RFC 7143 : iSCSI Protocol (Consolidated)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 avril 2014

Date de publication du RFC : Avril 2014

http://www.bortzmeyer.org/7143.html

Le protocole iSCSI permet à une machine d'accéder à des disques (ou autres dispositifs de stockage) situés en dehors, loin de l'atteinte des bus internes de la machine. iSCSI consiste à faire passer les traditionnelles requêtes SCSI sur IP, autorisant ainsi l'utilisation de tous les équipements réseaux qui font passer de l'IP, et dispensant ainsi d'acheter des équipements spécialisés dans les SAN, comme le très coûteux Fibre Channel. iSCSI était à l'origine normalisé dans plusieurs RFC, que ce nouveau document rassemble en une seule (très) grosse spécification.

Résultat, le RFC faisant autorité pour iSCSI est un document de 344 pages, un record ! Il remplace les anciennes normes, les RFC 3720 [1], RFC 3980, RFC 4850 et RFC 5048.

Si jamais vous envisagez de lire le RFC entier, je vous souhaite bon courage. Si vous voulez juste apprendre deux ou trois choses sur iSCSI, révisez d'abord un peu SCSI (les section 2.1 et 2.2 du RFC détaillent la terminologie, la 4.1 les concepts) : c'est un protocole client/serveur où le client (typiquement un ordinateur) se nomme l'initiateur et le serveur (typiquement un disque) la cible. L'initiateur envoie des commandes à la cible, qui répond. Le transport des commandes et des réponses peut se faire sur bien des supports, entre autres SAS, Parallel SCSI et FireWire. Ces supports traditionnels de SCSI ont généralement des faibles portées. iSCSI, au contraire, met les commandes et les réponses sur TCP, donc IP, et les cibles peuvent être à n'importe quelle distance de l'initiateur (même si, en pratique, initiateurs et cibles seront souvent dans le même réseau local).

SCSI peut aussi être utilisé pour parler à des lecteurs de bande, à des DVD, mais aussi, autrefois, à des imprimantes ou des "*scanners*". Chaque dispositif d'entrée/sortie SCSI se nomme un LU ("*Logical Unit*"). Une même cible peut comporter plusieurs LU, chacun ayant un numéro, le LUN ("*Logical Unit Number*").

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3720.txt

Les commandes sont regroupées en tâches. Chaque tâche est séquentielle (une commande après l'autre) mais un initiateur et une cible peuvent avoir plusieurs tâches en cours d'exécution parallèle.

Les données peuvent évidemment voyager dans les deux sens : si un ordinateur écrit sur un disque, l'initiateur (l'ordinateur) envoie des données à la cible (le disque). Si l'ordinateur lit un fichier, ce sera le contraire, les données iront de la cible vers l'initiateur. Le sens de voyage des données est toujours exprimé du point de vue de l'initiateur. Donc, du « trafic entrant » désigne des données allant de la cible vers l'initiateur (une lecture sur le disque, par exemple).

iSCSI consiste à envoyer les messages SCSI, baptisés PDU (pour "*Protocol data unit*"), sur TCP (port 3260 ou à la rigueur 860). Contrairement à ce qui se passe pour d'autres protocoles (comme SIP), données et commandes voyagent dans les mêmes connexions TCP. Pour diverses raisons, il peut y avoir **plusieurs** connexions TCP actives entre l'initiateur et la cible. On les appelle collectivement la **session** (équivalent du "*nexus*" en SCSI classique). Chaque session a son propre identificateur et chaque connexion TCP de la session a un identificateur de connexion. Des connexions TCP peuvent être ajoutées ou retirées en cours de session, augmentant ainsi la résilience.

La réponse à une commande donnée doit être envoyée sur la même connexion TCP que celle où la commande avait été reçue (« allégeance à la connexion »). Ainsi, si un initiateur demande une lecture de données, celles-ci arriveront sur la même connexion.

L'ordre des commandes est parfois essentiel en SCSI. Pour cela, elles sont numérotées ("*Command Sequence Number*", et cette numérotation est valable globalement pour toute la session), afin que la cible les exécute dans l'ordre. Il existe aussi d'autres numéros ("*Status Sequence Number*") qui ne servent que dans une connexion TCP donnée et sont plutôt utiliser en cas d'erreurs, pour associer un message d'erreur à une commande précise. (Le RFC 3783 décrit avec bien plus de détails le concept d'ordre dans iSCSI.)

À noter que les réponses aux commandes, en SCSI (et donc en iSCSI), ne respectent typiquement aucun ordre. Si une cible exécute la commande A puis la commande B, la réponse à B pourra arriver à l'initiateur avant la réponse à A (ce qui est logique, les opérations d'entrée/sortie pouvant avoir des durées très variables ; la commande B était peut-être très rapide à exécuter).

Les données, elles, ont un ordre. Si on lit 10 000 octets, on veut évidemment recevoir le premier octet des données en premier. D'où le "*Data Sequence Number*" qui se trouve dans les paquets de données et qui indique où se trouvent ces données dans le flux transmis.

