The TEK Search Engine

Libby Levison, Bill Thies, Saman Amarasinghe
MIT Laboratory for Computer Science
libby@mit.edu, thies@mit.edu, saman@lcs.mit.edu

Abstract

The Internet has the potential to deliver information to areas of the world that have no other information resources. High telephone and ISP fees – in combination with low bandwidth connections – make it unaffordable for many people to browse the Web online. We are developing the TEK system to enable users to search the Web using only email. TEK stands for "Time Equals Knowledge," since the user exchanges time (waiting for email) for knowledge (contained in the email). The system contains three components: 1) the client, which presents a graphical interface to the end user, 2) the server, which performs the searches from MIT, and 3) a reliable email-based communication protocol between the client and the server. The TEK search engine differs from others in that it is designed to return low-bandwidth results, which are achieved by special filtering, analysis, and compression on the server side. We believe that TEK will bring Web resources to people who otherwise would not be able to afford them.

1. Introduction

In many places in the world, there are no books, there are no libraries, and there is limited access to information. In places that have both computers and functioning phone lines, the Internet has the potential to provide access to a large amount of information electronically. However, there are obstacles. Bandwidth is so narrow that it can take the user a long time to find what she is looking for when browsing the Web, since she has to wait for each page to be loaded. Moreover, time spent online translates to higher telephone and ISP charges, which quickly become prohibitive when baseline fees are 10% of a local wage. Finally, unreliable network infrastructures can sometimes prevent access to the Internet altogether.

These conditions are compounded by the fact that prominent search engines such as Google and AltaVista are designed for reliable, high-bandwidth environments. That is, they optimize for speed, assuming that a user can immediately run a second, modified search if she is unhappy with the results of her first search. This tight feedback loop between the user and search engine is inappropriate for low-connectivity sites where the bottleneck is the time required to transfer the information, rather than the server's delay in finding the information. Also, mainstream search engines select pages without regard for their bandwidth requirements, a criterion which might be of primary interest to someone at the end of a slow connection.

The TEK project aims to address these problems in several ways. We believe that if the results of an Internet search were more affordable, more reliable, and more information-rich, then people would be willing to wait a few days to see them. With TEK – which stands for "Time Equals Knowledge" – a user sends an Internet search via email to a server at MIT, which performs the search using existing search engines, downloads actual pages and emails a subset of

those pages back to the user. To avoid sending the same page to a client a second time, the server keeps track of all pages sent to each client. The server also performs specialized ranking, filtering, and compression of the search results to make them as bandwidth-friendly as possible. Additionally, the system includes a reliable email protocol and a user-friendly interface, which are described below.

2. The TEK System

We have implemented the basic functionality of the TEK system. There are three main components: 1) the TEK client, which provides a graphical user interface for constructing queries and viewing results, 2) the TEK server, which performs searches from MIT and sends the processed results back to the client, and 3) an email-based communication protocol that manages the transfer of information over the unreliable connections between the client and the server.

We will discuss the end-to-end operation of the system in the context of an example: a student who wants to search for information on **solar food dryer**s. The scenario might proceed as follows:

2.1. Entering the Query

We expect that there is one TEK client machine in a school or tele-center and that it supports multiple users. When the student starts TEK, it appears as a set of local web pages that are viewed with a browser. To place a new search, the student must enter her username and password, after which she can also view all of her previous search results. There is a special administrator interface for managing user accounts.

Proceeding with the query, the student enters the search terms, **solar food dryer**, in a web-based form resembling a standard search engine. After the student confirms her search request, the query is placed under the student's "pending query" list and scheduled for mailing. The student can now log off—everything else is done automatically by the system.

2.2. Communicating with the Server

The next step is for the client to email the query to the server. This process can be initiated automatically by the TEK client, perhaps in the middle of the night when the telephone rates are cheapest and there is less demand for phone lines. However, since email delivery is very unreliable in some parts of the world, it is not sufficient to just send the email and expect it to be delivered.

Rather, the client and server follow a communication protocol that is designed to ensure reliable delivery of emails over unreliable connections (Prevost, 2001). Generally speaking, the protocol works by keeping track of which messages have been sent, and which ones were replied to. If a reply is not received within a given time, then the protocol resends the original message.

