US 20090313241A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2009/0313241 A1**

(43) **Pub. Date: Dec. 17, 2009**

Elsen

(54) **SEEDING SEARCH ENGINE CRAWLERS USING INTERCEPTED NETWORK TRAFFIC**

(75) Inventor: **Christian Elsen**, Rolle (CH)

Correspondence Address:
ABELMAN, FRAYNE & SCHWAB
666 THIRD AVENUE, 10TH FLOOR
NEW YORK, NY 10017 (US)

(73) Assignee: **CISCO TECHNOLOGY, INC.**

(21) Appl. No.: **12/214,133**

(22) Filed: **Jun. 16, 2008**

Publication Classification

(51) **Int. Cl.**
G06F 7/06 (2006.01)
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/5**; 707/E17.108

(57) **ABSTRACT**

A method includes monitoring data packets exchanged in a computer network over which documents having respective location identifiers are distributed, so as to detect a request to access a given document. A location identifier of the given document is extracted from the request. The location identifier is provided to a search engine that searches for data in a set of the documents, so as to cause the search engine to add the given document to the set.

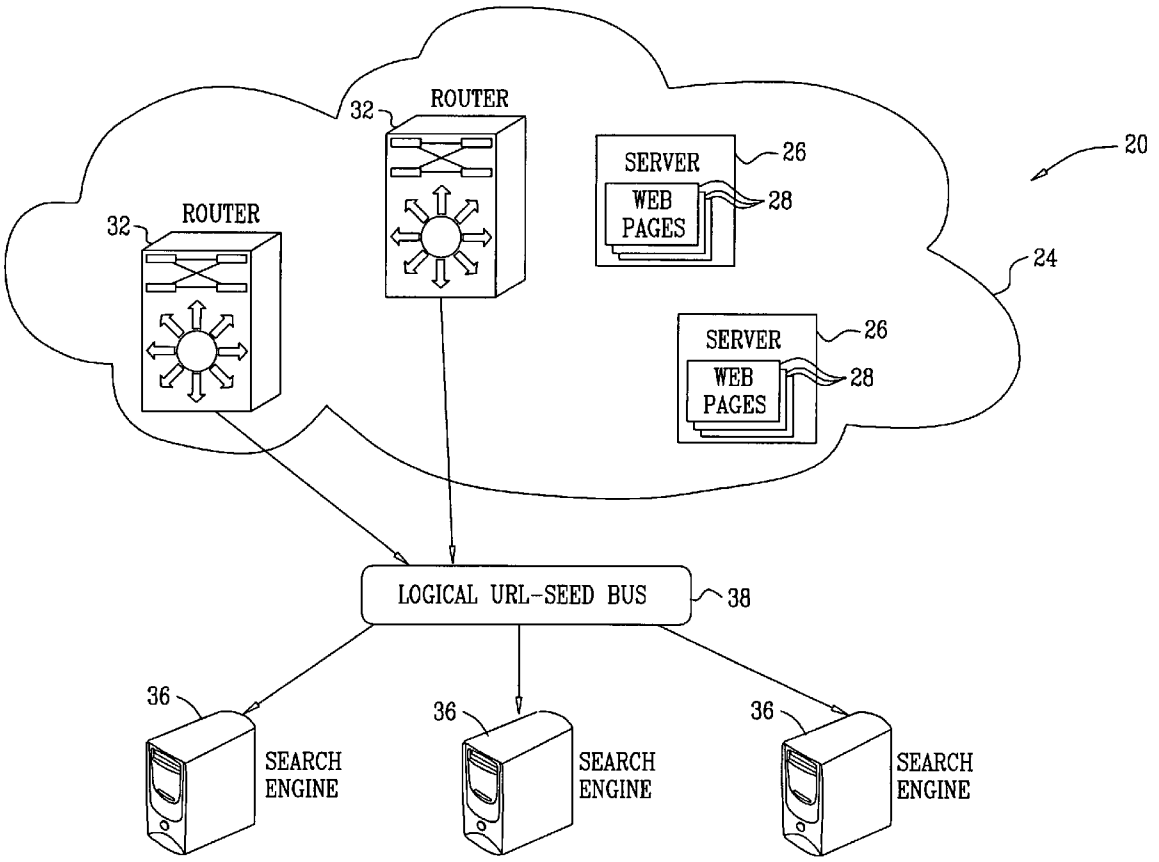


FIG. 1

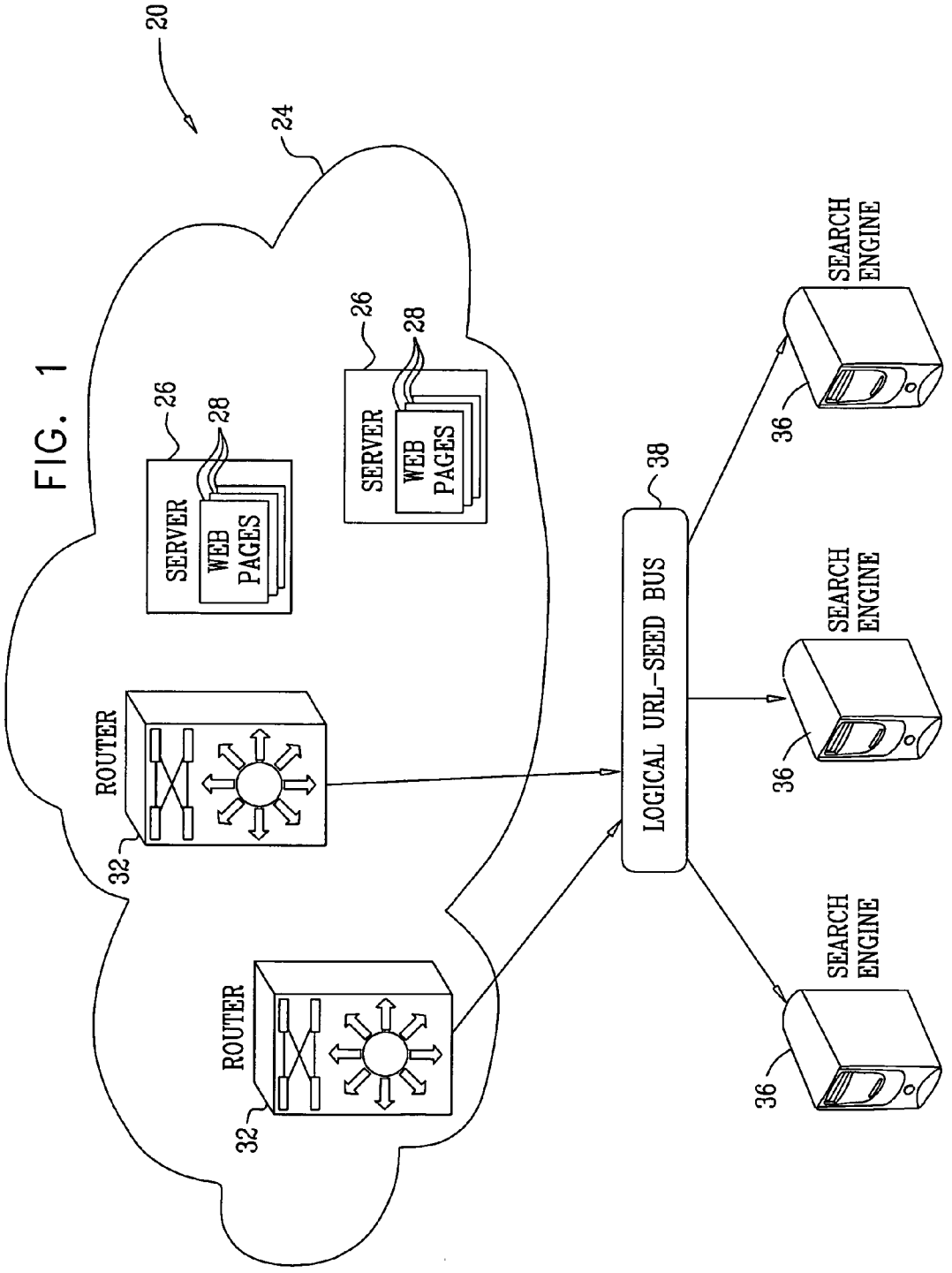

20
24
26
28
SERVER
WEB PAGES
26
28
SERVER
WEB PAGES
ROUTER
32
ROUTER
32
38
LOGICAL URL-SEED BUS
SEARCH ENGINE
36
SEARCH ENGINE
36
SEARCH ENGINE
36