Un autre point important : TCP fournit un service de flot d'octets continu. Il n'y a pas de séparation entre les PDU (les messages) iSCSI. Pour que le destinataire puisse donc savoir où se termine le PDU (et où commence le suivant), ceux-ci sont préfixés par leur longueur (comme dans beaucoup d'autres protocoles utilisant TCP, par exemple dans EPP et dans le DNS).

Les données peuvent être sensibles (confidentielles, par exemple) et iSCSI doit donc prévoir des mécanismes d'authentification. Ainsi, au début d'une session, l'initiateur se présente et prouve son identité. La cible doit aussi prouver qu'elle est bien la cible visée (authentification mutuelle). Il existe plusieurs mécanismes pour cela, décrits en détail en sections 6 et 12. La recommandation de ce RFC est de protéger les sessions avec IPsec (cf. section 9, ainsi que le RFC 3723).

Au fait, comme toujours en réseau, il faut des identificateurs pour désigner cibles et initiateurs. En iSCSI, il sont appelés **noms**. Contrairement au SCSI classique où les identificateurs n'ont besoin d'être uniques qu'à l'interieur d'un domaine très restreint (souvent un seul boîtier, parfois une salle machines),

iSCSI permet de faire parler des machines situées sur l'Internet et les noms doivent donc être mondialement uniques. (Pour une vue générale du nommage en iSCSI, voir le RFC 3721.) En outre, ces noms doivent (section 4.2.7.1) être stables sur le long terme, ne pas être dépendant de la localisation physique des machines, ni même du réseau où elles sont connectées, et le système de nommage doit être un système déjà existant : pas question de réinventer la roue ! À noter que ce sont des principes analogues à ceux des URN du RFC 1737. Les noms doivent en outre pouvoir être écrits en Unicode, canonicalisés par le profil stringprep de iSCSI, spécifié dans le RFC 3722, puis encodés en UTF-8, forme normale C.

iSCSI fournit trois formats pour atteindre ces objectifs. Un nom commence par un type indiquant quel format est utilisé, puis comprend le nom proprement dit. Les trois types sont :

— Nom de domaine, type `iqn`. Comme ceux-ci ne sont pas forcément stables sur le long terme (pour des raisons juridico-politiques, pas pour des raisons techniques), le format iSCSI ajoute une date indiquant un moment où le titulaire avait effectivement ce nom. C'est le même principe que les URI `tag:` du RFC 4151. Un nom valide comporte le nom de domaine (écrit à l'envers) et donc, on peut voir, par exemple, `iqn.2001-04.com.example:storage:diskarrays-sn-a8675309`, attribué par le titulaire de `example.com`.
— NAA ("*Network Address Authority*"), type `naa`, un système spécifique au monde Fibre Channel. Un exemple de nom NAA est `naa.52004567BA64678D`.
— EUI, type `eui`. un format de l'IEEE permettant de donner des noms uniques `<http://standards.ieee.org/regauth/oui>` aux machines. Par exemple, un nom iSCSI EUI peut être `eui.02004567A425678D`.

Il existe également des mécanismes de découverte des noms accessibles, décrits dans les RFC 3721 et RFC 4171. Notre RFC note que, malheureusement, certains de ces mécanismes dépendent de SLP (protocole décrit dans le RFC 2608) qui n'a jamais vraiment été implémenté et déployé.

Attention, ce RFC 7143 ne décrit que le transport de SCSI sur IP (en fait, sur TCP). SCSI, lui, reste décrit par ses normes originales (qui ne semblent plus accessibles en ligne `<http://www.t10.org/drafts.htm>`). Pour décrire le modèle, elles se servent d'UML, une technique qu'on trouve rarement dans un RFC, d'autant plus qu'elle repose sur de jolis dessins, qu'on ne peut pas facilement reproduire dans un RFC, avec leurs règles de formatage actuelles. Ce RFC comporte donc des diagrammes UML... en art ASCII (section 3).

iSCSI est décrit par une machine à nombre fini d'états et ladite machine figure en section 8.

Évidemment, les disques durs peuvent tomber en panne ou bien le réseau entre l'initiateur et la cible partir en vacances. Il faut donc prévoir des mécanismes de gestion des erreurs, et ils font l'objet de la section 7. C'est ainsi que, par exemple, une tâche qui échoue sur une connexion TCP peut être reprise sur une autre (qui peut passer par un chemin réseau différent). Notez que cette norme n'impose pas aux acteurs iSCSI de tous mettre en œuvre des techniques sophistiquées de récupération d'erreurs, juste de réagir proprement lorsque les erreurs se produisent. La section 7 est très détaillée : après tout, on ne veut pas perdre ses précieuses données juste parce que l'Internet a eu un hoquet pendant une opération d'écriture en iSCSI.

Mais comment détecte-t-on un problème sur une connexion TCP ? Cela peut être un message ICMP qui coupe la connexion mais aussi une absence prolongée de réponse à une commande, ou bien à un test iSCSI (la commande `NOP`, équivalente du ping IP) ou TCP (les "*keep-alives*" du RFC 1122, section 4.2.3.6).

