

Adresses universelles (URL, *Uniform Resource Locators*)

Statut du présent Mémo

Le présent document spécifie un protocole de normalisation Internet pour la communauté Internet, et appelle à discussion et suggestions en vue de son amélioration. Prière de se reporter à l'édition en cours des "Internet Official Protocol Standards" (*normes officielles du protocole Internet*) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémo n'est pas soumise à restrictions.

Résumé

Le présent document spécifie une adresse universelle (URL, *Uniform Resource Locator*), la syntaxe et la sémantique des informations formalisées pour la localisation et l'accès à des ressources via l'Internet.

1. Introduction

Le présent document décrit la syntaxe et la sémantique pour une représentation de chaîne compacte pour une ressource disponible via l'Internet. Ces chaînes sont appelées "adresses universelles" (URL, *Uniform Resource Locators*).

Cette spécification est dérivée de concepts introduits par l'initiative mondiale d'informations sur la Toile mondiale (*World-Wide Web*), qui utilise de tels objets depuis 1990 et est décrite dans la RFC 1630 "Identifiants de ressources universels dans le WWW". La spécification des URL est destinée à satisfaire aux exigences exposées dans "Exigences fonctionnelles pour les localiseurs de ressources de l'Internet" [12].

Le présent document a été écrit par le groupe de travail URI de du Groupe de travail d'ingénierie de l'Internet (IETF). Les commentaires peuvent être adressés aux éditeurs, ou au groupe de travail URI <uri@bunyip.com>. Les discussions du groupe sont archivées à <URL:<http://www.acl.lanl.gov/URI/archive/uri-archive.index.html>>

2. Syntaxe générale d'URL

Tout comme il y a de nombreuses méthodes différentes pour accéder aux ressources, il existe plusieurs schémas pour décrire la localisation de telles ressources.

La syntaxe générique pour les URL fournit un cadre pour de nouveaux schémas à établir en utilisant des protocoles autres que ceux définis dans le présent document.

Les URL sont utilisés pour 'localiser' des ressources, en fournissant une identification abstraite de la localisation de la ressource. En ayant localisé une ressource, un système peut effectuer diverses opérations sur la ressource, telles que caractérisées par des mots comme 'access' (*accès*), 'update' (*mise à jour*), 'replace' (*remplacer*, 'find attributes' (*trouver les attributs*)). En général, seule la méthode 'access' a besoin d'être spécifiée pour un schéma d'URL.

2.1 Les parties principales d'un URL

Une description complète en BNF de la syntaxe d'URL figure à la Section 5.

En général, les URL sont écrits comme suit :

<scheme>:<scheme-specific-part>

Un URL contient le nom du schéma utilisé (<scheme>) suivi de deux points puis d'une chaîne (la <scheme-specific-part>) dont l'interprétation dépend du schéma.

Les noms de schéma consistent en une séquence de caractères. Les lettres en minuscule "a" à "z", les chiffres, et les caractères plus ("+"), point ((".")), et trait d'union ("-") sont admis. Pour améliorer la robustesse, les programmes qui interprètent les URL devraient traiter les lettres majuscules comme équivalentes aux lettres minuscules dans les schémas de noms (par exemple, permettre "HTTP" aussi bien que "http").

2.2 Question du codage des caractères d'URL

Les URL sont des séquences de caractères, c'est-à-dire, des lettres, des chiffres, et des caractères spéciaux. Un URL peut être représenté de diverses façons : par exemple, de l'encre sur du papier, ou une séquence d'octets dans un ensemble de caractères codés. L'interprétation d'un URL dépend seulement de l'identité des caractères utilisés.

Dans la plupart des schémas d'URL, les séquences de caractères dans les différentes parties d'un URL sont utilisées pour représenter des séquences d'octets utilisés dans les protocoles de l'Internet. Par exemple, dans le schéma ftp, le nom d'hôte, le nom de répertoire et le nom de fichier sont de telles séquences d'octets, représentés par des parties de l'URL. Au sein de ces parties, un octet peut être représenté par le caractère qui a cet octet comme code dans l'ensemble des caractères codés de l'US-ASCII [20].

De plus, les octets peuvent être codés par un triplet de caractères comportant le caractère "%" suivi par les deux chiffres hexadécimaux (de "0123456789ABCDEF") qui forment la valeur hexadécimale de l'octet. (Les caractères "abcdef" peuvent aussi être utilisés dans les codages hexadécimaux.)

Les octets doivent être codés s'ils n'ont pas de caractère graphique correspondant dans l'ensemble de caractères codés de l'US-ASCII, si l'utilisation du caractère correspondant n'est pas sûre, ou si le caractère correspondant est réservé pour quelque autre interprétation au sein du schéma d'URL particulier.

Pas de correspondant graphique en US-ASCII :

Les URL sont seulement écrits avec les caractères graphiques imprimables de l'ensemble de caractères codés de l'US-ASCII. Les octets hexadécimaux 80-FF ne sont pas utilisés en US-ASCII, et les octets hexadécimaux 00-1F et 7F représentent des caractères de contrôle ; ils doivent être codés.

Pas sûrs :

Des caractères peuvent n'être pas sûrs pour un certain nombre de raisons. Le caractère espace n'est pas sûr parce que des espaces signifiants peuvent disparaître et des espaces non signifiants peuvent être introduits lorsque des URL sont transcrits ou que la typographie change ou qu'ils sont soumis à un programme de traitement de texte. Les caractères "<" et ">" ne sont pas sûrs parce qu'ils sont utilisés comme délimiteurs autour des URL en texte libre ; les guillemets ("") sont utilisés pour délimiter les URL dans certains systèmes. Le caractère "#" n'est pas sûr et devrait toujours être codé parce qu'il est utilisé sur la toile mondiale et dans d'autres systèmes pour délimiter un URL d'un identifiant de fragment/référence principale qui pourrait le suivre. Le caractère "%" n'est pas sûr parce qu'il est utilisé pour coder d'autres caractères. D'autres caractères ne sont pas sûrs parce que les passerelles et autres agents de transport sont réputés modifier parfois de tels caractères. Ces caractères sont "{", "}", "|", "\", "^", "~", "[", "]", et "`".

Tous les caractères non sûrs doivent toujours être codés au sein d'un URL. Par exemple, le caractère "#" doit être codé au sein des URL même dans les systèmes qui n'ont normalement rien à voir avec les identifiants de fragment ou de référence principale, de sorte que si l'URL est copié dans un autre système qui ne les utilise pas, il ne soit pas nécessaire de changer le codage de l'URL.