FIG. 2

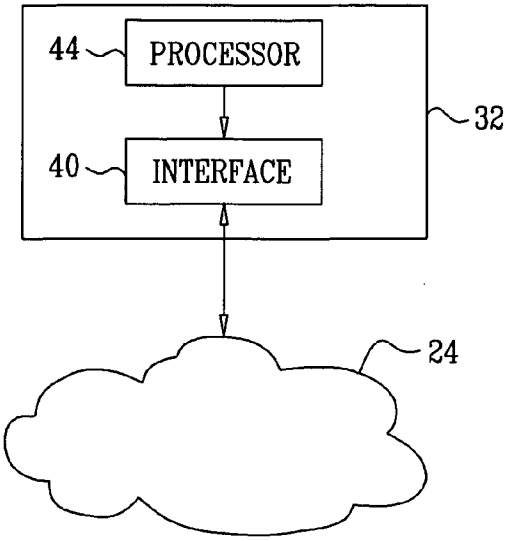
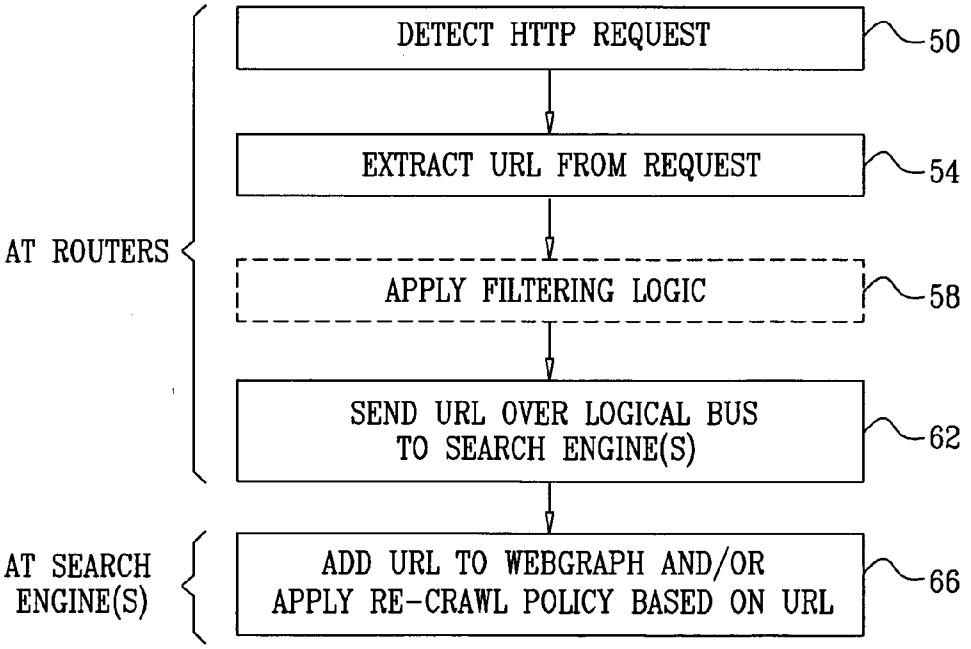

44
PROCESSOR
40
INTERFACE
32
24

FIG. 3


DETECT HTTP REQUEST
50
EXTRACT URL FROM REQUEST
54
AT ROUTERS
APPLY FILTERING LOGIC
58
SEND URL OVER LOGICAL BUS TO SEARCH ENGINE(S)
62
AT SEARCH ENGINE(S)
ADD URL TO WEBGRAPH AND/OR APPLY RE-CRAWL POLICY BASED ON URL
66

SEEDING SEARCH ENGINE CRAWLERS USING INTERCEPTED NETWORK TRAFFIC

FIELD OF THE INVENTION

[0001] The present invention relates generally to computer networks, and particularly to methods and systems for searching for data in computer networks.

BACKGROUND OF THE INVENTION

[0002] Various kinds of search engines are deployed extensively in computer networks. For example, some search engines gradually map the network by following links that point from one data page to another in order to traverse the network, and index the data stored therein. Such search engines are often referred to as "web-crawling" engines. A web-crawling search engine typically maintains a data structure, known as a web-graph, which represents the link relationships in the set of traversed pages.

