

# RFC 7065 : Traversal Using Relays around NAT (TURN) Uniform Resource Identifiers

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 novembre 2013

Date de publication du RFC : Novembre 2013

<https://www.bortzmeyer.org/7065.html>

---

Ce court RFC normalise un nouveau **plan** d'URI, `turn:`, qui sera utilisé pour la configuration des clients TURN, un protocole de traversée des obstacles comme les routeurs NAT.

C'est le travail sur WebRTC qui a ravivé l'intérêt pour ce nouveau modèle d'URI (envisagé depuis longtemps mais jamais normalisé). WebRTC (RFC 8825<sup>1</sup>) est un mécanisme de communication directe entre navigateurs Web. Souvent, les logiciels WebRTC vont devoir passer à travers des environnements hostiles comme des routeurs NAT. Une des techniques fréquemment utilisées pour aider ce passage est TURN, normalisé dans le RFC 8656, qui permet à un client TURN, en se connectant à un serveur TURN (en général installé par un fournisseur de services, par exemple SIP), de faire relayer ses paquets par le serveur TURN. TURN est une extension de STUN et s'utilise pour les cas désespérés, lorsque la seule solution pour communiquer est de faire relayer tout le trafic.

Mais, pour cela, il faut que le client soit configuré avec les coordonnées d'un serveur TURN. Actuellement, cela se fait d'une manière spécifique à chaque client. D'où l'idée d'avoir un mécanisme simple de désignation du serveur TURN, un URI comme `turn:example.net`. Il n'y aura plus qu'à le copier/coller à l'endroit indiqué. Cela simplifiera la configuration et la documentation. Combiné avec le mécanisme de résolution du RFC 5928, il n'y aura désormais presque rien à faire pour configurer TURN.

La section 3 fournit la syntaxe exacte. Il y a deux plans d'URI, `turn:` et `turns:`, le second servant aux connexions sécurisées avec TLS. Ils sont notés dans le registre IANA des plans <<https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>>. Un numéro de port est possible comme par exemple `turns:provider.example:8888`. S'il est absent, le port par défaut est 3478 pour `turn:` et 5349 pour `turns:`. Rappelez-vous que le nom du serveur, lui, sera déduit du nom de

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc8825.txt>

domaine indiqué dans l'URI par une recherche SRV, après passage par le mécanisme de résolution du RFC 5928.

Et les mises en œuvre? Le logiciel `turnuri` (distribué en `<http://debian.implementers.org/stable/source/turnuri.tar.gz>`) met en œuvre les URI `turn:` ainsi que le mécanisme de résolution du RFC 5928. C'est également le cas de la bibliothèque `libjingle` `<https://code.google.com/p/chromium/codesearch#chromium/src/third_party/libjingle>`, qu'utilise Chrome pour WebRTC (Firefox a sa propre implémentation).

Ce RFC a eu une histoire très longue et compliquée, remontant à plusieurs années `<https://www.ietf.org/mail-archive/web/uri-review/current/msg01089.html>`. Ceux qui s'intéressent aux choix effectués (par exemple d'un plan `turns:` plutôt que d'un paramètre `;proto=tls` dans l'URI) peuvent consulter un article de l'auteur sur son blog `<http://blog.marc.petit-huguenin.org/2012/09/on-design-of-stun-and-turn-uri-formats.html>` et la discussion à l'IETF `<http://www.ietf.org/mail-archive/web/behaveworking/current/msg10047.html>`. Parmi les nombreux messages échangés à l'IETF sur ces URI, je vous suggère la soumission originale et la discussion qui a suivi `<http://www.ietf.org/mail-archive/web/rtcweb/current/msg03476.html>`.