Une présentation de Gopher

François Dagorn
Centre de Ressources en Informatique
Université de Rennes I
-
15 juin 1993

1 Introduction

Le réseau Internet est maintenant accessible à la communauté scientifique dans de bonnes conditions, son débit permet l'émergence de nouvelles applications nécessitant de gros transferts d'informations. Parmi celles-ci, la diffusion et la consultation de documents de toutes sortes peuvent enfin devenir réalité. Diverses initiatives permettant de démocratiser la publication et la collecte d'informations ont récemment vu le jour. Baptisées *services d'information répartis*, elles permettent à un chercheur disposant d'un simple PC de publier le résultat de ses travaux, à un collègue géographiquement distant de les consulter aisément, . . .

Gopher, développé par l'université du Minnesota, s'inscrit dans ce cadre. Il a été conçu au départ comme un service d'information de campus mais, très rapidement, il s'est imposé sur tout l'Internet. Gopher considère le réseau Internet comme un livre gigantesque, les pages à lire sont constituées par l'information diffusée par des machines coopérantes (les serveurs). Un lecteur (le client) n'est bien sûr pas obligé de lire tout le livre, il peut se contenter de quelques pages dont il a connaissance. De plus Gopher permet de feuilleter les pages du livre au hasard et d'écorner celles qui présentent un intérêt particulier, de façon à suivre ensuite leur évolution. On présente souvent Gopher comme un outil de *navigation* tant sa capacité pour la *promenade* sur l'Internet est grande.

Gopher est basé sur un modèle client/serveur. Le serveur est responsable de la diffusion de l'information, le client interroge un serveur pour accéder aux données publiées. Il présente les documents à diffuser sous la forme d'une arborescence. Cette technique est généralement bien comprise par tous ceux qui ont une pratique (même légère) des systèmes de gestion de fichiers les plus couramment utilisés (MS-DOS, UNIX . . .). Un serveur Gopher indique la nature de l'information qu'il délivre, c'est au client demandeur de reconnaître ce qu'il a reçu et de réagir en conséquence. De cette façon Gopher permet de diffuser une grande variété de documents incluant des images, des fichiers de sons, des binaires . . .

Conçu dès le départ sur des bases très simples, Gopher est disponible sur toutes les plates-formes standard. On peut ainsi installer un serveur Gopher sur un PC, un Macintosh ou une boîte UNIX. Le logiciel client est également disponible sous différentes formes respectant les principaux standards dans les interfaces utilisateurs (NextStep, X-Window, Windows, Mac, Curses).

Le succès de Gopher va croissant, il y a au moins six cents serveurs disponibles sur l'Internet. De récentes statistiques sur la branche américaine de l'Internet le placent régulièrement aux dix premiers rangs des applications réseau les plus utilisées.

2 Le protocole

Gopher est construit suivant un modèle client/serveur au-dessus de tcp, un serveur écoute les requêtes de clients potentiels sur un port connu (en général le port 70). Un serveur Gopher montre l'information dont il dispose sous une forme hiérarchique, c'est le client qui est responsable de la présentation de celle-ci.

Les échanges entre client et serveur reposent sur un principe très simple. Un client contacte un serveur via une connexion tcp et lui envoie une ligne de texte identifiant les données qu'il désire consulter. Un serveur répond par un bloc de texte terminé par un *point*. Dans le bloc de texte il y a des champs séparés par le caractère *Tab*. À l'issue de sa réponse, le serveur coupe la connexion, le protocole est dit sans états.

Le dialogue entre les protagonistes dépend de la nature de l'information échangée :

- s'il s'agit d'un répertoire, chaque item fait l'objet d'une description (par le serveur), elle peut ensuite être utilisée par le client pour affiner sa demande ;
- dans le cas d'un document le serveur émet les différentes lignes de celui-ci, un *point* suivi des caractères *CR* et *LF* indique la fin.

2.1 Les champs échangés

Quand un serveur délivre le contenu d'un répertoire, il émet une ligne par item. Chaque ligne contient les champs **Type**, **Nom**, **Sélecteur**, **Machine** et **Port**.

2.1.1 Le champ « Type »

Le champ **Type** contient l'identification des documents, il va servir au client pour présenter l'information à l'utilisateur. C'est un des points clefs du protocole Gopher, le serveur qualifie les objets qu'il délivre, le client les interprète. Le procédé utilisé par le serveur pour déterminer la nature d'un document est très simple, c'est en général le suffixe du nom de fichier qui est analysé de la façon suivante :

- *.gif* indique une image ;
- *.src* désigne une source WAIS ;
- ...

Évidemment la méprise est facile, un fichier vide dont le nom est *bidon.gif* sera pris pour une éventuelle image, le nommage des objets est très important pour l'efficacité d'un service Gopher. La figure suivante montre les principaux **Type** émis par un serveur :

champ **Type**	document référencé
0	fichier de texte
1	répertoire
3	message d'erreur
4	fichier *BinHex* pour Macintosh
5	archive binaire Dos
6	fichier *uuencodé*
7	index (WAIS ou Next)
8	session telnet
9	fichier binaire
s	fichier audio
I	image GIF

2.1.2 Le champ « Nom »

Le champ **Nom** est la chaîne de caractères qui est présentée à l'utilisateur pour identifier la ressource. Il s'agit du nom d'un fichier, du nom d'un répertoire ou du contenu du champ **Name** d'un lien Gopher (cf. section 3). Le nommage des objets dans une arborescence Gopher est très important. Il ne faut pas hésiter à utiliser des noms clairs, même s'ils sont un peu longs, car dans la pratique ils sont référencés par l'utilisateur en choisissant une entrée dans un menu.

2.1.3 Le champ « Sélecteur »

Le champ **Sélecteur** est utilisé par un serveur pour indiquer aux clients le nom exact d'un document, ou pour faire référence à des informations diffusées par des services d'informations non Gopher (cf. section 5). Ce champ n'est jamais présenté à l'utilisateur, conjugué avec les champs **Machine** et **Port**, il localise précisément un item.

