# resolvectl default-route
Global
       Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub
Link 2 (eth0): yes
Link 3 (eth1): no
Link 4 (docker0): no
root@debian:~# resolvectl statistics
DNSSEC supported by current servers: no

Transactions
Current Transactions: 0
  Total Transactions: 43

Cache
  Current Cache Size: 2
          Cache Hits: 10
        Cache Misses: 27

DNSSEC Verdicts
```

```
              Secure: 0
            Insecure: 0
               Bogus: 0
       Indeterminate: 0

root@debian:~# resolvectl monitor
→ Q: www.iutbeziers.fr IN A
→ Q: www.iutbeziers.fr IN AAAA
← S: success
← A: www.iutbeziers.fr IN AAAA 2001:41d0:301::31
← A: www.iutbeziers.fr IN A 146.59.209.152
...
```

Il y a mieux. On peut activer le DNS over TLS avec l'option DNSOverTLS=opportunistic dans /etc/systemd/resolved.conf.

```
cat << EOF > /etc/systemd/resolved.conf
[Resolve]
DNS=1.1.1.1
FallbackDNS=
DNSOverTLS=opportunistic
DNSSEC=allow-downgrade
EOF
```

```
systemctl restart systemd-resolved
root@debian:~# resolvectl status
Global
       Protocols: +LLMNR +mDNS DNSOverTLS=opportunistic DNSSEC=allow-
downgrade/supported
resolv.conf mode: stub
      DNS Servers 1.1.1.1

Link 2 (eth0)
    Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6
         Protocols: +DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic
DNSSEC=allow-downgrade/supported
Current DNS Server: 8.8.8.8
       DNS Servers: 8.8.8.8 1.1.1.1 10.0.2.3
        DNS Domain: iutbeziers.fr

Link 3 (eth1)
Current Scopes: LLMNR/IPv4
     Protocols: -DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic
DNSSEC=allow-downgrade/supported

Link 4 (docker0)
Current Scopes: none
     Protocols: -DefaultRoute +LLMNR -mDNS DNSOverTLS=opportunistic
DNSSEC=allow-downgrade/supported
```

Le traffic DNS est maintenant chiffré avec DNS over TLS sur le port 853:

```
root@debian:~# resolvectl query --cache no www.nushell.sh
www.nushell.sh: 185.199.108.153                -- link: eth1
                185.199.109.153                -- link: eth1
                185.199.110.153                -- link: eth1
                185.199.111.153                -- link: eth1
                2606:50c0:8000::153            -- link: eth1
                2606:50c0:8003::153            -- link: eth1
                2606:50c0:8002::153            -- link: eth1
                2606:50c0:8001::153            -- link: eth1
                (nushell.github.io)

-- Information acquired via protocol DNS in 603.4ms.
-- Data is authenticated: no; Data was acquired via local or encrypted
transport: yes
-- Data from: network

root@debian:~# delv www.cloudflare.com
; fully validated
www.cloudflare.com.     51      IN      A       104.16.123.96
www.cloudflare.com.     51      IN      A       104.16.124.96
www.cloudflare.com.     51      IN      RRSIG   A 13 3 300 20240515225924
20240513205924 34505 www.cloudflare.co
m. XzPghTteKkr8Ew1N4AQGMCuo9KwFc0KsoO/ayJqB6eeB7sX3eYHU6yFW
ZxywdhUj0Qx6/3rjjPGkkiSmHhtxzg==
```

Pour le côté dynamique une application intéressante est de pouvoir configurer un serveur DNS quand une interface VPN est activée et de le supprimer quand l'interface VPN est désactivée.

On installe le paquet systemd-networkd-dispatcher qui déclenche des scripts lors de changements d'état des interfaces réseau.

```
apt install systemd-networkd-dispatcher
```

Le script suivant placé dans /etc/systemd/network/05-vpn.network permet de configurer le serveur DNS à utiliser quand l'interface VPN est activée.

```
#!/bin/bash
INTERFACE=tun0
# Add your search domains here
SEARCH_DOMAINS="~iutbeziers.fr"
resolvectl domain "$INTERFACE" $SEARCH_DOMAINS
resolvectl dns $INTERFACE 10.6.255.106
```

un resolvectl status permet de vérifier que la configuration a bien été prise en compte:

```
...
Link 15 (tun0)
    Current Scopes: DNS
         Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
Current DNS Server: 10.6.255.106
       DNS Servers: 10.6.255.106
        DNS Domain: ~iutbeziers.fr
```

Le tilde de ~iutbeziers.fr indique que le domaine de recherche est configuré pour l'interface VPN. Chaque fois qu'un FQDN contient iutbeziers.fr, la requête DNS est envoyée au serveur DNS accessible via l'interface VPN. Le ~ indique un routing domain. Si j'avais mis iutbeziers.fr sans le ~, on aurait un search domain. Il y aurait en plus complétion des noms de domaine sans domaine avec le domaine iutbeziers.fr.

Si on souhaite avoir un routing domain global qui matche tous les domaines, on peut utiliser le domaine ~. dans le fichier de configuration de l'interface VPN.

```
cat << EOF > /etc/systemd/resolved.conf
[Resolve]
DNS=1.1.1.1
DNSOverTLS=opportunistic
DOMAINS=~.
EOF
```

Tout ce qui n'est iutbeziers.fr est résolu par le serveur DNS global. C'est un routage par défaut des requêtes DNS.