Réservé :

De nombreux schémas d'URL réservent certains caractères pour une signification particulière : leur apparition dans la partie spécifique du schéma de l'URL a une sémantique précise. Si le caractère correspondant à un octet est réservé dans un schéma, l'octet doit être codé. Les caractères ";", "/", "?", ":", "@", "=", et "&" sont les caractères qui peuvent être réservés à une signification particulière au sein d'un schéma. Aucun autre caractère ne peut être réservé au sein d'un schéma.

Normalement, un URL a la même interprétation lorsqu'un octet est représenté par un caractère et quand il est codé. Cependant, ceci n'est pas vrai pour les caractères réservés : le codage d'un caractère réservé pour un schéma particulier peut changer la sémantique d'un URL.

Et donc, seuls les caractères alphanumériques, les caractères spéciaux "\$-_.+!*()'", et les caractères réservés utilisés pour leur objet particulier peuvent être utilisés non codés au sein d'un URL.

D'un autre côté, les caractères qu'il n'est pas obligatoire de coder (y compris les caractères alphanumériques) peuvent être codés au sein de la partie spécifique de schéma d'un URL, pour autant qu'ils ne sont pas utilisés pour un objet réservé.

2.3 Schémas hiérarchiques et liaisons qui s'y rapportent

Dans certains cas, les URL sont utilisés pour localiser des ressources qui contiennent des pointeurs sur d'autres ressources. Dans certains cas, ces pointeurs sont représentés comme des liens implicites où l'expression de la localisation de la seconde ressource est exprimée sous la forme de "au même endroit que celle-ci mais avec le chemin implicite suivant". Les liens implicites ne sont pas décrits dans le présent document. Cependant, l'utilisation de liens implicites dépend de l'URL d'origine qui contient une structure hiérarchique par rapport à laquelle le lien implicite se fonde.

Certains schémas d'URL (tels que le schéma ftp, http, et file) contiennent des noms qui peuvent être considérés comme hiérarchiques ; les composants de la hiérarchie sont séparés par "/".

3 Schémas spécifiques

La correspondance pour certains protocoles standard et expérimentaux existants est soulignée dans la définition de syntaxe en BNF. Des notes sur les protocoles particuliers suivent. Les schémas couverts sont :

ftp	File Transfer protocol (protocole de transfert de fichier)
http	Hypertext Transfer Protocol (protocole de transfert hypertexte)
gopher	protocole Gopher
mailto	adresse de messagerie électronique
news	nouvelles USENET
nntp	nouvelles USENET utilisant l'accès NNTP
telnet	référence à des sessions interactives
wais	serveurs d'information de zone étendue (<i>Wide Area Information Servers</i>)
file	noms de fichier spécifiques de l'hôte
prospero	service de répertoires Prospero

D'autres schémas pourront être spécifiés par des spécifications futures. La section 4 du présent document décrit comment de nouveaux schémas peuvent être enregistrés, et donne la liste d'un certain nombre de noms de schémas en cours de développement.

3.1 Syntaxe commune de schéma Internet

Alors que la syntaxe pour le reste de l'URL peut varier selon le schéma particulier choisi, les schémas d'URL qui impliquent l'utilisation directe d'un protocole fondé sur IP à un hôte spécifié sur l'Internet utilisent une syntaxe commune pour les données spécifiques du schéma :

```
//<utilisateur>:<mot-de-passe>@<hôte>:<port>/<chemin-d'url>
```

Certaines des parties "<utilisateur>:<mot-de-passe>@", ":", "<port>", et "/<chemin-d'url>", ou toutes, peuvent être exclues. Les données spécifiques du schéma commencent par une double barre oblique (*slash*) "/" pour indiquer qu'elles se conforment à la syntaxe commune de schéma Internet. Les différents composants obéissent aux règles suivantes :

utilisateur : nom facultatif d'utilisateur. Certains schémas (par exemple, ftp) permettent la spécification d'un nom d'utilisateur.

mot de passe : mot de passe facultatif. S'il est présent, il suit le nom d'utilisateur et est séparé par deux points.

Le nom d'utilisateur (et le mot de passe), s'ils sont présents, sont suivis par un arobase "@" (*qui se prononce "à" en français*). Dans les champs utilisateur et mot de passe, tout ":", "@", ou "/" doit être codé.

Noter qu'un nom d'utilisateur ou mot de passe vide est différent de pas de nom d'utilisateur ou de mot de passe ; il n'y a aucun moyen de spécifier un mot de passe sans spécifier de nom d'utilisateur. Par exemple, <URL:ftp://@host.com/>

a un nom d'utilisateur vide et pas de mot de passe, <URL:ftp://host.com/> n'a pas de nom d'utilisateur, alors que <URL:ftp://foo:@host.com/> a un nom d'utilisateur de "foo" et un mot de passe vide.

- host : Le nom de domaine pleinement qualifié d'un hôte réseau, ou son adresse IP comme un ensemble de quatre groupes de chiffres décimaux séparés par ".". Les noms de domaine pleinement qualifiés prennent la forme décrite au paragraphe 3.5 de la RFC 1034 [13] et au paragraphe 2.1 de la RFC 1123 [5] : une séquence d'étiquettes de domaine séparés par ".", chaque étiquette de domaine commençant et se terminant par un caractère alphanumérique et pouvant aussi contenir des caractères "-". L'étiquette de domaine la plus à droite ne commence jamais par un chiffre, ce qui distingue syntaxiquement tous les noms de domaine des adresses IP.
- port : Le numéro de port auquel se connecter. La plupart des schémas désignent des protocoles qui ont un numéro de port par défaut. Un autre numéro de port peut facultativement être fourni, en décimal, séparé de l'hôte par deux points. Si le port est omis, les deux points y sont quand même.
- chemin-d'url : Le reste de la localisation consiste en données spécifiques du schéma, et est connu sous le nom de "chemin-d'url". Il fournit les détails sur la façon d'accéder aux ressources spécifiées. Noter que le "/" entre l'hôte (ou le port) et le chemin-d'url ne fait PAS partie du chemin-d'url. La syntaxe de chemin-d'url dépend du schéma utilisé, ainsi que la manière de l'interpréter.

3.2 FTP

Le schéma d'URL FTP est utilisé pour désigner des fichiers et des répertoires sur des hôtes Internet accessibles en utilisant le protocole FTP (RFC959).

Un URL FTP suit la syntaxe décrite au paragraphe 3.1. Si :<port> est omis, le port par défaut est 21.