2.3. Server Processing

When the server receives a search query, it retrieves a set of candidate pages by invoking existing search engines such as Google and AltaVista. Because TEK is optimized for bandwidth, not response time, the server has the time to post-process the pages returned from the search engines. It does this by:

- **Filtering content.** All duplicate pages are removed. Next, all images are removed from the pages unless the user requests to keep them, and all non-essential HTML tags (such as comments and Java script) are deleted. Also, pages that are very similar (such as mirror pages from different sites) are eliminated from the candidate set.

- **Avoiding client-side redundancy.** The server keeps track of which pages have already been sent to the client so as not to waste bandwidth by resending the same version of a page. If the client has an outdated version of a given URL, then a new version is sent to take its place.

- **Clustering.** The candidate pages are grouped into clusters of similar pages; some pages from each cluster are sent to the client. For instance, in the case of the **solar food dryer** query, there could be clusters for pages relating to food to dry, usage of solar dryers and construction methods. Sending some pages from each cluster improves the likelihood that at least some of the information sent will be relevant to the aspect of **solar food dryer**s that the student was most interested in.

- **Identifying high-information pages.** A number of heuristics are applied to determine which pages have the highest "information content." These metrics are distinct from those used by today's search engines – for example, we prefer pages that have keywords appearing in paragraph text instead of links, since in a low-connectivity environment the user cannot readily explore the links.

- **Compressing the result set.** All of the results are compressed into a zip file before sending them back to the client, thereby further reducing the bandwidth needed to download the results.

Following the server's processing, the results are emailed to the client using the communication protocol described above.

2.4. Viewing the Results

When the results arrive on the client, the student needs to login to view them. The interface for viewing results is a special front page – constructed by the server – that organizes the pages by cluster and provides a link to each. The user can then browse through the pages as if they were being retrieved from online.

An added feature of the TEK client is that it accumulates the information from each search into a local digital reference library. This library serves as a miniature, offline version of the Web,

allowing users to follow links from page to page as long as the referenced pages were already downloaded during a preceding search. The user interface provides a local search utility so that the user can search the collection of local pages. Only when the information is not found locally is it necessary to send a query to the TEK server. In other words, if another user had previously searched for the same information, an Internet search can be avoided.

3. Rationale

In this section we argue that the TEK system will make Internet access cheaper, more robust, and, in some respects, even more convenient for users in low-connectivity regions.

3.1. Reduced Cost

There are numerous ways in which TEK will lower the cost of Internet access for the end user. In some regions, email-only accounts are much cheaper than accounts that allow full access to the World Wide Web (see Table 1). Thus, TEK will make Web resources available to those who could otherwise afford only email. In addition, telephone lines are often clearer, more stable, and cheaper to use during off-peak hours; TEK can be set up to run during these times.

Location	ISP	Unlimited Email	15 Hours of Internet Access	Extra Hours of Internet Access
Malawi	Epsilon & Omega	$15 / month	$30 / month	$1.50 / hour
Sri Lanka	LankaNet	$11 / month	$15 / month	$1.32 / hour (peak) $0.88 / hour (off-peak)

Table 1: Email and Internet rates as of July 2001. Sources: www.eomw.net and www.lankanet.org.

The TEK system also decreases costs by shortening the duration of each phone call to the ISP. First, the connection is shortened because the client machine spends all of its time either sending a query or downloading results; unlike Web browsing, there is no idle time during which the user is reading pages or contemplating what to do next. Second, when the results are being downloaded, all of the content is available on the ISP; the user does not have to wait for the ISP to fetch information from other sources. Third, the results themselves are more compact, since they are filtered and compressed on the server side.

Retrieval Method		Price	Price per Mbyte	Relative Cost per Mbyte
Hard Disk		$250 / 75 GB	$0.00325	1.0
28.8 kbs modem (Sri Lanka)	100% utilization	$1.75 / hour	$0.104	32.0
	10% utilization		$1.04	320.0
	1% utilization		$10.4	3200.0
128 kbs Cable / DSL (USA)	100% utilization	$30 / month unlimited access	$0.00074	0.23
	10% utilization		$0.0074	2.3
	1% utilization		$0.074	23.0

Table 2: Estimated costs of local storage vs. remote fetch as of July 2001.

