RFC 7605 : Recommendations on Using Assigned Transport Port Numbers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 août 2015

Date de publication du RFC : Août 2015

https://www.bortzmeyer.org/7605.html

Vous développez une application Internet et vous utilisez des ports enregistrés à l'IANA <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml> ? Ou bien aucun port ne convient et vous voulez en enregistrer un nouveau ? Ce RFC est fait pour vous en expliquant, du point de vue du créateur de nouveaux services Internet, comment utiliser les ports ou bien en demander un nouveau.

La procédure complète figure dans le RFC 6335 [1]. Ce nouveau RFC 7605 ne le remplace pas mais fournit une vision moins bureaucratique du processus, plus orientée vers le développeur, avec des recommandations concrètes.

Donc, c'est quoi, un port ? C'est un entier de 16 bits qui :
— Sert à démultiplexer les paquets entrants dans une machine (à les délivrer au bon processus),
— Sert à identifier un service (25 = SMTP).
Seule la deuxième utilisation nécessite un registre central.

Pour se connecter à une autre machine, il faut connaître le port de destination. Le cas le plus connu est celui des URL. Si l'URL n'indique pas de numéro de port, le port par défaut est 80 (RFC 7230, section 2.7.1). Donc, avec http://www.example.com/, le client HTTP se connectera au port 80 de la machine www.example.com. Mais on peut indiquer un port dans un URL (RFC 3986, section 3.2.3). Si l'URL est http://www.example.com:9081/, le client HTTP se connectera au port 9 081 de la machine www.example.com.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6335.txt

Bien sûr, rien n'oblige à faire tourner un service sur le port officiellement affecté. On peut mettre un serveur DNS sur le port 80 ou au contraire un serveur HTTP sur le port 53, normalement prévu pour le DNS. C'est ainsi que bien des serveurs SSH écoutent sur le port 443, de manière à ce qu'on puisse se connecter même depuis les points d'accès WiFi les plus pourris qui filtrent tous les ports que leur concepteur ne connait pas. Attention toutefois, si des équipements situés sur le trajet font du DPI, cela peut ne pas leur plaire de voir du trafic non-DNS sur le port 53 et cela peut les amener à couper la communication. Comme le note notre section 5, supposer que le trafic sur le port 53 est forcément du DNS est une erreur : l'interprétation des numéros de ports doit être laissé aux deux machines qui communiquent. Le registre des ports est juste là pour leur faciliter la tâche, pas pour ajouter des contraintes inutiles.

Bien sûr, ce principe n'est pas forcément suivi et des tas de logiciels utilisent le registre des ports en supposant que les services correspondent aux numéros de ports (cf. aussi la section 6.2). C'est fréquent sur les pare-feux, par exemple. On veut couper le DNS, pour forcer l'usage des DNS menteurs <https://www.bortzmeyer.org/dns-menteur.html>, on bloque le port 53 <https://www.bortzmeyer.org/port53-filtre.html>, ce qui sera contourné en envoyant le trafic DNS sur un autre port et ainsi de suite. Autre cas où un logiciel utilise le registre des ports pour se faciliter la vie, mais parfois à tort, celui des logiciels d'analyse du trafic réseau, comme tcpdump. Le trafic est avec le port 53, tcpdump l'analyse comme si c'était du DNS. Cela peut mener à des résultats amusants. Ici, tcpdump décode du trafic HTTP en croyant que c'est du DNS :

```
18:34:03.263985 IP 127.0.0.1.46680 > 127.0.0.1.53: \
        Flags [P.], seq 1:111, ack 1, win 342, \
        options [nop,nop,TS val 240543975 ecr 240543975], \
        length 11021536 update+ [b2&3=0x2f20] \
           [21584a] [18516q] [12081n] [11825au][|domain]
```

Ce qui le mène à des résultats absurdes comme de croire que la section « Question » du message DNS comprend 18 516 questions... À noter que Wireshark, lui, permet de choisir le décodeur, pour ne pas utiliser celui par défaut d'un port donné (menu "*Analyze*"/Analyser, puis "*Decode as*"/Decoder[sic]).

Aujourd'hui, au moment de la publication de notre RFC 7605, quelle est l'utilisation de l'espace des ports ? Cet espace est divisé (cf. RFC 6335) en trois catégories, « Système » (sur une machine Unix, il faut être root pour écouter sur ces ports), de 0 à 1 023, « Utilisateur », de 1 024 à 49 151, et « Dynamique » (aussi appelé « Privé », les ports locaux, non enregistrés à l'IANA, contrairement aux deux premières catégories), de 49 152 à 65 535.