3.2.1 Nom et mot de passe FTP

On peut fournir un nom d'utilisateur et un mot de passe ; ils sont utilisés dans les commandes ftp "USER" et "PASS" après avoir d'abord établi la connexion avec le serveur FTP. Si aucun nom d'utilisateur ni mot de passe n'est fourni et qu'il en est demandé par le serveur FTP, les conventions pour FTP "anonyme" doivent être utilisées, comme suit :

- Le nom d'utilisateur "anonyme" est fourni.
- Le mot de passe est fourni comme l'adresse de messagerie électronique Internet de l'utilisateur final accédant à la ressource.
- Si l'URL fournit un nom d'utilisateur mais pas de mot de passe, et que le serveur distant demande un mot de passe, le programme interprétant l'URL FTP devrait en demander un de la part de l'utilisateur.

3.2.2 chemin d'url FTP

Le chemin d'url d'un URL FTP a la syntaxe suivante :

```
<cwd1>/<cwd2>/.../<cwdN>/<nom>;type=<typecode>
```

où <cwd1> à <cwdN> et <nom> sont des chaînes (qui peuvent être codées) et <typecode> est un des caractères "a", "i", ou "d". La partie ";type=<typecode>" peut être omise. Les parties <cwdx> et <nom> peuvent être vides. La totalité du chemin d'url peut être omise, y compris le "/" le délimitant du préfixe qui contient le nom d'utilisateur, le mot de passe, l'hôte, et le port.

Le chemin d'url est interprété comme une série de commandes FTP comme suit :

- Chacun des éléments <cwd> est à fournir, à la suite, comme argument d'une commande de changement de répertoire de travail (CWD, *change working directory*).
- Si le typecode est "d", effectuer une commande NLST (*name list*) avec <name> comme l'argument, et interpréter le résultat comme un listing de répertoires de fichiers.
- Autrement, effectuer une commande TYPE avec <typecode> comme argument, puis accéder au fichier dont le nom est <name> (par exemple, en utilisant la commande RETR.)
- Au sein d'un composant name ou CWD, les caractères "/" et ";" sont réservés et doivent être codés. Les composants sont décodés avant leur utilisation par le protocole FTP. En particulier, si la séquence FTP appropriée pour accéder à un fichier particulier requiert la fourniture d'une chaîne contenant un "/" comme argument d'une commande CWD ou RETR, il est nécessaire de coder chaque "/".

Par exemple, l'URL <URL:ftp://myname@host.dom/%2Fetc/motd> est interprété par FTP-ing comme "host.dom", s'enregistrant comme "myname" (invitant à insérer un mot de passe s'il est demandé), puis exécutant "CWD /etc" et ensuite "RETR motd". Cela a une signification différente de <URL:ftp://myname@host.dom/etc/motd> qui aurait comme commande "CWD etc" puis "RETR motd" ; la commande initiale "CWD" peut être exécutée par rapport au répertoire par défaut pour "myname". D'un autre côté, l'<URL:ftp://myname@host.dom//etc/motd>, aurait la commande "CWD" avec un argument nul, puis "CWD etc", et puis "RETR motd".

Les URL FTP peuvent aussi être utilisés pour d'autres opérations ; par exemple, il est possible de mettre à jour un fichier sur un serveur de fichiers distant, ou de déduire des informations sur lui à partir des listes du répertoire. Le mécanisme qui permet de le faire n'est pas décrit ici.

3.2.3 Le typecode FTP est facultatif

La partie ;type=<typecode> toute entière d'un URL FTP est facultative. Si elle est omise, le programme client qui interprète l'URL doit supputer le mode approprié à utiliser. En général, le type de contenu de données d'un fichier peut seulement être deviné par le nom, par exemple, d'après le suffixe du nom ; le code de type approprié à utiliser pour transférer le fichier peut alors être déduit du contenu de données du fichier.

3.2.4 Hiérarchie

Pour certains systèmes de fichiers, le "/" utilisé pour noter la structure hiérarchique de l'URL correspond au délimiteur utilisé pour construire une hiérarchie de nom de fichiers, et donc, le nom de fichier va sembler similaire au chemin d'URL. Cela ne signifie PAS que l'URL est un nom de fichier Unix.

3.2.5 Optimisation

Les clients qui accèdent à des ressources via FTP peuvent employer des méthodes heuristiques supplémentaires pour optimiser l'interaction. Pour certains serveurs FTP, par exemple, il peut être raisonnable de garder ouverte la connexion de contrôle lors de l'accès à plusieurs URL sur le même serveur. Cependant, il n'y a pas de modèle hiérarchique commun au protocole FTP, de sorte que si une commande de changement de répertoire a été donnée, il est impossible en général d'en déduire quelle séquence devrait être donnée pour naviguer vers un autre répertoire pour une seconde restitution, si les chemins sont différents. Le seul algorithme fiable est pour déconnecter et rétablir la connexion de contrôle.

3.3 HTTP

Le schéma d'URL HTTP sert à désigner des ressources Internet accessibles en utilisant HTTP (Protocole de transfert hypertexte).

Le protocole HTTP est spécifié par ailleurs. La présente spécification décrit seulement la syntaxe des URL HTTP.

Un URL HTTP prend la forme :

http://<hôte>:<port>/<chemin>?<searchpart>

où <host> et <port> sont décrits au paragraphe 3.1. Si :<port> est omis, le port par défaut est 80. Aucun nom d'utilisateur ni mot de passe n'est admis. <chemin> est un sélecteur HTTP, et <searchpart> est une chaîne d'interrogation. Le <chemin> est facultatif, tout comme <searchpart> et le "?" qui le précède. Si ni <chemin> ni <searchpart> n'est présent, le "/" peut aussi être omis.

Dans les composants <chemin> et <searchpart>, "/", ";", "?" sont réservés. Le caractère "/" peut être utilisé dans HTTP pour désigner une structure hiérarchique.

3.4 GOPHER

Le schéma d'URL Gopher sert à désigner des ressources Internet accessibles en utilisant le protocole Gopher.

Le protocole Gopher de base est décrit dans la RFC 1436 et prend en charge des éléments et collections d'éléments (répertoires). Le protocole Gopher+ est un ensemble d'extensions compatibles en amont avec le protocole Gopher de base et il est décrit en [2]. Gopher+ prend en charge l'association d'ensembles arbitraires d'attributs et de

représentations de données de remplacement avec des éléments Gopher. Les URL Gopher accommodent à la fois les éléments et attributs d'éléments Gopher et Gopher+.