Finally, there will be further savings if some TEK searches can be eliminated altogether – which will happen when the local search utility finds the sought information in the client's local digital library. To emphasize that it is a cost-effective strategy for the client to keep a persistent copy of each page that it downloads from the server, let us consider a few calculations (see Table 2). Assuming that a 75 GB hard disk costs $250 dollars, it follows that one megabyte (MB) of hard disk space costs $0.0032. On the other hand, downloading one MB of data over a 28.8 kbs modem at a rate of $1.75 per hour would cost $0.104 – more than 32 times as much as storing the data on disk! And this figure assumes a perfect utilization of the modem's bandwidth; with a more realistic utilization between 1% and 10%, retrieving pages over the phone becomes three orders of magnitude more expensive than storing them on disk. Thus, even if there is only a 1% chance that a downloaded page will be needed again in the future, it is economically advantageous to buy a hard disk on which to store downloaded pages, rather than planning to download them a second time. Note that, given the Internet prices in the United States, these results are reversed – i.e., there is not an economic incentive to support an extensive client-side digital library.

3.2. Improved Reliability

TEK improves the robustness of Web access by reducing the user's dependence on the ISP's external network. That is, when the user wants to browse the Web in real-time, two connections need to be working: from the user to the ISP, and from the ISP to the rest of the world. However, with an email-based protocol, these connections are decoupled. First, over some period of time, there needs to be a working path from the MIT server to the user's ISP. Then, at some other time, the user needs to connect to the ISP and download the results. In other words, it is possible to obtain a page using TEK even if the page is constantly unavailable to a Web browser using the same ISP.

Assuming that the client sends and receives TEK emails once per day, the user can expect to find the results of a query within 48 hours (since the query will be sent within 24 hours, and the results received within the next 24 hours). In cases where the email is delayed or lost en route, the communication protocol automatically manages the retransmission procedures.

3.3. Improved Convenience

At first glance, it might appear that TEK is inconvenient because of the delay it imposes between searching and receiving the results. However, there are many ways in which using TEK is more convenient than using an online Web browser in a low-connectivity area. Primarily, once the results have arrived via email, one can browse through them all in real-time, instead of enduring the slow, unreliable, and frustrating process of loading each page when one is connected. Further, one can look at the results at any time that is convenient, and the results will remain available to all users of the machine as long as there is space on the hard drive. The results themselves might be more relevant to the user's query, since the TEK server spent more time analyzing and processing the results than conventional, speed-optimized search engines. Finally, TEK's nighttime download feature could free up one's phone line for other uses during daylight hours, as well as avoiding phone line congestion in trying to connect to the ISP during peak hours.

4. Related Work

There are a number of search engines that have something in common with TEK. Google eliminates pages that are very similar; NorthernLight and Vivisimo perform clustering of pages, and MetaCrawler invokes multiple search engines to perform the search. However, all of these search engines are optimized for speed. TEK is fundamentally different in that it is optimized for low-bandwidth and low-connectivity.

Orthogonally, there are a number of email-based services that return text representations of a given web page, with some that provide an interface to search engines (e.g., GetWeb, www4mail, Web^{2}Mail). These services, however, return only the page listing the search results, instead of downloading the discovered pages and passing on the most useful ones to the client. Moreover, they lack two of TEK's key features: 1) a server that records which pages are already on the client, thereby eliminating redundant client/server communication, and 2) a series of specialized information retrieval techniques that filter, analyze, and compress the results on the server before sending them to the client.

5. Discussion and Future Work

The TEK search engine is in its infancy. There are many questions that we will not be able to further research until the system is deployed and we can gather usage statistics. How broad is each location's knowledge needs? How much repetition and overlap is there among queries? What information should initially be included in the local library on the client machine? How do information needs differ in different cultures? While fascinating, these questions must all wait. We have designed the TEK system to be flexible, such that the specific information retrieval techniques it employs can be adjusted depending on observed usage patterns.

However, there are several enhancements that could be made now to the basic system. Because it could take up to two days for the server to notify the client that a query is badly formed, it will be valuable to provide a more sophisticated query builder on the client to help ensure that a query is appropriate – for instance, by detecting spelling errors or estimating the number of pages that would match the given terms.