2.1.4 Le champ « Machine »

Le champ **Machine** contient l'identité de la machine qui héberge le serveur délivrant l'information référencée par le champ **Sélecteur**.

2.1.5 Le champ « Port »

Le champ **Port** indique le numéro de port tcp sur lequel fonctionne le serveur Gopher référencé.

3 L'espace Gopher

Un serveur Gopher accède aux informations qui lui sont accessibles dans une arborescence de données. À cet effet il possède un répertoire d'accueil en dessous duquel il a une vision complète. Tous les objets contenus dans ce répertoire pourront être délivrés aux clients demandeurs et constituent ce que l'on appelle un espace Gopher.

Pour donner accès à des informations détenues par d'autres serveurs, un administrateur local peut créer des liens vers des services distants. Un lien Gopher est un fichier dont le nom commence par un *point* (il n'est pas montré tel quel à l'utilisateur), son contenu réalise un aiguillage. Cette technique à base de liens permet de faire des « sauts de puce » de services en services et donne facilement accès à un immense tissu d'informations : l'espace Gopher mondial (**Gopher space**).

Le contenu d'un lien Gopher reprend les champs définis dans le protocole comme dans l'exemple suivant :

```
Name=Distribution officielle de Gopher
Type=1
Port=70
Path=1/gopher
Host=boombox.micro.umn.edu
```

Le champ **Name** est présenté à l'utilisateur qui ne sait pas que l'item correspondant à « Distribution officielle de Gopher » est distant. C'est cette technique qui est à la base de la répartition du service Gopher.

4 Les principaux clients

Les clients Gopher donnent accès au **Gopher space**, et procurent tous les mêmes fonctionnalités de base en utilisant des interfaces utilisateur différentes. Après avoir sélectionné un document, l'utilisateur peut le visionner, l'imprimer, le copier ou se l'adresser par courrier électronique. Tous les clients autorisent également de « marquer » des pages de l'Internet, des **bookmarks** sont gérées à cet effet. Certaines fonctionnalités n'ont de sens que dans un environnement particulier, elles ne sont pas implantées dans tous les clients (que faire d'un fichier audio sur un terminal asynchrone, . . .).

Divers clients sont disponibles, parmi ceux-ci on trouve :

– **gopher**, le client Unix pour terminaux (en mode *curses*) ;

– **PcGopher**, un client pour PC sous Dos ;

– **xgopher**, un client pour l'environnement X-Window ;

– **GopherBook**, un client pour PC sous Windows ;

– **TurboGopher**, un client pour Macintosh.

4.1 gopher

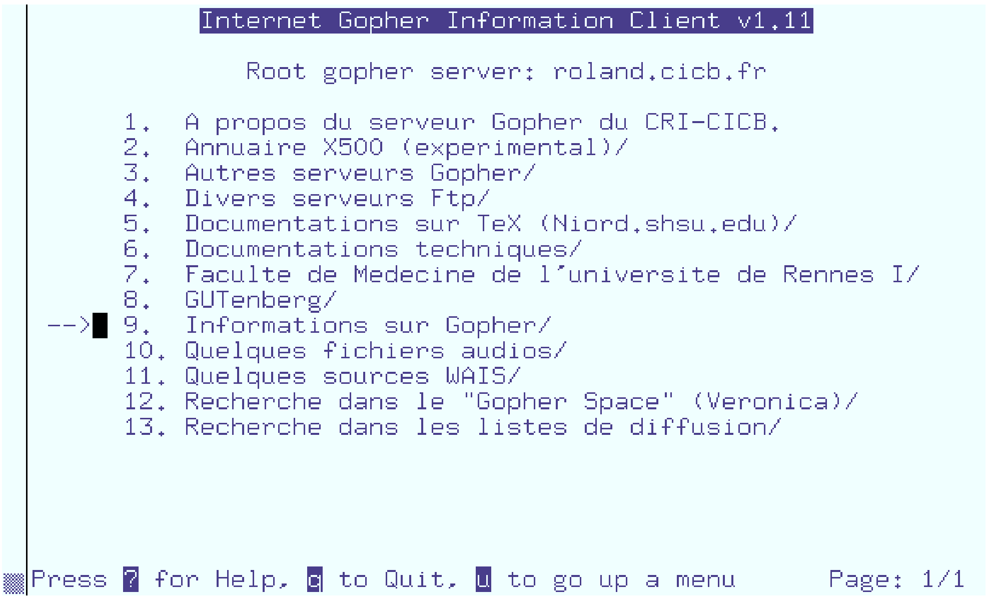
gopher s'utilise très simplement, appelé sans arguments, le client contacte un serveur par défaut indiqué lors de la compilation de l'outil. Pour atteindre un autre serveur, il suffit de citer le nom de la machine qui l'héberge sur la ligne de commande (suivi éventuellement du numéro de port, si celui-ci est différent de 70). Les touches du clavier sont utilisées pour mettre en œuvre les différentes fonctionnalités présentées dans le tableau suivant :

Touches du clavier	fonctionnalités
Retour chariot	visualisation d'un document sélectionné
flèches	positionnement sur l'item précédent ou suivant
<Ctrl-N>, j	positionnement sur l'item suivant
<Ctrl-P>, k	positionnement sur l'item précédent
<numéro>	les items sont numérotés, on peut les sélectionner directement
>, +, espace	page suivante d'un menu comportant un grand nombre d'entrées
<, -, b	page précédente d'un menu comportant un grand nombre d'entrées
m	retour au menu principal, c'est la page d'accueil du premier serveur contacté
u	revenir au menu précédemment affiché
a	marquer une page, ajouter un item dans la liste des **bookmarks**
A	marquer un chapitre, ajouter un menu dans la liste des **bookmarks**
v	consulter la liste des marques
d	supprimer une marque
?	faire apparaître une page d'aide en ligne, ce tableau en est une traduction
q	quitter gopher (avec confirmation)
Q	quitter gopher (sans confirmation)
=	afficher les champs nécessaires à la constitution d'un lien sur l'item sélectionné
/	rechercher une chaîne de caractères dans les items d'un menu

