Gopher (Gopher Protocol)

DESCRIPTION
The Internet Gopher, or simply Gopher, is a distributed document delivery service. It allows users to explore, search and retrieve information residing on different locations in a seamless fashion. When browsing it, the information appears to the user as a series of nested menus. This kind of menu structure resembles the organization of a directory with many subdirectories and files. The subdirectories and the files may be located either on the local server site or on remote sites served by other Gopher servers. From the user point of view, all information items presented on the menus appear to come from the same place.

The information can be a text or binary file, directory information (loosely called phone book), image or sound. In addition, Gopher offers gateways to other information systems (World-Wide Web, WAIS, archie, WHOIS) and network services (Telnet, FTP). Gopher is often a more convenient way to navigate in a FTP directory and to download files.

A Gopher server holds the information and handles the users' queries. In addition, links to other Gopher servers create a network wide cooperation to form the global Gopher web (Gopherspace).

History
The original Gopher system was released in late spring of 1991 by Mark McCahill, Farhad Anklesaria, Paul Lindner, Dan Torrey, and Bob Alberti of the University of Minnesota. Its central goals were:

A file-like hierarchical arrangement that would be familiar to users

A simple syntax

A system that can be created quickly and inexpensively

Extending the file system metaphor to include things like searches

The source of the name "Gopher" is claimed to be three-fold:

Users instruct it to "go for" information

It does so through a web of menu items analogous to gopher holes

The sports team of the University of Minnesota is the Golden Gophers

Gopher combines document hierarchies with collections of services, including WAIS, the Archie and Veronica search engines, and gateways to other information systems such as ftp and Usenet.

The general interest in Campus-Wide Information Systems (CWISs) in higher education at the time, and the ease with which a Gopher server could be set up to create an instant CWIS with links to other sites' online directories and resources were the factors contributing to Gopher's rapid adoption. By 1992, the standard method of locating someone's e-mail address was to find their organization's CSO nameserver entry in Gopher, and query the nameserver.

The exponential scaling of utility in social networked systems seen in Gopher, and then the web, is a common feature of networked hypermedia systems with distributed authoring. In 1993¨C1994, Web pages commonly contained large numbers of links to Gopher-delivered resources, as the Web continued Gopher's embrace and extend tradition of providing gateways to other services.

The World Wide Web was in its infancy in 1991, and Gopher services quickly became established. However, by the late 1990s, Gopher had almost disappeared. Insofar as information management is concerned, the progress from gopher to the web as a standard can be seen simply as a natural progression from text-based to graphical interfaces.

As of 2006, there are still a few Gopher servers present on the net, in organizations such as the Smithsonian Institution and the US government; a few are also being maintained by enthusiasts of the protocol, where almost all growth is occurring.

Some have suggested that the bandwidth-sparing simple interface of Gopher would be a good match for mobile phones and Personal digital assistants (PDAs), but so far, the market prefers Wireless Markup Language (WML)/Wireless Application Protocol (WAP), DoCoMo i-mode, XHTML Basic or other adaptations of HTML and XML. The PyGopherd server, however, provides a built-in WML front-end to Gopher sites served with it.

Internet Gopher Model

In essence, the Gopher protocol consists of a client connecting to a server and sending the server a selector (a line of text, which may be empty) via a TCP connection. The server responds with a block of text terminated with a period on a line by itself, and closes the connection. No state is retained by the server between transactions with a client. The simple nature of the protocol stems from the need to implement servers and clients for the slow, smaller desktop computers (1 MB Macs and DOS machines), quickly, and efficiently.

Below is a simple example of a client/server interaction; more complex interactions are dealt with later. Assume that a "well-known" Gopher server listens at a well known port for the campus (much like a domain-name server). The only configuration information the client software retains is this server's name and port number (in this example that machine is rawBits.micro.umn.edu and the port 70). In the example below the F character denotes the TAB character.