On the server side, a number of techniques could be explored to improve the quality of the search results. The search terms could be augmented with category information that will direct the search engines to search a subset of the Web. Similarly, the user could provide a document that is similar in format to the one that she is seeking, but on a different subject – for example, she might send a reference to a guide on growing corn when she is seeking a guide to growing rice. In addition, a mechanism to gather feedback from the users of TEK on the usefulness of each returned page will be critical for evaluating the effectiveness of the heuristics employed by the server. The server could even use this information on a client-by-client basis to choose the search methodologies that are best suited to a given user. Finally, we will have to expand TEK to support other languages.

6. Conclusions

TEK is a technical solution to a social need. From its conception, TEK was based on an understanding of the cultural context it needs to serve. While cutting-edge Information Technology tends towards "more information, faster," TEK is designed to work in a low-connectivity, low-bandwidth setting, where the aim is to guarantee the delivery of "better information, slower."

We do not consider TEK to be a permanent solution to the problem of providing Internet access in developing countries. Instead, we believe that there is a need for an interim solution – a more reasonable way for people to access the Internet – while more ambitious and long-term telecommunication initiatives are implemented. By its gains in affordability, reliability, and convenience, we believe that TEK will meet exactly that need: it will bring Web access to some people who would otherwise be without it.

7. Acknowledgements

The TEK system has been designed and developed with the help of the following students: Alexandro Artola, Sheldon Chan, Genevieve T. Cuevas, Mark Halsey, Sid Henderson, Yuliya Litvak, Tazeen Mahtab, Janelle Prevost, Saad Shakhshir and Binh D. Vo. We have had advice and discussions with: David Clark, Michael Dertouzos, David Karger, Jaime Teevan, Lynn Andrea Stein, and Peter Szolovits. Thank you.

This work was partly funded by a Faculty Fellowship from Singapore University, a Graduate Fellowship from Siebel Systems, and the Summer Undergraduate Research Opportunity Program at the MIT Laboratory for Computer Science.

8. References

GetWeb. http://www.satellife.org/webcontent.php

Prevost, Janelle, *A Reliable Low-Bandwidth Email-Based Communication Protocol*, Master's Thesis, Massachusetts Institute of Technology, 2001.

Web for Mail. http://www4mail.org/

Web^{2}Mail. http://www.web2mail.com/

TEK Screenshots: Sending a Query

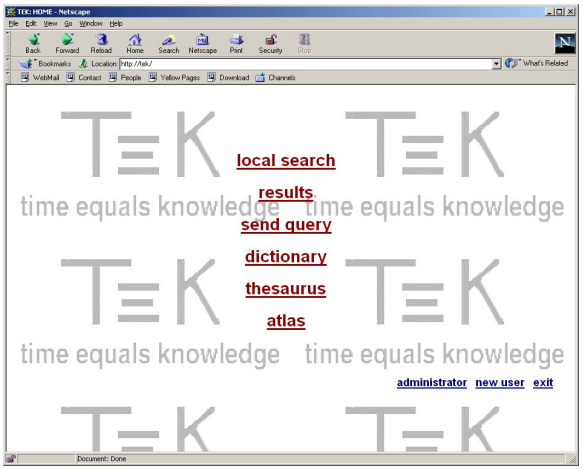

Figure 1: The TEK front page allows users to conduct different types of local searches and remote queries, as well as to view results.

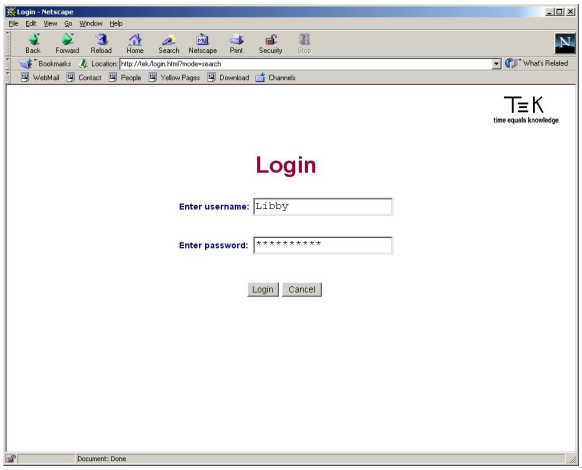

Figure 2: To place a remote query or view results, the user must first login.