Les marques positionnées sont conservées dans un fichier nommé **.gopherrc**, il est situé dans le répertoire d'accueil de l'utilisateur. **gopher** indique la nature des items qu'il présente en plaçant un repère derrière chacun d'eux, comme indiqué dans le tableau suivant :

repère	type du document
.	fichier texte
	répertoire
<HQX>	binaire pour Macintosh
<BIN>	binaire Unix
<PC BIN>	archive binaire DOS
< >) >	fichier audio
<Picture>	image GIF
<? >	index Wais ou Next

gopher autorise l'écoute des fichiers audios (à condition de disposer du matériel approprié), il convient pour cela de positionner la variable d'environnement **GOPHER_PLAY**. La commande *setenv GOPHER_PLAY "cat >/dev/audio"* est opérante sur les stations Sun. La figure suivante montre une session **gopher** en cours d'utilisation :



4.2 PcGopher

Pc Gopher III est le client pour l'environnement MS-DOS. Il est adapté aux versions de DOS 3.3 ou postérieures, sur des appareils disposant d'au moins 640K de mémoire, et fonctionne sur tous les compatibles sans a priori sur le type d'écran. La souris est souhaitable mais n'est pas obligatoire.

Pc Gopher III nécessite un **CRYNWR packet driver**, et est incompatible avec LAN Workplace ou PC-NFS. Après son initialisation, **Pc Gopher III** affiche sa page de travail, des menus déroulants permettent de personnaliser son fonctionnement. L'entrée *Application* du menu *Configure* est utilisée pour adapter des paramètres tels que *serveur gopher à contacter*, l'entrée *Network* pour le service de noms . . .

Pc Gopher III connaît les types de document suivants :

– **<F>**, fichier de texte visualisé dans une fenêtre particulière. Si la taille du fichier est trop importante, l'utilisateur est invité à donner un nom de fichier de sauvegarde ;

– **<B>**, fichier binaire copié sur l'espace local, l'utilisateur est invité à donner un nom de fichier de sauvegarde (les images GIF sont considérées comme des binaires, elles doivent être visualisées avec un outil approprié) ;

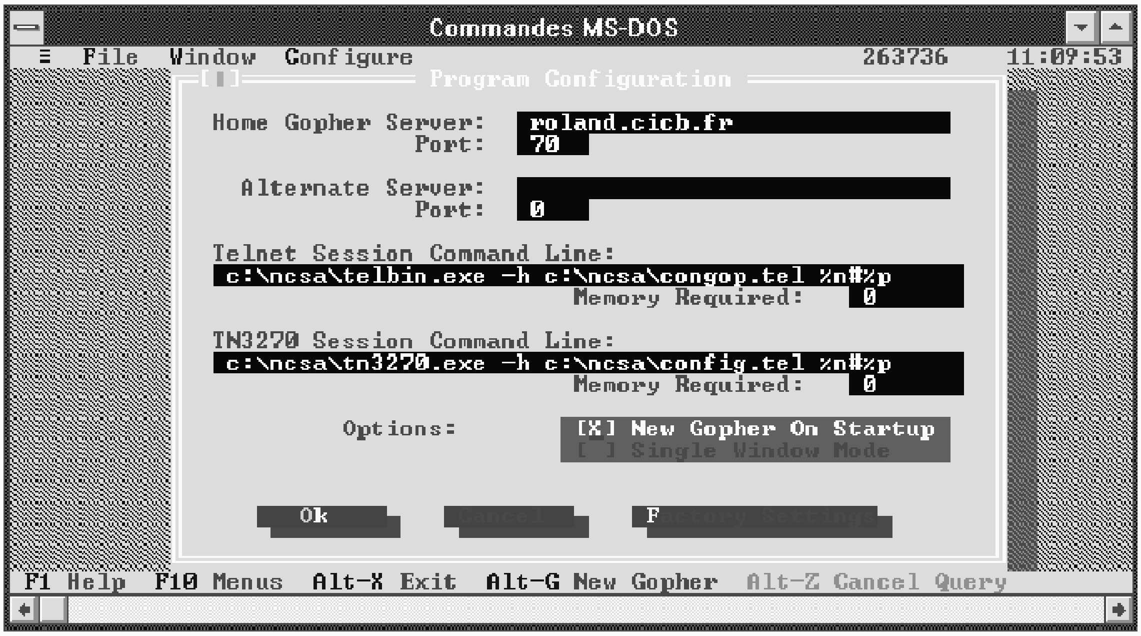
– **<D>**, répertoire ;

– **<S>**, index, l'utilisateur est invité à donner un mot clef de recherche dans une fenêtre de dialogue ;

– **<T>**, session telnet vers un service qui ne *parle* pas le protocole gopher ;

– **<???>**, document dont le type est inconnu, aucune action n'est engagée.

La figure suivante montre une fenêtre de configuration de l'application **Pc Gopher III** :



4.3 xgopher

xgopher est le client pour le monde X-Window, il utilise les widgets Athena ainsi que les bibliothèques X11R5 (à défaut X11R4). La personnalisation de son apparence est réalisée par le schéma classique des fichiers de ressources, l'installation par défaut doit être modifiée pour fixer le nom du serveur à contacter.

xgopher s'utilise très simplement, il autorise de nommer un serveur en argument de la ligne de commande. Les items sont sélectionnés à l'aide de la souris, en cliquant deux fois sur l'objet désiré, ou en utilisant un bouton de commande approprié. Les types de documents connus sont présentés de la manière suivante (les préfixes pouvant être spécifiés dans les ressources) :

- **<espace>**, fichier texte, une fenêtre de texte est ouverte pour le visualiser ;
- **> >**, répertoire ;
- **< idx>**, index WAIS ou Next ;
- **<tel>**, session telnet vers un service non gopher ;
- **<tn3270>**, session d'émulation de terminal 3270 vers un service non gopher ;
- **<img>**, images, l'action effectuée dépend du positionnement de ressources (cf. ci-après) ;
- **<bin>**, binaire, un nom de fichier de sauvegarde est demandé à l'utilisateur ;
- **<snd>**, fichier audio, le traitement effectué dépend du positionnement de ressources (cf. ci-après).

