# **Les limites du DNS Round Robin**

Le **DNS Round Robin** est une technique simple de répartition de charge (load balancing) basée sur le DNS. Elle consiste à associer plusieurs adresses IP à un même nom de domaine. À chaque requête DNS, le serveur retourne ces IP dans un ordre tournant (round robin). Mais cette méthode a plusieurs **limites importantes** :

**1. Aucune prise en compte de la charge réelle des serveurs**

Le DNS ne connaît pas l’état des serveurs (CPU, mémoire, trafic, etc.). Il distribue les IPs sans savoir si un serveur est déjà surchargé ou même indisponible.

Exemple : Si un serveur tombe, il reste quand même dans la liste DNS et les clients peuvent toujours être envoyés vers lui.

**2. Pas de vérification automatique de disponibilité**

Contrairement aux load balancers intelligents, le DNS Round Robin ne peut pas détecter si un serveur est hors ligne. À moins d’un TTL très court (voir point suivant), les clients peuvent continuer à essayer d’accéder à un serveur inactif.

**3. Dépendance au TTL et au cache DNS**

Les enregistrements DNS sont souvent mis en cache par les résolveurs DNS ou même les navigateurs. Cela signifie que :

* Un client peut utiliser la même IP pendant longtemps.
* Les modifications (ajout/retrait d’IP) mettent du temps à se propager.

Durée de vie (TTL) des enregistrements: Un TTL trop long empêche une redistribution rapide, tandis qu'un TTL trop court augmente la charge sur les serveurs DNS.

Comportement incohérent des résolveurs : Différents fournisseurs DNS et systèmes d'exploitation interprètent différemment les réponses Round Robin.

**4. Répartition non uniforme**

Certains clients ou résolveurs DNS ne respectent pas toujours l’ordre round robin :

* Certains les trient par ordre alphabétique ou IP.
* D’autres conservent la première IP et ne consultent jamais les autres.

**5. Pas de connexion persistante**

Le DNS Round Robin ne gère pas la "sticky session" (session persistante), ce qui peut poser problème pour des applications web où un utilisateur doit rester sur le même serveur (ex : panier d’achat, session utilisateur...).

De plus, L’incompatibilité avec HTTP Keep-Alive : Les connexions persistantes peuvent contourner la répartition de charge en restant connectées au même serveur.

**6. Pas adapté aux grandes architectures ou CDN**

Il ne prend pas en compte la **localisation géographique** des utilisateurs ni **l’état réseau**.

Latence variable non prise en compte : Tous les serveurs sont traités de manière égale, même si certains sont géographiquement plus éloignés ou sur des réseaux plus lents (Ex : Un utilisateur en France peut être envoyé vers un serveur en Asie, ce qui augmente la latence.)

Surcharge possible de serveurs spécifiques : Certains services peuvent créer des charges inégales (Ex : le streaming vs les requêtes API).

**Conclusion**

Le DNS Round Robin est une solution **très simple et peu coûteuse** pour répartir les requêtes sur plusieurs serveurs, mais elle est **loin d’être suffisante pour des architectures critiques ou de haute disponibilité**. Pour des cas plus avancés, il vaut mieux utiliser :

* Des **load balancers (HAProxy, NGINX, etc.)**
* Des **DNS intelligents** (avec health check et géo-répartition)
* Des **CDN ou reverse proxies** pour des applications web à grande échelle.