Tim Berners-Lee
CERN, 1211 Geneva 23, Switzerland
February 1992

What W3 needs from WAIS and x.500

Abstract

There has been much discussion about the relative roles of the WAIS protocol and the x.500 distributed naming scheme in the information universe. This paper notes a few requirements on such protocols for a global hypertext/index documentation system.

Status

Informal, intended for discussion before, during and after the March 1992 WAIS/x.500 IETF "BOF" meeting. Distribution unlimited.

Background

The World-Wide Web (W^3) initiative is an attempt to make all online information easily accessible as a web of hypertext, indexes, and other documents. It uses existing and new protocols to achieve this. In order to use the WAIS protocol, a few changes should be made to the WAIS protocol which are outlined here.

The need for WAIS protocol

A fast, stateless, search and retrieve protocol such as the WAIS protocol is requires for remote document and index search. Currently, W3 uses a different internet protocol (HTTP) as an alternative to WAIS but merging of the functionality would is desirable. In a hypertext world, search or retrieval is typically initiated by a user clicking with a mouse on a sensitive area (text, icon etc) or typing a question into a query panel. The results are expected to appear ideally within 0.1 – 5 seconds. A reader advances from document to document by series of such operations. Sometimes there is a strong correlation between the physical locations of successive documents, and sometimes there is none.

Use of Universal Document Identifiers

The need for universal document identifiers has been identified by many people. The global hypertext information space requires such objects. A search protocol should return its results in terms of such identifiers. the current WAIS implementations return document identifiers which ar not open enough to point to non-WAIS objects or objects registered in a directory service. A separate paper discusses this in more detail and makes a proposal [1]. This is the most important requirement.

Limited round trips

An example search may start by following a link from an overview page at CERN to a WAIS "archie" server in the USA. The search may lead to a file on an anonymous FTP server in Australia. Three simple user actions have accessed three servers on three continents.

The protocol must therefore have a very light session if any with the server. The number of round-trip delays between the search or retrieve operation being started to the first data being present at the client must be minimized. This suggests a protocol in which the entire request including any initial session-oriented information currently sent with a session initialization function be sent in one APDU.   (Compare “fast connect” enhancements to connection-oriented protocols) The data which is sent in response to that request should, in the case of a small document or search result, be all that is needed to complete the presentation to the user, and, in the case of a long document, sufficient to start that display.

Throughput

After the initial data has come in, the rest of a long document must be retrieved as quickly as possible. The requirement now changes to a throughput requirement characteristic of a stream protocol. The use of TCP/IP as a byte stream has proved very efficient in this regard, as the delay associated with the buffering of complete APDUs at application level is removed. We mention this only because the requirement above for small documents might have suggested an RPC protocol such as that underlying OSI/DCE, whereas such protocols might prove inefficient for long documents. In practice, while the size of documents in a given database is often distributed over a certain range, there are very large variations between databases, and optimization for only the short or the long case is not satisfactory.

Request APDU

The size of the request APDU is in general small, and enlarging it somewhat would not decrease response time significantly until it is comparable in size with typical small documents. Information which our experience suggests would be useful in a request APDU includes

- Any details about maximum buffer size, when underlying layers are buffer-oriented. This is not strictly an S&R level function, but if it has to be at this level, then it should be sent with the request as discussed above.

- Document id of requested document or index to be searched.

- Search criteria. These are not discussed here but it is suggested that an open attitude here is important to allow for evolution in methods of representing search criteria.

- Client profile for logging purposes: optional client user name, host domain name to allow statistics. (We find that many clients are not DNS registered, so an indication of a name and location in these cases would be helpful).

- Client software type and version, to allow obscure client software related problems to be identified, to allow the propagation of client software versions to be monitored.

- Client authorization strings.   These may be sent with the request by the client software. The z39.50 method of authority checking involves the server calling back the client, which adds round trip delays to the operation. The first time a request fails as being unauthorized, the client software would request authorization strings (keys, name/password etc) from the user. These would be sent next time requests were made for documents specified in the same domain.

- Acceptable data representations. One side must decide in which format the data should be transmitted. The quickest way to do this is for the client to send over a complete list of acceptable formats (or predefined sets of formats) so that the server can make the choice with no further communication. This format negotiation was foreseen as an important part of the W3 architecture. Whilst it has not (Feb 92) yet been implemented its lack is most noticeable.

The need for a directory service

Current document identifiers are often stored for the duration of a user/client session, or only for the duration of a client/server session (ISO S&R protocol). In a hypertext universe, links may be made from one document to any document (or search result, in W3). The links are lasting, and so require lasting names for their destinations. Already the WAIS world has found a need for lasting names for index servers to allow servers to be moved. Lasting names can only be provided by reference to a name registration directory service such as x.500. They are independent of the physical location of the information. Let us assume that x.500 be used for this purpose.

We envisage x.500 names as being used to refer to all documents which have been registered in x.500. Other, less permanent, documents may have no x.500 registration, in which case they may be referred to by their physical addresses. The directory look-up we therefore see happening potentially every time a document is fetched.

- Ubiquity: where x.500 is not available, gateways using another protocol must be available to allow x.500 name translation.

- Speed: Lookup must add no appreciable delay to the process, or it will not in practice be used. This suggests extensive caching.

- An x.500 entity must exist representing a document. Its attributes must include one or more universal document identifiers for the physical addresses.

- A universal document identifier must be able to hold an x.500 address.

Notice that x.500 is being used here as a method for registering documents and being able to move them and still find them later. It is not being used as a resource discovery tool itself.

There is no reason why a hypertext S&R server might not be set up as a gateway into the x.500 world for those who specifically want to search the x.500 name space.

References

[1] Berners-Lee, T, et al., "Universal Document Identifiers for the Network", CERN, February 1993, UDI=file://info.cern.ch./pub/www/doc/udi1.ps

and other references in that document.