Parmi les ressources de **xgopher**, il convient de porter une attention particulière aux suivantes :

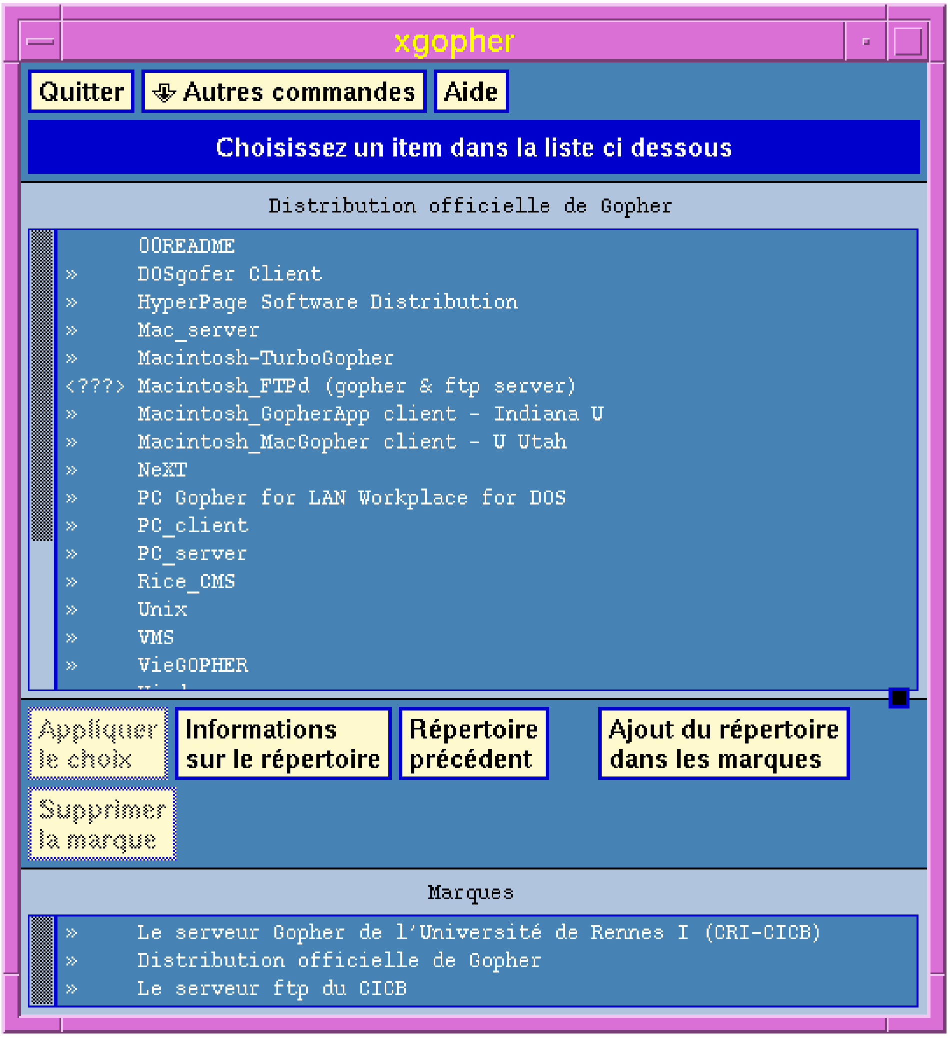
- **rootServer**, le serveur à contacter ;
- **mainTitle**, le titre accompagnant la page de garde du serveur cité par **rootServer** ;
- **telnetCommand**, la commande à utiliser pour les items marqués **<tel>**. Par défaut il s'agit de *xterm -e telnet* ;
- **tn3270Command**, la commande à utiliser pour une session 3270. Par défaut, il s'agit de *xterm -e tn3270* ;

– **imageCommand**, la commande à utiliser pour visualiser les images. La valeur par défaut est *xloadimage -quiet*, mais *xv* peut être plus intéressant. L'appel de **imageCommand** suppose que la ressource **allowImage** soit positionnée à la valeur **true** ;

– **soundCommand**, la commande à utiliser pour soumettre les fichiers audios. Elle est positionnée par défaut à *play*, mais la valeur *cat >/dev/audio* est souvent pertinente. L'appel de **soundCommand** suppose que la ressource **hasSound** soit positionnée à la valeur **true** ;