Figure 3: After logging in, the user can perform remote searches, including advanced search, specific URL request, and image search.

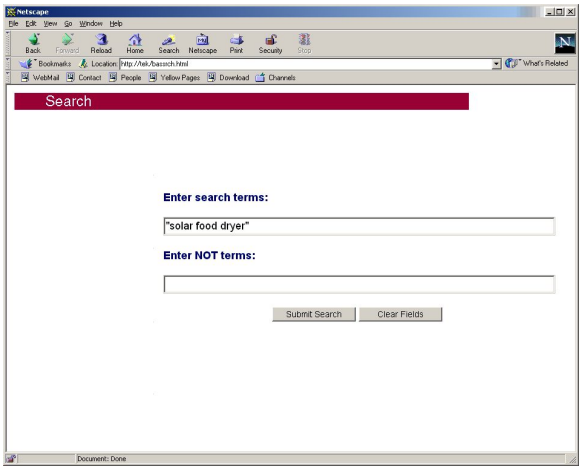

Figure 4: The basic TEK search interface has two fields: one for terms that must appear, and one for terms that must NOT appear.

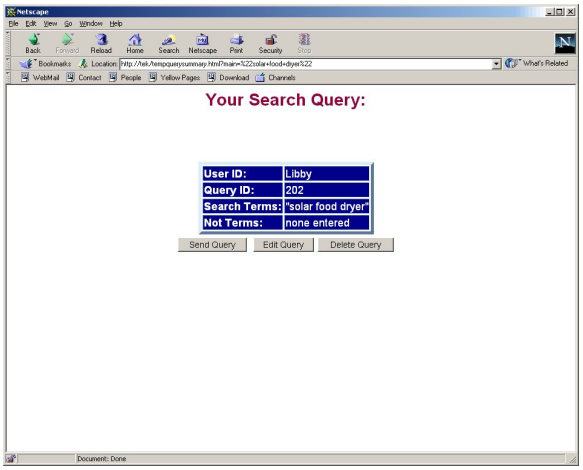

Figure 5: The user is asked to confirm the query.

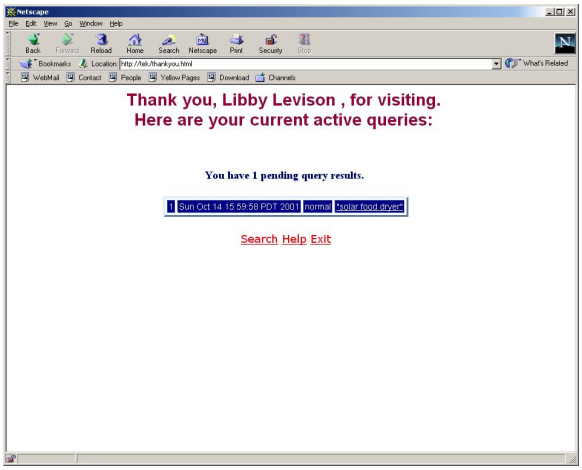

Figure 6: Confirmation that the query is complete.

TEK Screenshots: Viewing Results

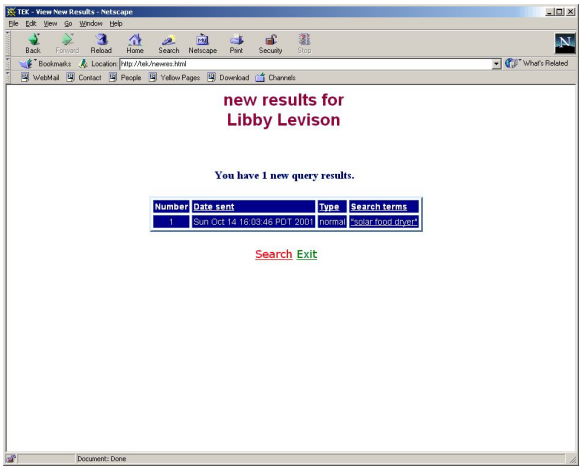

Figure 7: After logging in, the user can see a list of recently returned query results.