3.4.1 Syntaxe d'URL Gopher

Un URL Gopher prend la forme :

```
gopher://<hôte>:<port>/<chemin-gopher>
```

où <chemin-gopher> est d'un de

```
<gophertype><selector>
<gophertype><selector>%09<search>
<gophertype><selector>%09<search>%09<gopher+_string>
```

Si <port> est omis, le port par défaut est 70. <gophertype> est un champ d'un seul caractère pour noter le type de ressource Gopher à laquelle l'URL se réfère. Le <chemin-gopher> entier peut aussi être vide, auquel cas le "/" de délimitation est aussi facultatif et le <gophertype> prend la valeur par défaut de "1".

<selector> est la chaîne de sélecteur de Gopher. Dans le protocole Gopher, les chaînes de sélecteur Gopher sont une séquence d'octets qui peuvent contenir tous les octets sauf l'hexadécimal 09 (HT ou tab en US-ASCII), le 0A hexadécimal (caractère LF en US-ASCII), et le 0D (caractère CR en US-ASCII).

Les clients Gopher spécifient quel élément récupérer en envoyant la chaîne de sélecteur Gopher à un serveur Gopher.

Aucun caractère n'est réservé au sein de <chemin-gopher>.

Noter que certaines chaîne <selector> de Gopher commencent par une copie du caractère <gophertype>, auquel cas ce caractère apparaîtra deux fois consécutivement. La chaîne de sélecteur Gopher peut être une chaîne vide ; c'est la façon dont les clients Gopher se réfèrent au répertoire de niveau supérieur sur un serveur Gopher.

3.4.2 Spécification des URL pour les moteurs de recherche Gopher

Si l'URL se réfère à une recherche à soumettre à un moteur de recherche Gopher, le sélecteur est suivi par une tabulation codée (%09) et par la chaîne de recherche. Pour soumettre une recherche à un moteur de recherche Gopher, le client Gopher envoie la chaîne <selector> (après décodage), une tabulation et la chaîne de recherche au serveur Gopher.

3.4.3 Syntaxe d'URL pour les éléments Gopher+

Les URL pour les éléments Gopher+ ont une seconde tabulation codée (%09) et une chaîne Gopher+. Noter que dans ce cas, la chaîne %09<search> doit être fournie, bien que l'élément <search> puisse être une chaîne vide.

La chaîne <gopher+_string> sert à représenter les informations nécessaires pour la récupération de l'élément Gopher+. Les éléments Gopher+ peuvent avoir des présentations de remplacement, des ensembles arbitraires d'attributs, et peuvent avoir des formes électroniques associées.

Pour récupérer les données associées à un URL Gopher+, un client se connectera au serveur et enverra le sélecteur Gopher, suivi par une tabulation et la chaîne de recherche (qui peut être vide), suivi par une tabulation et les commandes Gopher+.

3.4.4 Représentation des données Gopher+ par défaut

Lorsqu'un serveur Gopher retourne une liste de répertoires à un client, les éléments Gopher+ sont étiquetés avec un "+" (notant un élément Gopher+) ou un "?" (notant un élément Gopher+ qui a une forme +ASK associée). Un URL Gopher avec une chaîne Gopher+ consistant seulement en un "+" se réfère à la représentation par défaut (représentation de données) de l'élément alors qu'une chaîne Gopher+ contenant seulement un "?" se réfère à un élément avec une forme électronique Gopher associée.

3.4.5 Éléments Gopher+ avec formes électroniques

Les éléments Gopher+ qui ont un +ASK associé (c'est-à-dire des éléments Gopher+ étiquetés avec un "?") requièrent du client qu'il aille chercher l'attribut+ASK de l'élément pour obtenir la définition de forme, et demander ensuite à l'utilisateur de remplir la forme et de retourner les réponses de l'utilisateur avec la chaîne de sélecteur pour récupérer l'élément. Les clients Gopher+ savent comment faire cela mais dépendent de l'étiquette "?" dans la description

d'élément Gopher+ pour savoir quand traiter ce cas. Le "?" sert dans la chaîne Gopher+ à la cohérence avec l'utilisation de ce symbole dans le protocole Gopher+.

3.4.6 Collections d'attribut d'élément Gopher+

Pour se référer aux attributs Gopher+ d'un élément, la chaîne Gopher+ de l'URL Gopher consiste en "!" ou "\$". "!" se réfère à l'ensemble des attributs d'un élément Gopher+. "\$" se réfère à tous les attributs d'élément pour tous les éléments dans un répertoire Gopher.

3.4.7 Référence à des attributs Gopher+ spécifiques

Pour se référer à des attributs spécifiques, la chaîne gopher+ de l'URL est "!<attribute_name>" ou "\$<attribute_name>". Par exemple, pour se référer à l'attribut qui contient le résumé d'un élément, la chaîne gopher+_string serait "+ABSTRACT".

Pour se référer à plusieurs attributs, la chaîne gopher+_string consiste en noms d'attribut séparés par des espaces codés. Par exemple, "+ABSTRACT%20+SMELL" se réfère aux attributs +ABSTRACT et +SMELL d'un élément.

3.4.8 Syntaxe d'URL pour Gopher+ en représentations de remplacement

Gopher+ permet des représentations de remplacement de données facultatives (*alternate views*) des éléments. Pour restituer une représentation de remplacement de Gopher+, un client Gopher+ envoie la représentation appropriée et l'identifiant de langage (trouvé dans l'attribut +VIEW de l'élément). Pour se référer à une représentation de remplacement spécifique de Gopher+, la chaîne Gopher+ de l'URL devrait être de la forme :

+<view_name>%20<language_name>

Par exemple, une chaîne Gopher+ de "+application/postscript%20Es_ES" se réfère à une représentation de remplacement postscript en langue espagnole d'un élément Gopher+.

3.4.9 Syntaxe d'URL pour formes électroniques de Gopher+

La chaîne gopher+_string pour un URL qui se réfère à un élément référencé par une forme électronique Gopher+ (un bloc ASK) rempli avec des valeurs spécifiques est une version codée de ce que le client envoie au serveur. La chaîne gopher+_string est de la forme :

+%091%0D%0A+-1%0D%0A<ask_item1_value>%0D%0A<ask_item2_value>%0D%0A.%0D%0A

Pour restituer cet élément, le client Gopher envoie :

```
<a_gopher_selector><tab>+<tab>1<cr><lf>
+-1<cr><lf>
<ask_item1_value><cr><lf>
<ask_item2_value><cr><lf>
.<cr><lf>
```

au serveur Gopher.