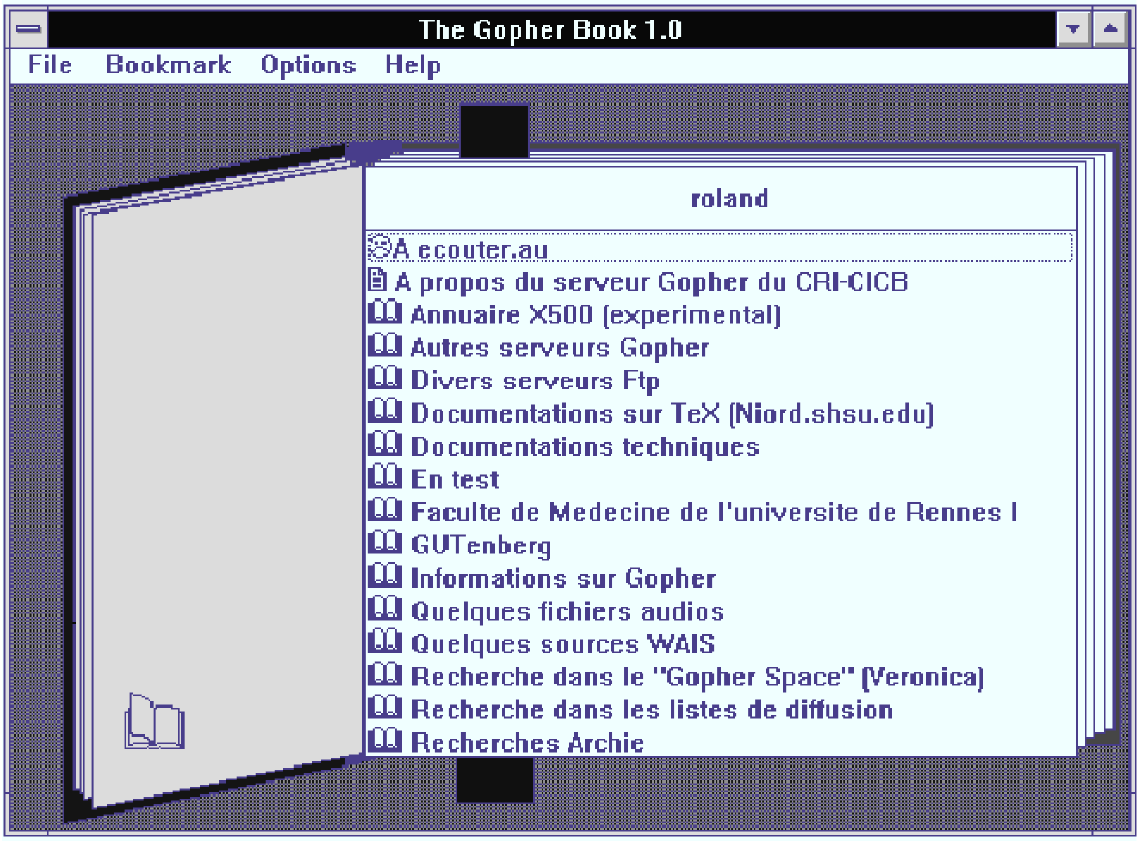
– **bookmarkFile**, le nom du fichier contenant les marques. La valeur par défaut est *.gopherrc* (compatible avec le client **gopher** (curses)).

La figure suivante montre une session **xgopher** en cours d'utilisation :



4.4 Gopherbook

Gopherbook est un client pour l'environnement Windows. Il a été developpé par le CNIDR [1] et necessite un *winsock.dll* [2] **Gopherbook** présente l'espace Gopher sous la forme d'un livre à feuilleter (cf. section 1), la figure suivante montre une session en cours :



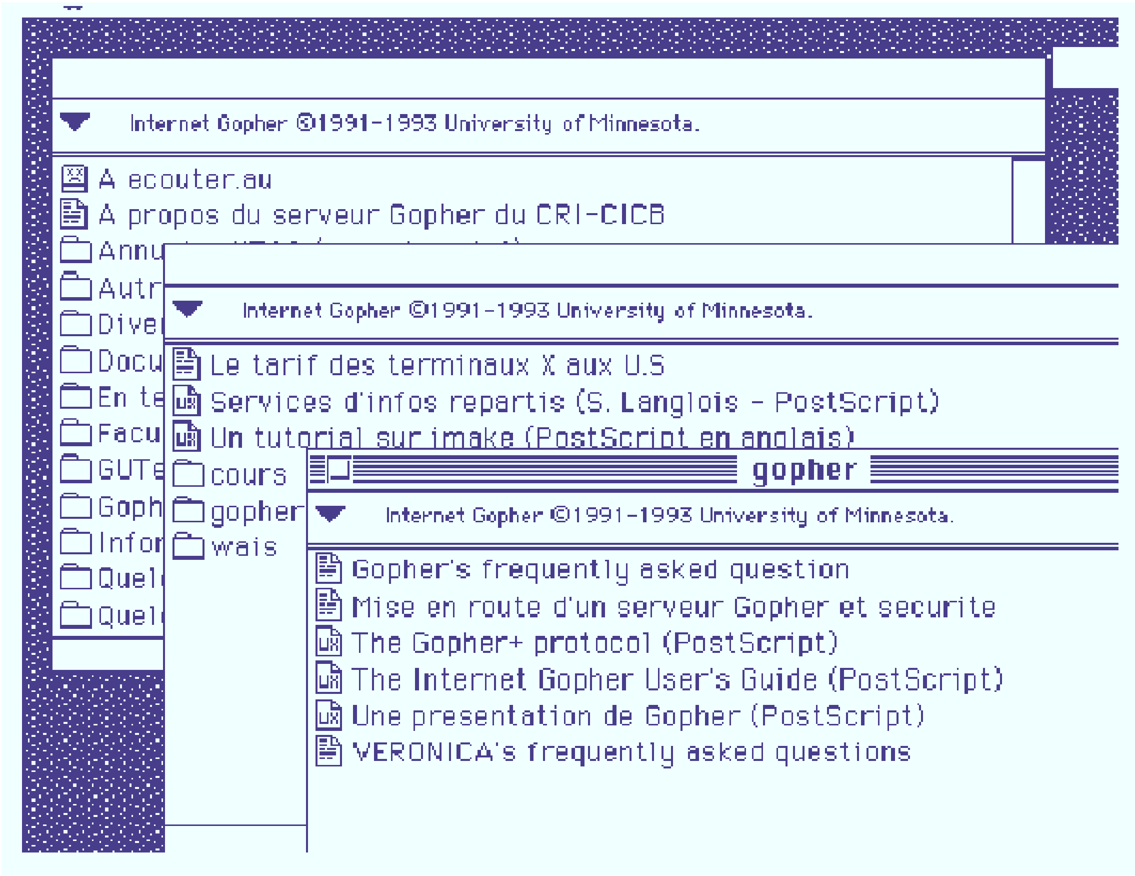
Le menu déroulant **options** permet de configurer le serveur à contacter ainsi que le nom des outils à utiliser pour visualiser les items sélectionnés (une application en *shareware* est fournie pour visualiser les images, . . .).