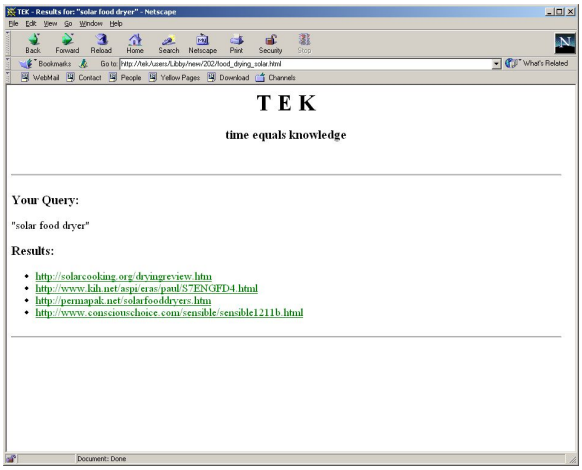

Figure 8: TEK presents the set of pages corresponding to the user's query.

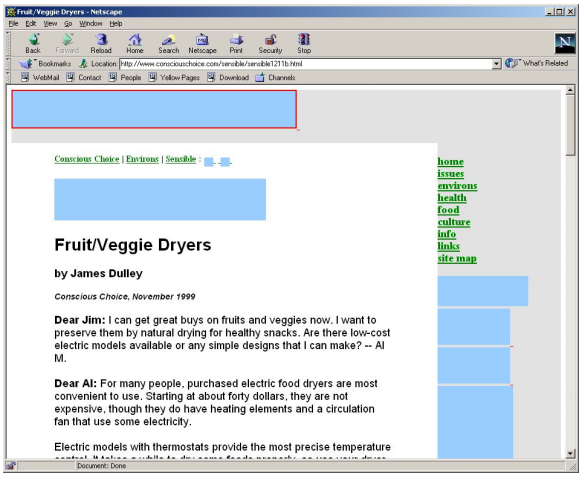

Figure 9: A resulting page, as seen by the user. TEK refines pages, removing images to save bandwidth.

Figure 10: The original, unrefined version of the page seen in Figure 9.

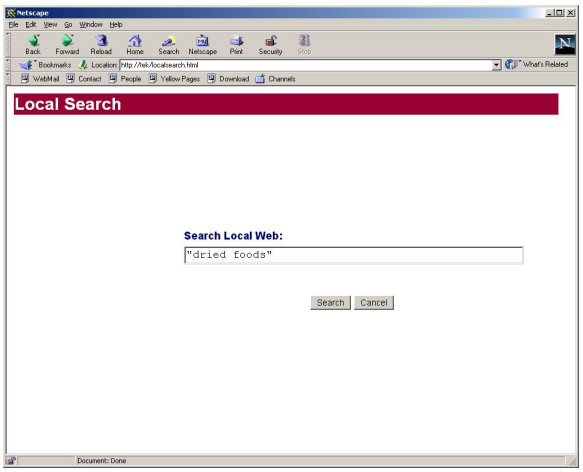

Figure 11: Returned results are stored in the local database, which can be searched with a local engine.

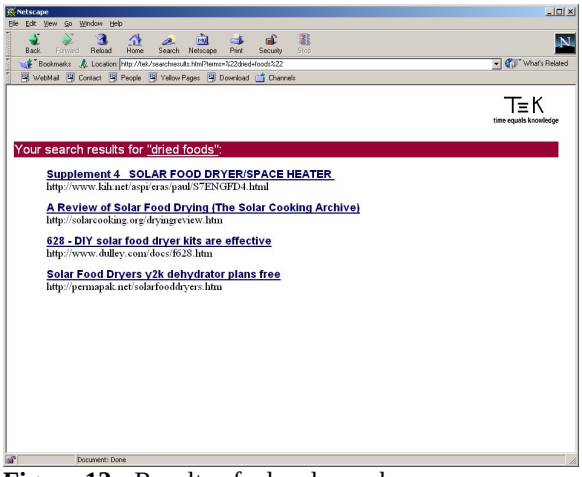

Figure 12: Results of a local search.