3.5 MAILTO

Le schéma d'URL mailto sert à désigner l'adresse de messagerie Internet d'un individu ou d'un service. Aucune information supplémentaire autre qu'une adresse de messagerie Internet n'est présente ou impliquée.

Un URL mailto prend la forme :

mailto:<rfc822-addr-spec>

où <rfc822-addr-spec> est (le codage d'une) addr-spec, comme spécifié dans la RFC 822 [6]. Au sein des URL mailto, il n'y a pas de caractères réservés.

Noter que le signe pourcent ("%") est utilisé de façon ordinaire dans les adresses de la RFC 822 et doit être codé.

A la différence de nombreux URL, le schéma mailto ne représente pas un objet de données accessibles directement ; il ne désigne en aucune façon un objet. Il a une utilisation différente de celle du type message/corps extérieur dans MIME.

3.6 NEWS

Le schéma d'URL news sert à se référer soit à des groupes de nouvelles soit à des articles individuels des nouvelles USENET, comme spécifié dans la RFC 1036.

Un URL news prend une des deux formes :

```
news:<newsgroup-name>
news:<message-id>
```

Un <newsgroup-name> est un nom hiérarchique de période délimitée, comme "comp.infosystems.www.misc". Un <message-id> correspond au Message-ID du paragraphe 2.1.5 de la RFC 1036, sans le "<" et ">" de clôture ; il prend la forme <unique>@<full_domain_name>. Un identifiant de message peut être distingué d'un nom de groupe de nouvelles par la présence du caractère arobase "@". Aucun caractère supplémentaire n'est réservé parmi les composants d'un URL news.

Si <newsgroup-name> est "*" (comme dans <URL:news:*>), il est utilisé pour se référer à "tout groupe news disponible".

Les URL news sont inhabituels en ce que par eux-mêmes, ils ne contiennent pas d'informations suffisantes pour localiser une ressource, mais sont plutôt indépendants de la localisation.

3.7 NNTP

Le schéma d'URL nntp est une méthode de remplacement pour référencer des articles de nouvelles, utiles pour spécifier des articles de nouvelles à partir de serveurs NNTP (RFC 977).

Un URL nntp prend la forme :

```
nntp://<hôte>:<port>/<newsgroup-name>/<article-number>
```

où <host> et <port> sont tels que décrits au paragraphe 3.1. Si <port> est omis, le port par défaut est 119.

Le <newsgroup-name> est le nom du groupe, alors que <article-number> est l'identifiant numérique de l'article au sein de ce newsgroup.

Noter qu'alors que les URL nntp: spécifient une localisation unique pour la ressource d'article, la plupart des serveurs NNTP actuellement sur l'Internet sont configurés aujourd'hui pour ne permettre l'accès qu'aux clients locaux, et donc les URL nntp ne désignent pas des ressources accessibles mondialement. Et donc, la forme d'URL news: est de préférence une façon d'identifier des articles de nouvelles.

3.8 TELNET

Le schéma d'URL Telnet sert à désigner des services interactifs auxquels on peut accéder par le protocole Telnet.

Un URL telnet prend la forme :

```
telnet://<utilisateur>:<mot-de-passe>@<hôte>:<port>/
```

comme spécifié au paragraphe 3.1. Le caractère final "/" peut être omis.

Si <port> est omis, le port par défaut est 23. Le <mot-de-passe> peut être omis, ainsi que toute la partie <utilisateur>:<mot-de-passe> .

Cet URL ne désigne pas un objet de données, mais plutôt un service interactif. Les services interactifs à distance varient considérablement dans les moyens par lesquels ils permettent l'enregistrement à distance ; en pratique, le <utilisateur> et le <mot-de-passe> fournis sont seulement conseillés : les clients qui donnent l'accès à des URL telnet conseillent rarement l'utilisation du nom d'utilisateur et du mot de passe suggérés.

3.9 WAIS

Le schéma d'URL WAIS sert à désigner des bases de données WAIS, des recherches, ou des documents individuels disponibles à partir d'une base de données WAIS. WAIS est décrit en [7]. Le protocole WAIS est décrit dans la RFC 1625 [17]. Bien que le protocole WAIS soit fondé sur Z39.50-1988, le schéma d'URL WAIS n'est pas destiné à être utilisé avec des services Z39.50 arbitraires.

Un URL WAIS prend une des formes suivantes :

```
wais://<hôte>:<port>/<database>
wais://<hôte>:<port>/<database>?<search>
wais://<hôte>:<port>/<database>/<wtype>/<wpath>
```

où <hôte> et <port> sont décrits au paragraphe 3.1. Si <port> est omis, le port par défaut est 210. La première forme désigne une base de données WAIS qui est disponible pour des recherches. La seconde forme désigne une recherche particulière. <database> est le nom de la base de données WAIS qui va être interrogée.

La troisième forme désigne un document particulier au sein d'une base de données WAIS à restituer. Dans cette forme <wtype> est la désignation WAIS du type de l'objet. De nombreuses mises en œuvre WAIS exigent qu'un client connaisse le "type" d'un objet avant la restitution, le type étant retourné avec l'identifiant d'objet interne dans la réponse à la recherche. Le <wtype> est inclus dans l'URL afin de permettre au client qui interprète l'URL d'avoir des informations adéquates pour récupérer effectivement le document.

Le <wpath> d'un URL WAIS consiste en un identifiant de document WAIS, codé en tant que de besoin en utilisant la méthode décrite au paragraphe 2.2. L'identifiant de document WAIS est traité de façon opaque ; il ne peut être décomposé que par le serveur qui l'a produit.

3.10 FILES

Le schéma d'URL file est utilisé pour désigner des fichiers accessibles sur un ordinateur hôte particulier. Ce schéma, à la différence de la plupart des autres schémas d'URL, ne désigne pas une ressource universellement accessible sur l'Internet.

Un URL file prend la forme :

```
file://<hôte>/<chemin>
```

où <hôte> est le nom de domaine pleinement qualifié du système sur lequel le <chemin> est accessible, et <chemin> est un chemin de répertoire hiérarchique de la forme <répertoire>/<répertoire>/.../<nom>.

Par exemple, un fichier VMS

```
DISK$USER:[MY.NOTES]NOTE123456.TXT
```

peut devenir

```
<URL:file://vms.host.edu/disk$user/my/notes/note12345.txt>
```

Dans un cas particulier, <hôte> peut être la chaîne "localhost" ou la chaîne vide ; ceci est interprété comme 'la machine à partir de laquelle l'URL est interprété'.