Les pannes, c'est ennuyeux, mais les piratages, c'est pire. On ne veut pas qu'un méchant situé sur le trajet puisse lire nos données secrètes ou modifier ce qu'on écrit sur le disque. En SCSI traditionnel, une grande partie de la sécurité est physique : initiateur et cible sont dans le même boîtier (celui du serveur) ou à la rigueur dans la même armoire avec juste un SAN entre les deux. En iSCSI, on perd cette sécurité physique. Si l'initiateur et la cible sont connectés par le grand Internet, tout peut arriver sur le chemin.

La section 9, « "*Security Considerations*" », fait donc le tour des menaces et des contre-mesures. (À quoi il faut ajouter le RFC 3723.)

iSCSI peut être utilisé sans sécurité, comme l'ancien SCSI, si on est parfaitement sûr de la sécurité physique. Mais notre RFC déconseille de courir ce risque. iSCSI fournit deux mécanismes de sécurité et le RFC demande qu'on les utilise. Le premier est interne à iSCSI, c'est l'authentification réciproque des deux parties lors de l'établissement de la session. Le deuxième est externe, et c'est IPsec. Ces deux mécanismes doivent être mis en œuvre (même s'ils ne seront pas forcément activés par l'administrateur système, notamment pour IPsec, complexe et pas forcément utile si tout le monde est dans la même salle machines).

À noter qu'une limite d'IPsec est que les systèmes d'exploitation typiques ne permettent pas à une application de savoir si ses connexions sont protégées ou non par IPsec. Ainsi, une application qui voudrait faire une authentification par mot de passe en clair uniquement si la connexion est chiffrée ne peut pas être sûre et doit donc supposer le pire (une connexion non chiffrée). L'authentification interne ne doit donc jamais utiliser de mot de passe en clair. La méthode recommandée est CHAP (RFC 1994, mais avec quelques ajouts). Dans le futur, un mécanisme comme celui du RFC 5433 ou comme SRP - RFC 2945 - sera peut-être utilisé.

Le mécanisme interne n'assure que l'authentification réciproque. Il ne protège pas contre l'écoute, ni même contre la modification des messages en cours de route. Il ne doit donc être utilisé seul que si on est sûr que ces écoutes et ces modifications sont impossibles (par exemple si on a une bonne sécurité physique du réseau). Imaginez les conséquences d'une modification des données avant leur écriture sur le disque !

IPsec dans iSCSI doit suivre le RFC 3723 et le tout récent RFC 7146. Avec IPsec, les sessions iSCSI peuvent garantir la confidentialité et l'intégrité des données. À noter que les numéros de séquence IPsec ne sont par défaut que sur 32 bits. iSCSI verra probablement des débits élevés (par exemple plusieurs Gb/s) et ce numéro sera donc vite arrivé au bout. Le RFC demande donc les numéros de séquence étendus (64 bits) de la section 2.2.1 du RFC 4303.

Les programmeurs tireront profit de la section 10, qui rassemble des conseils utiles pour mettre en œuvre iSCSI sans se planter. Un des enjeux de iSCSI est la performance, vu les grandes quantités de données à traiter... et l'impatience des clients. Le protocole a été soigneusement conçu pour permettre d'éventuelles implémentations directement dans le matériel, et pour permettre le DDP ("*Direct Data Placement*", RFC 5041). Ainsi, on peut mettre iSCSI dans le "*firmware*", permettant le démarrage sur une cible iSCSI <http://www.intel.com/support/network/adapter/pro100/sb/CS-023748.htm>.

Parmi les points notés par cette section, la gestion des machines dotées de plusieurs cartes réseau (on peut en utiliser plusieurs pour la même session iSCSI), le délicat choix des délais d'attente maximaux (non normalisés, car dépendant de l'infrastructure, notamment matérielle), et l'ordre des commandes (une commande qui change l'état de la cible, par exemple un vidage - "*flush*" - des tampons, ne doit pas être envoyée comme commande immédiate, pour éviter qu'elle ne double d'autres commandes).

La section 11 décrit le format exact des PDU sur le câble. Si vous le désirez, on trouve des traces iSCSI sur pcapr <http://www.pcapr.net/browse?q=iscsi>.

Il existe de nombreuses mises en œuvre d'iSCSI et des programmeurs de nombreuses entreprises (Dell, EMC, Microsoft, NetApp, Red Hat, VMware, etc) ont participé à la création de ce RFC. Une liste partielle de mises en œuvre :

— Sur Linux, en cible <http://iscsitarget.sourceforge.net/>, et en initiateur <http://linux-iscsi.sourceforge.net/>. Pour l'administrateur système, consulter le HOWTO <http://iscsi-init.sourceforge.net/HOWTO.html>.
— Pour NetBSD : <ftp://ftp.netbsd.org/pub/NetBSD/misc/agc/>.

À part le gros changement éditorial (consolider plusieurs RFC en un seul), les changements de iSCSI dans ce RFC sont peu nombreux. Par exemple, une stricte prohibition des caractères de ponctuation dans les noms SCSI est devenue une forte mise en garde. La section 2.3 liste toutes les modifications.

Merci à Bertrand Petit pour sa relecture attentive.