[1] Clearinghouse for Networked Information Discovery and Retrieval, organisme *sponsorisé* par la National Science Foundation aux États Unis.

[2] interface de programmation *standard* se conformant aux *sockets BSD*, elle permet d'affranchir les applications des piles IP utilisées. Un certain nombre de *winsock.dll* sont disponibles sur le serveur Ftp **sunsite.unc.edu** dans le répertoire */pub/micro/pc-stuff/ms-windows/winsock*.

4.5 TurboGopher

TurboGopher est un client pour l'environnement Macintosh, il fonctionne sous *System 7* et nécessite l'installation de MacTcp. **TurboGopher** est réputé rapide, il permet de visualiser le début d'un document alors que son transfert n'est pas terminé. Certaines fonctionnalités de Gopher+ (cf. section 7) sont accessibles, notamment les vues multiples d'un document. **TurboGopher** est francisé et dispose d'une aide en ligne remarquable. La figure suivante montre une session en cours d'utilisation :



Remarque :

*La description ci-dessus n'est pas une liste exhaustive des clients Gopher. De nombreuses contributions sont disponibles, notamment dans l'environnement Macintosh (***GopherApp*** Indiana University,* **MacGopher** *Utah University).*

5 Les passerelles vers d'autres services

Gopher permet d'interroger des services d'informations qui n'utilisent pas son protocole. On parle de passerelles (**gateways**) qui effectuent le lien entre des protocoles différents. Certaines passerelles sont enlisées dans le code du serveur Gopher, d'autres sont réalisées par des programmes particuliers. Les passerelles de Gopher vers Ftp et WAIS sont partie intégrante du serveur, celles vers X500 ou Archie sont réalisées en dehors de celui-ci. Les passerelles ne sont pas implantées dans les clients, ils peuvent ainsi tous profiter d'un nouveau service rendu par un serveur, de plus ils conservent leur simplicité.

5.1 La passerelle vers Ftp

La passerelle de Gopher vers Ftp est réalisée par le serveur, elle est actionnée à l'émission par un client d'un sélecteur particulier comme dans l'exemple ci-dessous :

ftp:ftp.urec.fr@/pub/

Ce sélecteur est composé de trois parties séparées par les caractères **:** et @ :

- le mot clef **ftp** ;
- le nom du serveur ftp à solliciter (*ftp.urec.fr*);
- le répertoire distant à consulter (*/pub*).

En recevant ce type de sélecteur, un serveur entreprend une connexion Ftp sur la machine référencée. Il effectue le login *anonymous* et indique son identité comme mot de passe (*gopher@* . . .). Les commandes Gopher sont traduites en requêtes Ftp, la sélection d'un nom de fichier génère un *get fichier* . . . De cette façon Gopher procure à Ftp une interface utilisateur conviviale.

5.2 La passerelle vers WAIS

WAIS permet d'indexer divers types de documents, de faciliter les recherches à distance, et d'utiliser des techniques évoluées de visualisation (multimédia . . .). Gopher propose une passerelle vers des bases WAIS, elle est réalisée par le serveur, et est actionnée quand un client émet un sélecteur comme le suivant :

waissrc:/Quelques sources WAIS/bib-rev-mathinfo-mrs.src

Le mot clef *waissrc* active la passerelle, il est suivi d'un nom de fichier qui est un *.src* WAIS (un fichier indiquant la localisation de la base à consulter). En recevant ce type de sélecteur, un serveur

Gopher se comporte en client WAIS et joue un rôle normal en face de son client (il lui présente les résultats d'une recherche sous la forme d'un menu).

Pour procurer une ouverture vers des bases WAIS, un administrateur d'espace Gopher doit simplement installer des *.src* WAIS dans un répertoire. C'est évidemment très facile à réaliser mais parfois limité dans l'usage, Gopher ne permettant pas pour l'instant (cf. Gopher+ en section 7) de visualiser des types de documents très courants (DVI, PostScript).

Gopher autorise également d'interroger des bases WAIS locales créées par l'outil *waisindex* (elles ne sont pas interfacées par un serveur WAIS). Il suffit pour cela de créer un lien Gopher vers le répertoire contenant les fichiers qui constituent une base WAIS comme dans l'exemple suivant :

```
Name=Recherche de mail dans ip.archive
Type=7
Path=7/listes-de-diffusion/ip.archive/.waisindex/index
Host=+
Port=+
```

5.3 La passerelle vers Archie

Archie est un outil adapté à la recherche de documents (programmes sources, documentations ...) diffusés par un très grand nombre de serveurs Ftp. La passerelle de Gopher vers Archie est réalisée par le produit additionnel **go4gw**. Il s'agit d'un ensemble de procédures écrites en PERL, elles réalisent des interfaces vers différents services d'informations (Whois, nntp, Archie ...). Pour configurer la passerelle vers Archie, il faut installer la procédure **g2archie** (et la modifier pour tenir compte de particularités locales). L'ensemble **go4gw** et **g2archie** se comporte comme un serveur Gopher vis-à-vis du client Gopher demandeur (en utilisant en général le port 4320), et comme un client Archie en face du serveur Archie sollicité (en faisant appel au programme client **archie**).

Quand la passerelle est activée, elle propose (par l'intermédiaire de **g2archie**) deux items de recherche indexée :

- une recherche sur un serveur Archie en mode *Exact* ;
- une recherche sur un serveur Archie en mode *Substr*.