Le schéma d'URL file est inhabituel en ce qu'il ne spécifie pas un protocole Internet ou une méthode d'accès pour de tels fichiers ; comme tel, son utilité dans les protocoles réseau entre hôtes est limitée.

3.11 PROSPERO

Le schéma d'URL Prospero est utilisé pour désigner des ressources auxquelles on accède via le service d'annuaire Prospero. Le protocole Prospero est décrit en [14].

Les URL Prospero prennent la forme :

```
prospero://<hôte>:<port>/<hsoname>;<champ>=<valeur>
```

où <hôte> et <port> sont comme décrit au paragraphe 3.1. Si <port> est omis, le port par défaut est 1525. Aucun nom d'utilisateur ni mot de passe n'est admis.

Le <hsoname> est le nom d'objet spécifique de l'hôte dans le protocole Prospero, convenablement codé. Ce nom est opaque et il est interprété par le serveur Prospero. Le point-virgule ";" est réservé et peut ne pas apparaître si on ne met pas le <hsoname> entre guillemets.

Les URL Prospero sont interprétés en contactant un serveur de répertoires Prospero sur l'hôte et le port spécifiés pour déterminer les méthodes d'accès appropriées pour une ressource, qui peuvent être eux-mêmes représentés comme des URL différents. Les liens externes de Prospero sont représentés comme des URL de la méthode d'accès sous jacente et ne sont pas représentés comme des URL Prospero.

Noter qu'une barre oblique "/" peut apparaître dans le <hsoname> sans guillemets et aucune signification ne peut être supposée par l'application. Bien que les barres obliques puissent indiquer une structure hiérarchique sur le serveur, une telle structure n'est pas garantie. Noter que de nombreux <hsoname> commencent par une barre oblique, auquel cas l'hôte ou port sera suivi par une double barre oblique : la barre oblique de la syntaxe d'URL, suivie par la barre oblique initiale provenant du <hsoname>. (Par exemple, <URL:prospero://host.dom//pros/nom> désigne un <hsoname> de "/pros/nom".)

De plus, après le <hsoname>, des champs et de valeurs facultatifs associés à une liaison Prospero peuvent être spécifiés au titre de l'URL. Lorsqu'ils sont présents, chaque paire de champ/valeur est séparée de l'autre et du reste de l'URL par un ";" (point-virgule). Le nom du champ et sa valeur sont séparés par un "=" (signe égal). S'ils sont présents, ces champs servent à identifier la cible de l'URL. Par exemple, le champ OBJET-VERSION peut être spécifié de façon à identifier une version spécifique d'un objet.

4 Enregistrement de nouveaux schémas

Un nouveau schéma peut être introduit en définissant une correspondance avec une syntaxe d'URL conforme, en utilisant un nouveau préfixe. Les URL pour des schémas expérimentaux peuvent être utilisés par accord mutuel entre les parties. Les noms de schémas commençant par les caractères "x-" sont réservés pour les expérimentations.

L'Autorité d'allocation des numéros de l'Internet (IANA) entretiendra un registre des schémas d'URL. Toute soumission de nouveau schéma d'URL doit inclure une définition d'un algorithme d'accès aux ressources au sein de ce schéma et la syntaxe de représentation d'un tel schéma.

Les schémas d'URL doivent avoir une utilité et une praticabilité démontrables. On peut fournir une telle démonstration via une passerelle qui fournit des objets dans le nouveau schéma pour les clients qui utilisent un protocole existant. Si le nouveau schéma ne localise pas les ressources qui sont des objets de données, les propriétés des noms dans le nouvel espace doivent être clairement définies.

Les nouveaux schémas pourraient essayer de suivre les mêmes conventions syntaxiques que les schémas existants, lorsque c'est approprié. Il est vraisemblablement recommandé que, lorsqu'un protocole permet la restitution par URL, le logiciel client soit provisionné pour être configuré à utiliser des localiseurs de passerelle spécifiques pour un accès indirect à travers les nouveaux schémas de nommage.

Les schémas suivants ont été proposés à diverses époques, mais le présent document ne définit pas leur syntaxe ou leur utilisation actuelle. Il est suggéré que l'IANA réserve leurs noms de schéma pour une future définition :

afs	Noms mondiaux de fichiers Andrew File System.
Mid	Identifiants de message pour messagerie électronique.
cid	Identifiants de contenu pour parties de corps MIME.
Nfs	Noms de fichiers Network File System (NFS).
tn3270	Sessions d'émulation 3270 interactives.
Mailserver	Accès aux données disponibles à partir de serveurs de messagerie.
z39.50	Accès aux services ANSI Z39.50.

5 BNF pour schémas d'URL spécifiques

Ce qui suit est une description de style BNF de la syntaxe d'adresse universelle, utilisant les conventions de la RFC 822, excepté que "|" sert à désigner les alternatives, et les crochets [] entourent les éléments facultatifs ou répétés. En bref, les explications sont citées avec "", les éléments facultatifs sont entre [crochets], et des éléments peuvent être précédés de <n>* pour désigner n ou plus répétitions de l'élément suivant ; la valeur par défaut de n est 0.

; La forme générique d'un URL est :
 genericurl = scheme ":" schemepart

; Les schémas spécifiques prédéfinis sont définis ici ; les nouveaux schémas peuvent être enregistrés auprès de IANA

url = httpurl | ftpurl | newsur | nntpurl | telneturl | gopherurl | waisurl | mailtourl | fileurl | prosperourl | otherurl

; les nouveaux schémas suivent la syntaxe générale otherurl = genericurl

; le schéma est en minuscules ; les interpréteurs devraient utiliser le schéma ignorer la casse

= 1*[lowalpha | digit | "+" | "-" | "."]

schemepart = *xchar | ip-schemepart

; schemeparts d'URL pour protocoles fondés sur ip:

ip-schemepart = "://" login ["/" urlpath]

login = [user [":" password] "@"] hostport

hostport = host [":" port]

host = hostname | hostnumber

hostname = *[domainlabel "."] toplabel

domainlabel = alphanum | alphanum *[alphanum | "-"] alphanum

toplabel = alpha | alpha *[alphanum | "-"] alphanum

alphanum = alpha | digit

hostnumber = digits "." digits "." digits "." digits

port = digits

user = *[uchar | ";" | "?" | "&" | "="]

password = *[uchar | ";" | "?" | "&" | "="]

urlpath = *xchar ; dépend du protocole, voir le paragraphe 3.1

; Les schémas prédéfinis :