Client: {Opens connection to rawBits.micro.umn.edu at port 70}

Server: {Accepts connection but says nothing}

Client: {Sends an empty line: Meaning "list what you have"}

Server: {Sends a series of lines, each ending with CR LF}

0About internet GopherFStuff:About usFrawBits.micro.umn.eduF70

1Around University of MinnesotaFZ,5692,AUMFunderdog.micro.umn.eduF70

1Microcomputer News & PricesFPrices/Fpserver.bookstore.umn.eduF70

1Courses, Schedules, CalendarsFFevents.ais.umn.eduF9120

1Student-Staff DirectoriesFFuinfo.ais.umn.eduF70

1Departmental PublicationsFStuff:DP:FrawBits.micro.umn.eduF70

{.....etc.....}

. {Period on a line by itself}

{Server closes connection}

The first character on each line tells whether the line describes a document, directory, or search service (characters '0', '1', '7'; there are a handful more of these characters described later). The succeeding characters up to the tab form a user display string to be shown to the user for use in selecting this document (or directory) for retrieval. The first character of the line is really defining the type of item described on this line. In nearly every case, the Gopher client software will give the users some sort of idea about what type of item this is (by displaying an icon, a short text tag, or the like).

The characters following the tab, up to the next tab form a selector string that the client software must send to the server to retrieve the document (or directory listing). The selector string should mean nothing to the client software; it should never be modified by the client. In practice, the selector string is often a pathname or other file selector used by the server to locate the item desired. The next two tab delimited fields denote the domain-name of the host that has this document (or directory), and the port at which to connect. If there are yet other tab delimited fields, the basic Gopher client should ignore them. A CR LF denotes the end of the item.

In the example, line 1 describes a document the user will see as "About internet Gopher". To retrieve this document, the client software must send the retrieval string: "Stuff:About us" to rawBits.micro.umn.edu at port 70. If the client does this, the server will respond with the contents of the document, terminated by a period on a line by itself. A client might present the user with a view of the world something like the following list of items:

About Internet Gopher

Around the University of Minnesota...

Microcomputer News & Prices...

Courses, Schedules, Calendars...

Student-Staff Directories...

Departmental Publications...

In this case, directories are displayed with an ellipsis and files are displayed without any. However, depending on the platform the client is written for and the author's taste, item types could be denoted by other text tags or by icons. For example, the UNIX curses-based client displays directories with a slash (/) following the name; Macintosh clients display directories alongside an icon of a folder.

The user does not know or care that the items up for selection may reside on many different machines anywhere on the Internet.

Suppose the user selects the line "Microcomputer News & Prices...". This appears to be a directory, and so the user expects to see contents of the directory upon request that it be fetched. The following lines illustrate the ensuing client-server interaction:

Client: (Connects to pserver.bookstore.umn.edu at port 70)

Server: (Accepts connection but says nothing)

Client: Prices/ (Sends the magic string terminated by CRLF)

Server: (Sends a series of lines, each ending with CR LF)

0About PricesFPrices/AboutusFpserver.bookstore.umn.eduF70

0Macintosh PricesFPrices/MacFpserver.bookstore.umn.eduF70

0IBM PricesFPrices/IckFpserver.bookstore.umn.eduF70

0Printer & Peripheral PricesFPrices/PPPFpserver.bookstore.umn.eduF70

(.....etc.....)

. (Period on a line by itself)

(Server closes connection)

More details

Locating services

Documents (or other services that may be viewed ultimately as documents, such as a student-staff phonebook) are linked to the machine they are on by the trio of selector string, machine domain-name, and IP port. It is assumed that there will be one well-known top-level or root server for an institution or campus. The information on this server may be duplicated by one or more other servers to avoid a single point of failure and to spread the load over several servers. Departments that wish to put up their own departmental servers need to register the machine name and port with the administrators of the top-level Gopher server, much the same way as they register a machine name with the campus domain-name server. An entry which points to the departmental server will then be made at the top level server. This ensures that users will be able to navigate their way down what amounts to a virtual hierarchical file system with a well known root to any campus server if they desire.

Note that there is no requirement that a department register secondary servers with the central top-level server; they may just place a link to the secondary servers in their own primary servers. They may indeed place links to any servers they desire in their own server, thus creating a customized view of the Gopher information universe; links can of course point back at the top-level server. The virtual (networked) file system is therefore an arbitrary graph structure and not necessarily a rooted tree. The top-level node is merely one convenient, well-known point of entry. A set of Gopher servers linked in this manner may function as a campus-wide information system.

Servers may of course point links at other than secondary servers. Indeed servers may point at other servers offering useful services anywhere on the internet. Viewed in this manner, Gopher can be seen as an Internet-wide information system.