Donc, un problème à garder en tête avant de réclamer son port, la nécessite de préserver une ressource finie. L'espace des numéros de ports est partagé par tous les utilisateurs de l'Internet et cet espace est petit, moins de $2\hat{}16$. Sur les 49 151 numéros ouverts à l'enregistrement (à l'exception, donc, de la catégorie « Dynamique »), 5 850 ont été enregistrés pour TCP. Les enregistrements sont en théorie annulables (RFC 6335, section 8.4) mais permanents en pratique (la procédure du RFC 6335 semble parfaitement irréaliste, cf. aussi la section 7.9 de notre RFC). Il faut donc y aller mollo (cette question du conservatisme dans l'allocation a été une des plus disputées au sein du groupe de travail IETF). Par exemple, chaque service ne devrait avoir qu'un seul numéro de port (outre l'économie des numéros de port, cela évite des problèmes de débogages pénibles au cas où certains ports soient bloqués par un pare-feu et pas d'autres). La section 7.2 détaille cette question des ports multiples.

Donc, plutôt que de demander plusieurs ports pour votre service :

— Prenez un seul port bien connu et indiquez les éventuels autres ports dynamiquement, comme le "*passive FTP*" (RFC 959),

— Utilisez un service externe pour indiquer un port dynamique, comme le permet le "*portmapper*" (RFC 1833),
— Utilisez une fonction de signalisation du protocole, comme le champ `Host:` de HTTP, qui évite d'avoir un port (ou une adresse IP) par site Web.

Les ports sont censés permettre de séparer des services différents, pas juste des variations d'un même service.

La section 7 du RFC couvre toutes les questions qu'il faut se poser avant l'enregistrement d'un nouveau port. D'abord, est-il vraiment nécessaire, compte tenu des exigences de conservation indiquées plus haut ?

— Est-ce vraiment un nouveau service ? Si c'est une variante d'un service existant, il peut réutiliser le port de ce service. Notamment, le port ne devrait pas indiquer la version du service, cette version doit être marquée dans le protocole (notez que POP est une exception à cette règle).
— Si ce service est expérimental (RFC 3692), pourquoi ne pas utiliser les ports réservés à ces usages expérimentaux (RFC 2780) ?
— A-t-il vraiment besoin d'un port bien connu, statique ? Ne pourrait-il pas utiliser un port dynamique, indiqué dans la configuration, ou dans le DNS (par exemple avec les SRV) ou dans l'identificateur (comme avec les URL cités plus haut), ou échangé via le protocole (comme fait SIP), ou avec un protocole dédié comme le "*portmapper*" cité plus haut ?

OK, on a déterminé qu'on avait vraiment besoin d'un numéro de port, on s'apprête à la demander à l'IANA. Mais quel numéro choisir ? Ce n'est pas une obligation. On peut ne pas indiquer de numéro particulier, et laisser l'IANA en choisir un. En revanche, il faut indiquer si le port doit être dans la plage Système ou dans la plage Utilisateur. La distinction est nettement moins importante aujourd'hui. Autrefois, quand l'Internet ne connectait que des grosses machines gérées professionnellement, la distinction servait à isoler les services critiques des autres. Sur ces grosses machines, l'utilisateur ordinaire ne pouvait pas écouter sur les ports de numéro inférieurs à 1 024 (par exemple, sur Unix, il faut être root pour cela) et cela garantissait que le service sur ces ports était « sérieux ». Aujourd'hui où certains systèmes ne font plus la différence et où, de toute façon, tout le monde est super-utilisateur de son Raspberry Pi, la différence a beaucoup moins de sens. Et cette distinction ne doit pas être utilisé pour la sécurité (cf. section 7.4) comme le faisait malheureusement rlogin. Néanmoins, la différence demeure, au moins comme un marqueur de la criticité du service. La plage Système étant petite et très demandée, les chances d'y obtenir un numéro de port sont plus faibles (il faut un examen par l'IETF ou bien une approbation par l'IESG, RFC 6335, section 8.1, et aussi le RFC 5226, sur la définition des politiques d'enregistrement).

Comme toute ressource limitée dans un espace commun, les numéros de port peuvent susciter des frictions. Une fois le logiciel déployé, il peut être difficile de changer le numéro de port de manière coordonné. D'où les deux règles :

— Ne pas déployer de code avec un numéro de port statique tant que celui-ci n'a pas été enregistré par l'IANA, afin d'éviter le squat (cf. plus loin),
— Si on utilise les ports expérimentaux du RFC 4727 (1 021 et 1 022), il ne faut le faire que dans un environnement qu'on contrôle bien, pour garantir qu'on pourra migrer ensuite vers le port final, sans bloquer éternellement ces numéros de port expérimentaux.

Et c'est quoi, ce risque de squat ? C'est l'utilisation d'un port statique sans l'avoir obtenu par les procédures normales. Cela peut être dû à un développeur pressé ou négligent qui écrit son code sans tenir compte des procédures, et le publie. Cela peut être aussi délibéré de la part de gens qui s'estiment au-dessus des règles et qui se réservent sans vergogne une part de la ressource commune. Dans les deux cas, c'est très gênant car, une fois le code largement déployé sur l'Internet, il sera vraiment difficile de le changer, ou même de savoir si le port est encore utilisé (voir aussi la section 7.9). Le squat est donc nettement condamné. Un exemple fameux de squat est l'enregistrement de CARP `<http://kerneltrap.org/node/2873>`.