; FTP (voir aussi la RFC 959)

ftpurl = "ftp://" login ["/" fpath [";type=" ftptype]]

fpath = fsegment *["/" fsegment]

fsegment = *[uchar | "?" | ":" | "@" | "&" | "="]

ftptype = "A" | "I" | "D" | "a" | "i" | "d"

; FILE

fileurl = "file://" [host | "localhost"] "/" fpath

; HTTP

httpurl = "http://" hostport ["/" hpath ["?" search]]

hpath = hsegment *["/" hsegment]

hsegment = *[uchar | ";" | ":" | "@" | "&" | "="]

search = *[uchar | ";" | ":" | "@" | "&" | "="]

; GOPHER (voir aussi la RFC 1436)

gopherurl = "gopher://" hostport [/ [gtype [selector ["%09" search ["%09" gopher+_string]]]]]]

gtype = xchar

selector = *xchar

gopher+_string = *xchar

; MAILTO (voir aussi la RFC 822)

mailtourl = "mailto:" encoded822addr

encoded822addr = 1*xchar ; mieux défini dans la RFC 822

; NEWS (voir aussi la RFC 1036)

newsurl = "news:" grouppart

grouppart = "*" | group | article

group = alpha *[alpha | digit | "-" | "." | "+" | "_"]

article = 1*[uchar | ";" | "/" | "?" | ":" | "&" | "="] "@" host

; NNTP (voir aussi la RFC 977)

nntpurl = "nntp://" hostport "/" group ["/" digits]

; TELNET

telneturl = "telnet://" login ["/"]

; WAIS (voir aussi la RFC 1625)

```

waisurl      = waisdatabase | waisindex | waisdoc
waisdatabase = "wais:/" hostport "/" database
waisindex    = "wais:/" hostport "/" database "?" search
waisdoc      = "wais:/" hostport "/" database "/" wtype "/" wpath
database     = *uchar
wtype        = *uchar
wpath        = *uchar

```

; PROSPERO

```

prosperourl  = "prospero:/" hostport "/" ppath *[ fieldspec ]
ppath        = psegment *[ "/" psegment ]
psegment     = *[ uchar | "?" | ":" | "@" | "&" | "=" ]
fieldspec    = ";" fieldname "=" fieldvalue
fieldname    = *[ uchar | "?" | ":" | "@" | "&" ]
fieldvalue   = *[ uchar | "?" | ":" | "@" | "&" ]

```

; Définitions diverses

```

lowalpha     = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" |
"v" | "w" | "x" | "y" | "z"
hialpha      = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" |
"T" | "U" | "V" | "W" | "X" | "Y" | "Z"
alpha        = lowalpha | hialpha
digit        = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
safe         = "$" | "-" | "_" | "." | "+"
extra        = "!" | "*" | "" | "(" | ")" | ","
national     = "{" | "}" | "|" | "\" | "^" | "~" | "[" | "]" | "`"
punctuation  = "<" | ">" | "#" | "%" | "<"
reserved     = ";" | "/" | "?" | ":" | "@" | "&" | "="
hex          = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"
escape       = "%" hex hex
unreserved   = alpha | digit | safe | extra
ucha         = unreserved | escape
xchar        = unreserved | reserved | escape
digits       = 1*digit

```

6 Considérations sur la sécurité

Le schéma d'URL ne pose pas par lui-même de menace sur la sécurité. Les utilisateurs doivent être avertis qu'il n'y a pas de garantie générale qu'un URL qui pointe à un moment sur un objet donné continue à le faire, et peut même après un certain temps pointer sur un objet différent du fait du mouvement des objets sur les serveurs.

Une menace sur la sécurité en relation avec les URL est qu'il est parfois possible de construire un URL de telle sorte qu'une tentative d'effectuer une opération aussi innocente que la restitution de l'objet va en fait causer la survenance d'une opération distante potentiellement dommageable. L'URL non sûr est typiquement construit en spécifiant un numéro de port autre que celui réservé pour le protocole réseau en question. Le client contacte involontairement un serveur qui est en fait sur un protocole différent. Le contenu de l'URL contient des instructions qui, lorsqu'elles sont interprétées conformément à cet autre protocole, causent une opération inattendue. Un exemple a été l'utilisation des URL gopher pour causer l'envoi d'un message grossier via un serveur SMTP. Il faut faire attention lorsqu'on utilise tout URL qui spécifie un numéro de port autre que le port par défaut pour le protocole, spécialement lorsque c'est un nombre dans l'espace réservé.

Il faut faire attention lorsque les URL contiennent des délimiteurs codés incorporés pour un protocole donné (par exemple, les caractères CR et LF pour les protocoles telnet) que ceux-ci ne soient pas codés avant la transmission. Cela violerait le protocole mais pourrait être utilisé pour simuler une opération ou paramètre supplémentaire, causant aussi une opération inattendue potentiellement dommageable à distance.

L'utilisation d'URL contenant des mots de passe qui devraient être secrets est clairement déconseillée.

7 Remerciements

Ce document s'appuie sur le concept de WWW de base (RFC 1630) et sur d'abondantes discussions de ces questions par de nombreuses personnes dans le réseau. La discussion a été particulièrement stimulée par les articles de Clifford Lynch, Brewster Kahle [10] et Wengyik Yeong [18]. Des contributions de John Curran, Clifford Neuman, Ed Vielmetti et plus tard du groupe de travail de l'IETF URL BOF and URI ont été incorporées.

Plus récemment, des relectures attentives et des commentaires de la part de Dan Connolly, Ned Freed, Roy Fielding, Guido van Rossum, Michael Dolan, Bert Bos, John Kunze, Olle Jarnefors, Peter Svanberg et de nombreux autres ont aidé à préciser cette RFC.

APPENDICE : Recommandations pour les URL dans leur contexte

Les URI, incluant les URL, sont destinés à être transmis à travers des protocoles qui fournissent un contexte pour leur interprétation.

Dans certains cas, il sera nécessaire de distinguer les URL d'autres structures de données possibles dans une structure syntaxique. Dans ce cas, il est recommandé que les URL soient précédés d'un préfixe constitué des caractères "URL:". Par exemple, ce préfixe peut être utilisé pour distinguer les URL des autres sortes d'URI.

De plus, il y a de nombreuses occasions où les URL sont inclus dans d'autres sortes de textes ; on peut citer par exemple la messagerie électronique, les messages de nouvelles USENET, ou imprimés sur le papier. Dans de tels cas, il est utile d'avoir une enveloppe syntaxique séparée qui délimite l'URL et le sépare du reste du texte, et en particulier des marques de ponctuation qui pourraient être confondues avec le corps de l'URL. A cette fin, il est recommandé d'utiliser les signes inférieur et supérieur à (" $<$ " et " $>$ "), avec le préfixe "URL:", pour délimiter les frontières de l'URL. Cette enveloppe ne fait pas partie de l'URL et ne devrait pas être utilisée dans des contextes dans lesquels des délimiteurs sont déjà spécifiés.