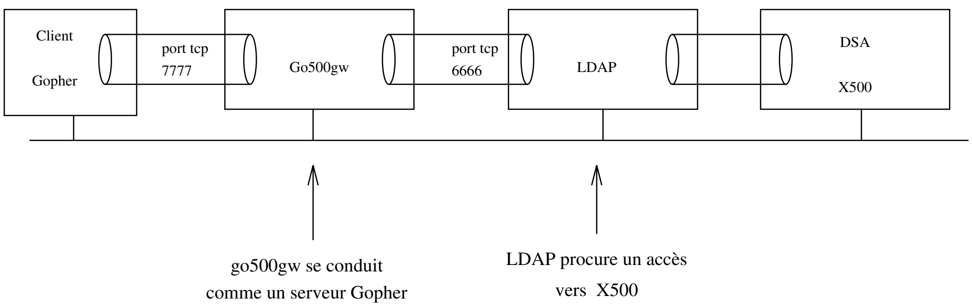
Les réponses du serveur Archie sont présentées sous la forme d'un menu Gopher traditionnel. Les fichiers sélectionnés par la recherche peuvent être transférés en actionnant (de façon transparente) la passerelle Gopher vers Ftp.

5.4 La passerelle vers X500

X500 est une norme définissant l'accès à des services d'annuaires (de personnes, de services . . .). La passerelle de Gopher vers X500 est réalisée en utilisant le programme **go500gw** qui propose une interface au produit **LDAP**. Développé par l'université du Michigan, **LDAP** permet d'interroger (de façon allégée) un annuaire X500.

go500gw émule un serveur Gopher spécialisé (il utilise en général le port tcp 7777), la figure suivante montre les différents protagonistes d'une passerelle Gopher vers X500 :

Architecture d'une passerelle de Gopher vers X500



6 Comment se procurer Gopher

L'équipe de développement de Gopher à l'université du Minnesota diffuse l'ensemble du logiciel développé par l'intermédaire d'un serveur Ftp installé sur la machine **boombox.micro.umn.edu**. Ftp est la manière d'effectuer une première amorce, et dans **/pub/gopher** on trouve des répertoires adaptés aux différents cas :

- **Mac_server**, le logiciel serveur pour Mac Intosh ;
- **Macintosh-TurboGopher**, le client *officiel* pour Mac Intosh ;
- **PC_client**, le client pour PC sous DOS (**PC Gopher III**) ;
- . . .

Lorsque l'utilisateur dispose d'un client Gopher, il peut suivre les évolutions en effectuant une connexion sur **boombox.micro.umn.edu** (port 70).

7 Le nouveau protocole Gopher+

Spécifié au départ pour un service d'information de campus, le protocole Gopher se révèle aujourd'hui insuffisant pour un service d'informations réparti sur l'Internet. Un élément aussi important que la taille d'un document à visualiser n'est pas transmis aux clients potentiels, ils ne peuvent donc pas se protéger contre des temps (et des coûts) de transferts élevés. Gopher+ est la réponse des développeurs de l'université du Minnesota, rédigé pendant l'été 1992, il est maintenant quasiment opérationnel. Il s'agit d'améliorations au *vénérable* protocole d'origine, elles garantissent toutefois la cohabitation entre anciens et nouveaux protagonistes (clients comme serveurs).

Les nouveaux clients et serveurs ajoutent des champs derrière ceux cités en section 2.1, ils seront ignorés par les acteurs qui n'ont pas évolué. Ces nouveaux champs sont utilisés par les clients pour détailler la description d'un document en s'informant sur :

– sa taille en octets ;
– les différents formats dans lesquels il est disponible (texte, PostScript . . .) ;
– les différentes langues dans lesquelles il est rédigé ;
– les coordonnées de la personne qui gère sa diffusion ;
– l'existence d'un résumé le présentant brièvement ;
– . . .

La mise en route de Gopher+ s'opère lentement, le logiciel n'est pas encore complètement stabilisé et est disponible pour l'instant en version Béta (gopher+1.2b4).

8 Le service VERONICA

Un très grand nombre de serveurs coopèrent pour former le **Gopher space** où l'utilisateur peut naviguer à son gré. La découverte d'informations est parfois liée au hasard, Gopher ne prévoit pas de mécanismes pour orienter l'utilisateur dans ses recherches.

VERONICA [3] développé à l'université du Nevada tente d'apporter une aide à la localisation de documents. Il s'agit d'un service d'indexation de tous les titres de tous les menus Gopher déclarés sur la totalité de l'Internet. Les index constitués sont interrogeables par un client Gopher, les résultats des recherches sont présentés sous la forme de menus Gopher. En aidant à localiser des documents, **VERONICA** se positionne vis-à-vis de Gopher comme Archie face aux serveurs Ftp.

[3] Very Easy Rodent-Oriented Net-wide Index to Computerized Archives

Le service **VERONICA** s'appuie sur les éléments suivants :

- un processus de collecte de tous les titres des menus du **Gopher space** et l'indexation de ceux-ci. Lorsqu'un serveur est déclaré (cf. section 10), il est automatiquement sondé, seul les items locaux sont collectés ;
- un outil servant à offrir le service de recherche.

Les deux constituants sont indépendants, le but est de favoriser la mise en service d'accès **VERONICA**, un serveur pouvant simplement hériter de la base d'index constituée par un autre site. De cette manière les sondages de serveurs Gopher sont limités et les réponses aux recherches homogènes.

Il y a deux façons d'interroger un serveur **VERONICA**, en donnant un seul mot clef, ou en donnant plusieurs mot clefs séparés par des opérateurs booléens (and, or). Les résultats sont présentés sous la forme de menus, chaque item est un titre d'un menu du **Gopher space** qui contient le mot clef recherché.