La section 8 de notre RFC couvre les questions de sécurité liées aux ports. D'abord, il ne faut pas se fier au numéro de port. Rien ne garantit que ce qui circule sur le port 53 soit du DNS, et encore moins que ce soit du DNS correct (voir l'exemple plus haut avec tcpdump). Rien ne garantit qu'un paquet où

le port source est inférieur à 1 024 a vraiment été émis par une personne de confiance, et ainsi de suite. Les mesures classique d'authentification sont donc nécessaires si on veut vraiment se protéger.

Ensuite, un mot sur la vie privée. Le système des ports bien connus est l'antithèse de la vie privée puisqu'un simple coup d'œil à l'en-tête de couche 4, sans aller examiner le contenu applicatif, indique quel service on utilise. Bien sûr, ce n'est pas une garantie (paragraphe précédent...) mais cela donne déjà une indication. Autre indiscrétion liée aux ports : en envoyant des demandes de connexion TCP à une machine, vers les différents ports (comme ce que fait nmap), on peut savoir quels services elle offre, sans avoir besoin d'analyser les applications. C'est un problème, mais qui est fondamentalement lié à l'architecture de TCP/IP et qui est difficile à corriger (on peut toujours recourir à des trucs comme le "*port knocking*").

Moins directement utile mais passionnante, l'histoire des ports était présentée dans la section 3. Le terme est apparu pour la première fois dans le RFC 33, en 1970 (« "*We assume here that a process has several input-output paths which we will call ports.*" »). À l'époque, les couches 3 et 4 étaient encore mêlées, dans le protocole NCP. L'adresse désignant une machine, le port devait désigner une voie de communication d'un processus sur cette machine. (Notez qu'avec des adresses IP suffisamment longues et abondantes, comme ce que fournit IPv6 aujourd'hui, on pourrait se débarrasser complètement de la distinction adresse/port, un préfixe IP pourrait désigner une machine et chaque adresse une prise ou un processus. La séparation adresse/port est le souvenir d'une époque révolue.) Le RFC 37 et le RFC 38 ont ensuite précisé et modifié l'idée (« "*The destintation [sic] socket must be added to the header of each message on the data link. Presumably this would consist of 32 bits [ce sera finalement 16 bits] immediately after the header and before the marking.*" »). Puis le RFC 48, toujours en 1970, introduit l'idée d'« écoute » sur un port, lorsqu'un processus est en attente d'une requête. Et le RFC 61 est le premier à se demander comment on est censé connaître le port de destination d'un message, et à introduire le concept de « port bien connu » ("*well-known port*") comme le futur port 80 pour HTTP. Le RFC 76 s'attaque à la question posée, mais non résolue, par le RFC 61 et propose un annuaire permettant d'associer des noms de services à des numéros de port (le `/etc/services` de votre machine Unix en est le lointain descendant). « "*most permanently assigned devices and/or processes are known by standard mnemonic labels such as DSK (disk), LP (line printer), CR (card reader), TECO (PDP-10 text editor), etc.*" »

Une importante étape est ensuite franchie avec le RFC 333 en 1972, qui est le premier à suggérer que les ports de source et de destination servent (avec les adresses IP) à identifier une connexion, ce qui sera le choix final de TCP (RFC 793, en 1981, mais décrit deux ans avant avant dans le document IEN 112 <`https://www.rfc-editor.org/ien/ien112.txt`>). Le RFC 717 est le premier à indiquer des numéros de port officiellement affectés mais c'est le RFC 739, en 1977, qui avait généralisé l'idée (sous le nom de "*socket number*"). Dans ce RFC, telnet avait déjà le port 23, qu'il a gardé jusqu'à aujourd'hui, comme bien des protocoles depuis oubliés. Les procédures de l'époque étaient plus simples qu'aujourd'hui (« "*please contact Jon [Postel] to receive a number assignment*" »).

Quant au RFC 758, il était le premier à étendre le concept de port à TCP (les RFC précédents ne parlaient que de NCP). Certaines plages de numéros de ports étaient réservées à des usages spécifiques, mais son successeur, le RFC 820, n'a pas retenu cette distinction (en prime, le RFC 820 a étendu l'usage des ports à UDP). Avec le RFC 900 disparait la notation octale, avec le RFC 1340 (en 1992), la liste des « ports bien connus » va désormais de 0 à 1 023 et plus seulement jusq'à 255 (on y note l'apparition de 80, pour HTTP). Cette liste sera encore révisée dans le RFC 1700 puis remplacée en 2002 par un registre en ligne <`https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml`> à l'IANA, registre décrit par le RFC 3232. (Le RFC listant les ports était toujours en retard sur l'allocation réelle.)