Dans le cas où un identifiant de fragment/ancrage est associé à un URL (suivant un "#"), l'identifiant devrait aussi être placé entre les guillemets.

Dans certains cas, des espaces blancs supplémentaires (espaces, sauts de lignes, tabulations, etc.) peuvent devoir être ajoutés lorsque de longs URL dépassent la ligne. Les espaces blancs devraient être ignorés lors de l'extraction d'URL. Aucun espace blanc ne devrait être introduit après un caractère tiret ("-"). Certains claviers et imprimantes peuvent (à tort) introduire un tiret à la fin d'une ligne lors de la rupture de ligne, et l'interprète d'un URL qui contient un saut de ligne immédiatement après un tiret devrait ignorer tout espace blanc non codé autour de la rupture de ligne, et devrait savoir que le tiret peut ou non faire en réalité partie de l'URL.

Exemple :

Oui, Jim, je l'ai trouvé à `<URL:ftp://info.cern.ch/pub/www/doc; type=d>` mais tu peux probablement l'avoir à `<URL:ftp://ds.internic.net/rfc>`.

Noter l'avertissement dans `<URL:http://ds.internic.net/instructions/overview.html#WARNING>`.

Références

[1] Anklesaria, F., McCahill, M., Lindner, P., Johnson, D., Torrey, D., and B. Alberti, "The Internet Gopher Protocol (a distributed document search and retrieval protocol)", RFC 1436, University of Minnesota, mars 1993.
`<URL:ftp://ds.internic.net/rfc/rfc1436.txt;type=a>`

[2] Anklesaria, F., Lindner, P., McCahill, M., Torrey, D., Johnson, D., and B. Alberti, "Gopher+: Upward compatible enhancements to the Internet Gopher protocol", University of Minnesota, juillet 1993.
`<URL:ftp://boombox.micro.umn.edu/pub/gopher/gopher_protocol/Gopher+/Gopher+.txt>`

[3] Berners-Lee, T., "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630, CERN, juin 1994.
`<URL:ftp://ds.internic.net/rfc/rfc1630.txt>`

[4] Berners-Lee, T., "Hypertext Transfer Protocol (HTTP)", CERN, novembre 1993.
`<URL:ftp://info.cern.ch/pub/www/doc/http-spec.txt.Z>`

- [5] Braden, R., Editor, "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, IETF, octobre 1989. <URL:ftp://ds.internic.net/rfc/rfc1123.txt>
- [6] Crocker, D. "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, avril 1982. <URL:ftp://ds.internic.net/rfc/rfc822.txt>
- [7] Davis, F., Kahle, B., Morris, H., Salem, J., Shen, T., Wang, R., Sui, J., and M. Grinbaum, "WAIS Interface Protocol Prototype Functional Specification", (v1.5), Thinking Machines Corporation, avril 1990. <URL:ftp://quake.think.com/pub/wais/doc/protspec.txt>
- [8] Horton, M. and R. Adams, "Standard For Interchange of USENET Messages", RFC 1036, AT&T Bell Laboratories, Center for Seismic Studies, December 1987. <URL:ftp://ds.internic.net/rfc/rfc1036.txt>
- [9] Huitema, C., "Naming: Strategies and Techniques", Computer Networks and ISDN Systems 23 (1991) 107-110.
- [10] Kahle, B., "Document Identifiers, or International Standard Book Numbers for the Electronic Age", 1991. <URL:ftp://quake.think.com/pub/wais/doc/doc-ids.txt>
- [11] Kantor, B. and P. Lapsley, "Network News Transfer Protocol: A Proposed Standard for the Stream-Based Transmission of News", RFC 977, UC San Diego & UC Berkeley, février 1986. <URL:ftp://ds.internic.net/rfc/rfc977.txt>
- [12] Kunze, J., "Functional Requirements for Internet Resource Locators", RFC 1736, février 1995. <URL:ftp://ds.internic.net/internet-drafts /draft-ietf-uri-irl-fun-req-02.txt>
- [13] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, USC/Information Sciences Institute, novembre 1987. <URL:ftp://ds.internic.net/rfc/rfc1034.txt>
- [14] Neuman, B., and S. Augart, "The Prospero Protocol", USC/Information Sciences Institute, juin 1993. <URL:ftp://prospero.isi.edu/pub/prospero/doc /prospero-protocol.PS.Z>
- [15] Postel, J. and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, USC/Information Sciences Institute, octobre 1985. <URL:ftp://ds.internic.net/rfc/rfc959.txt>
- [16] Sollins, K. and L. Masinter, "Functional Requirements for Uniform Resource Names", RFC 1737, MIT/LCS, Xerox Corporation, décembre 1994. <URL:ftp://ds.internic.net/rfc/rfc1737.txt>
- [17] St. Pierre, M, Fullton, J., Gamiel, K., Goldman, J., Kahle, B., Kunze, J., Morris, H., and F. Schiettecatte, "WAIS over Z39.50-1988", RFC 1625, WAIS, Inc., CNIDR, Thinking Machines Corp., UC Berkeley, FS Consulting, juin 1994. <URL:ftp://ds.internic.net/rfc/rfc1625.txt>
- [18] Yeong, W. "Towards Networked Information Retrieval", Technical report 91-06-25-01, Performance Systems International, Inc.<URL:ftp://uu.psi.com/wp/nir.txt>, juin 1991.
- [19] Yeong, W., "Representing Public Archives in the Directory", Work in Progress, novembre 1991.
- [20] "Coded Caractère Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4-1986.

Adresse des éditeurs

Tim Berners-Lee	Larry Masinter	Mark McCahill
World-Wide Web project	Xerox PARC	Computer and Information Services,
CERN,	3333 Coyote Hill Road	University of Minnesota
1211 Geneva 23,	Palo Alto, CA 94034	Room 152 Shepherd Labs
Switzerland	USA	100 Union Street SE
Phone: +41 (22)767 3755	Phone: (415) 812-4365	Minneapolis, MN 55455
Fax: +41 (22)767 7155	Fax: (415) 812-4333	Phone: (612) 625 1300
EMail: timbl@info.cern.ch	EMail: masinter@parc.xerox.com	EMail: mpm@boombox.micro.umn.edu