Le nombre de serveurs **VERONICA** est pour l'instant limité, les quatre liens suivants donnent accès au service :

```
Name =  'VERONICA (Universite du Nevada)'
Type =  1
Host =  'veronica.scs.unr.edu'
Port =  70
Path =  '1/veronica'

Name =  'VERONICA (CNIDR)'
Type =  1
Host =  'wisteria.cnidr.org'
Port =  70
Path =

Name =  'VERONICA (NYSERNet)'
Type =  1
Host =  'nysernet.org'
Port =  70
Path =  1/Search the Internet

Name =  'VERONICA (Universite de PISE)'
Type =  1
Host =  'serra.unipi.it'
Port =  70
Path =
```

9 Gérer un serveur

La présente section est un bref résumé d'une expérience de mise en route d'un serveur Gopher sur une machine Unix.

9.1 Le démarrage

La compilation de Gopher pose peu de problèmes (**gopher1.12**), deux fichiers doivent être configurés pour tenir compte de particularités locales (conf.h et Makefile.config). Pour disposer de la passerelle vers WAIS, il convient de positionner deux liens vers une partie des sources de WAIS (liens vers les répertoires ir et ui de wais-8-b5.1).

Le point le plus délicat concerne les éventuels trous dans la sécurité que peut introduire un serveur Gopher. Le processus serveur **gopherd** possède un répertoire d'accueil dans lequel sont stockées les informations à diffuser. Une option de démarrage de **gopherd** permet de limiter son champ d'action à ce répertoire. Ce procédé est en principe *secure*, mais peut manquer de souplesse pour la gestion de l'espace Gopher local (impossibilité de créer des liens symboliques). La solution souple (et probablement peu sécurisée) laisse au serveur Gopher l'opportunité de résoudre les liens symboliques (argument **-c** *Do not chroot before processing connections*), il est alors impératif d'executer **gopherd** sous l'identité d'un usager aux droits limités (*nobody* par exemple).

La solution très sécurisée déporte le *home dir* de **gopherd** via NFS (en *Read-only* bien sûr, tout en maintenant le *chroot*). Si Gopher à besoin de servir des informations qui ne se situent pas sous son *home dir*, il convient alors d'effectuer les montages NFS adéquats (toujours en *Read-only*).

9.2 Partager un espace avec un serveur Ftp

Les sites diffusant de l'information via Ftp doivent continuer à le faire, toutefois il peut être intéressant de montrer via Gopher une hiérarchie distribuée par Ftp. La valeur ajoutée de Gopher réside alors dans l'interface utilisateur qu'il propose. Pour faciliter la cohabitation d'une diffusion par Ftp et Gopher, un administrateur d'espace documentaire peut utiliser des répertoires spéciaux appelés **.cap**.

Lorsque Gopher consulte un répertoire, il teste l'existence dans celui-ci d'un répertoire emboîté de nom **.cap**. Si **.cap** existe, Gopher le sonde pour chercher d'éventuels fichiers dont le nom est identique à ceux figurant dans le répertoire père. Il s'agit d'éventuels fichiers ayant la structure d'un lien Gopher, et autorisant des renommages pour l'usage de Gopher. L'exemple suivant est tiré de l'espace partagé par les serveurs Gopher et Ftp du CRI de l'université de Rennes I :

– ci-dessous, le contenu du répertoire /pub/doc/gopher vu par le serveur ftp (ftp.univ-rennes1.fr) :

```
Gopher+.ps
gopher-user-guide.ps
Gopher.ps
VERONICA.FAQ
```

Il n'y a pas trop de fioritures dans les noms des fichiers, l'utilisateur est probablement obligé de les nommer explicitement ;

– un client Gopher montre le contenu suivant :

```
1.  The Gopher+ protocol (PostScript)
2.  The Internet Gopher User's Guide (PostScript)
3.  Une presentation de Gopher (PostScript)
4.  VERONICA's frequently asked questions.
```

– le contenu du répertoire /pub/doc/gopher/.cap (vu par le serveur Gopher) est apparemment identique à celui de /pub/doc/gopher mais les fichiers sont des liens Gopher, comme Gopher.ps qui contient :

```
Name=Une presentation de Gopher (PostScript)
Type=9
```

Le champ **Name** est utilisé pour renseigner au mieux le client Gopher, l'utilisateur n'a pas à nommer le document, il fait un simple choix dans un menu. Le champ **Type** est ici utilisé pour forcer la nature du document (il n'est probablement pas utile de visionner un document PostScript comme un fichier de texte).

10 Trucs utiles

Pour déclarer un serveur Gopher en France il convient d'expédier un courrier électronique à **gopher@jussieu.fr**.

La liste de distribution **gopher-francophone@jussieu.fr** permet aux utilisateurs et administrateurs d'espace Gopher d'aborder tout problème relatif à Gopher. Pour être enregistré dans la liste, adresser un courrier à **gopher-francophone-request@jussieu.fr**.

Le forum de News **comp.infosystems.gopher** est également disponible. Il est utilisé par les développeurs de l'université du Minnesota, ainsi que par des contributeurs, pour dialoguer avec les utilisateurs.

11 Bibliographie

– « *The whole Internet* »
Ed Krol

O'Reilly and Associates
Septembre 1992

– Tous les documents suivants sont disponibles sur le serveur Ftp du CRI de l'université de Rennes I (ftp.univ-rennes1.fr) dans le répertoire **/pub/doc**. Ils sont également accessibles par Gopher (**roland.univ-rennes1.fr**) dans le répertoire "Documentations techniques".

– « *Internet Gopher User's Guide* »
Paul Lindner
University of Minnesota

– « *Gopher+ : proposed enhancements to the internet Gopher protocol* »
Gopher development team
University of Minnesota

– « *Services d'informations répartis : un panorama sélectif* »
Sylvain Langlois - Electricité de France
17 février 1993

– « *Wide Area Information Servers* »
Luc Ottavj - INRIA
29 mars 1993

– « *Le projet WAIS* »
Pierre Barthélémy
Laboratoire de Mathématiques discrètes - U.P.R. 9016 CNRS
et
Robert Rolland
Centre International de Rencontres Mathématiques - SMF / U.M.S. 822 CNRS

– « *Aide en ligne de TurboGopher* »
Gopher development team
traduction de J.P. Kuypers