Server portability and naming

It is recommended that all registered servers have alias names (domain name system CNAME) that are used by Gopher clients to locate them. Links to these servers should use these alias names rather than the primary names. If information needs to be moved from one machine to another, a simple change of domain name system alias (CNAME) allows this to occur without any reconfiguration of clients in the field. In short, the domain name system may be used to re-map a server to a new address. There is nothing to prevent secondary servers or services from running on otherwise named servers or ports other than 70, however these should be reachable via a primary server.

Contacting server administrators

It is recommended that every server administrator have a document called something like: "About Bogus University's Gopher server" as the first item in their server's top level directory. In this document should be a short description of what the server holds, as well as name, address, phone, and an e-mail address of the person who administers the server. This provides a way for users to get word to the administrator of a server that has inaccurate information or is not running correctly. It is also recommended that administrators place the date of last update in files for which such information matters to the users.

Modular addition of services

The first character of each line in a server-supplied directory listing indicates whether the item is a file, a directory, or a search. This is the base set of item types in the Gopher protocol. It is desirable for clients to be able to use different services and speak different protocols (simple ones such as finger; others such as CSO phonebook service, or Telnet, or X.500 directory service) as needs dictate. CSO phonebook service is a client/server phonebook system typically used at Universities to publish names, e-mail addresses, and so on. The CSO phonebook software was developed at the University of Illinois and is also sometimes referred to as ph or qi.

On the other hand, subsets of other document retrieval schemes may be mapped onto the Gopher protocol by means of "gateway-servers". Examples of such servers include Gopher-to-FTP gateways, Gopher-to-archie gateways, Gopher-to-WAIS gateways, etc. There are a number of advantages of such mechanisms. First, a relatively powerful server machine inherits both the intelligence and work, rather than the more modest, inexpensive desktop system that typically runs client software or basic server software. Equally important, clients do not have to be modified to take advantage of a new resource.

Building clients

A client simply sends the retrieval string to a server if it wants to retrieve a document or view the contents of a directory. Of course, each host may have pointers to other hosts, resulting in a "graph" (not necessarily a rooted tree) of hosts. The client software may save (or rather "stack") the locations that it has visited in search of a document. The user could therefore back out of the current location by unwinding the stack. Alternatively, a client with multiple-window capability might just be able to display more than one directory or document at the same time.

A smart client could cache the contents of visited directories (rather than just the directory's item descriptor), thus avoiding network transactions if the information has been previously retrieved.

Building ordinary internet Gopher servers

The retrieval string sent to the server might be a path to a file or directory. It might be the name of a script, an application or even a query that generates the document or directory returned. The basic server uses the string it gets up to but not including a CR-LF or a TAB, whichever comes first.

Special purpose servers

There are two special server types (beyond the normal Gopher server) also discussed below:

A server directory listing can point at a CSO nameserver to allow a campus student-staff phonebook lookup service. This may show up on the user's list of choices, perhaps preceded by the icon of a phone-book.

A server can also point at a "search server". Such servers may implement campus network (or subnet) wide searching capability. The most common search servers maintain full-text indexes on the contents of text documents held by some subset of Gopher servers.

Item type characters

The client software decides what items are available by looking at the first character of each line in a directory listing. Augmenting this list can extend the protocol. A list of defined item-type characters follows:

0 Item is a file
1 Item is a directory
2 Item is a CSO phone-book server
3 Error
4 Item is a BinHexed Macintosh file
5 Item is DOS binary archive of some sort. Client must read until the TCP connection closes. Beware.
6 Item is a UNIX uuencoded file.
7 Item is an Index-Search server.
8 Item points to a text-based telnet session.
9 Item is a binary file! Client must read until the TCP connection closes. Beware.
+ Item is a redundant server
T Item points to a text-based tn3270 session.
g Item is a GIF format graphics file.
I Item is some kind of image file. Client decides how to display.

User display strings and server selector strings

User display strings are intended to be displayed on a line on a typical screen for a user's viewing pleasure. While many screens can accommodate 80 character lines, some space is needed to display a tag of some sort to tell the user what sort of item this is. Because of this, the user display string should be kept under 70 characters in length. Clients may truncate to a length convenient to them.