[0003] The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. **1** is a block diagram that schematically illustrates a system for searching for data in a computer network, in accordance with an embodiment of the present invention;

[0005] FIG. **2** is a block diagram that schematically illustrates a network router, in accordance with an embodiment of the present invention; and

[0006] FIG. **3** is a flow chart that schematically illustrates a method for seeding a web-crawling search engine, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

Overview

[0007] A web-crawling search engine typically begins traversing a searched computer network with a web-graph, which begins from a set of "seed" pages that are provided a priori. The search engine follows links in the seed pages that point to other pages, adds the linked pages to its web-graph, and continues to expand the web-graph by following links in the newly-added pages.

[0008] As can be appreciated, such a search engine can index and search only pages that belong to its web-graph. Pages that are not linked to the seed pages, directly or indirectly, will not be reached at all using conventional web-crawling processes. Other pages, which are linked to the seed pages, may not exist in the web-graph at a given point in time because the crawling process has not reached them yet. Regardless of the reason for not belonging to the web-graph, pages that do not exist in the web-graph may hold important information that might be missed by the search engine, either temporarily or permanently.

[0009] Embodiments of the present invention that are described hereinbelow provide improved methods and systems for supplying seed pages to web-crawling search engines. The methods and systems described herein identify pages that are accessed by network users and report the identified pages to the search engine, so as to cause the search engine to update its web-graph with these pages. Using the disclosed techniques, web-crawling search engines are able to reach pages that are not linked to the initial seed pages.

[0010] The methods and systems described herein are typically deployed in network elements that process data packets in the computer network, such as in routers, multilayer switches or any other suitable device. In a typical implementation, a network element monitors data packets that are exchanged in the computer network. The network element detects a request to access a given document, e.g., a Hyper-Text Transfer Protocol (HTTP) request to access a certain Web page. The network element extracts a location identifier of the given document (e.g., a Uniform Resource Locator—URL) from the request. The network element sends the extracted identifier to the search engine, so as to cause the search engine to add the given document to the web-graph.

[0011] In some embodiments, the network element filters or otherwise pre-processes the extracted identifiers before sending them to the search engine. In other embodiments, all extracted identifiers are sent to the search engine without filtering. In either case, the search engine typically chooses whether or not to add the reported identifiers to its web-graph.

[0012] In some embodiments, the reported identifiers are used by the search engine in determining when to revisit ("re-crawl") a certain document that already exists in the web-graph. For example, if the search engine receives frequent reports on a given document, it may assume that the content of the document may have changed, and thus decide to revisit it in order to capture the potentially-new content. In some embodiments, one or more network elements capture and report identifiers to one or more search engines using a reporting protocol, referred to as a logical bus.

[0013] Thus, the methods and systems described herein considerably improve the performance of web-crawling search engines. In some embodiments, the methods and systems described herein can be deployed with little or no modification to the search engine. Moreover, the methods and systems described herein may be implemented with little or no modification of network element hardware.

System Description

[0014] FIG. **1** is a block diagram that schematically illustrates a system **20** for searching in a computer network **24**, in accordance with an embodiment of the present invention. Network **24** may comprise, for example, a Wide-Area Network (WAN) such as the Internet, a Metropolitan-Area Network (MAN), a Local-Area Network (LAN) or a combination of such network types. Network **24** may comprise a public network or an enterprise network (sometimes referred to as an Intranet). Additionally or alternatively, network **24** may comprise any other suitable network type. The network typically comprises a packet-switched network, such as an Internet Protocol (IP) network.

[0015] Network **24** comprises servers **26**, which store data in Web pages **28**. Each page is assigned a unique location identifier, such as a Uniform Resource Locator (URL). In some embodiments, the servers host Web pages that are produced a-priori. In alternative embodiments, the servers generate Web pages directly based on user input. The methods and systems described herein can be used in any suitable network over which documents are distributed, regardless of whether the documents are stored a-priori or generated on-demand. Although the exemplary embodiment of FIG. **1** refers to servers, the methods and systems described herein can be used with any other sort of storage or computing

devices known in the art. Moreover, although the embodiments described herein refer to Web pages, the disclosed methods and systems can be used with any other suitable type of document. In the context of the present patent application and in the claims, the term "document" refers to any kind of data resource having a location identifier, such as, for example, a file, a Web page, a database record, a web service or another generic computing service.

[0016] Network **24** comprises network elements, such as routers **32**, which perform routing or forwarding of data packets in the network. Although the description that follows refers to network routers, the methods and systems described herein can be used with various other kinds of network elements that process data packets, such as switches or gateways.

[0017] System **20** comprises one or more search engines **36**, which search for data in network **24** in response to user queries. Search engines **36** use web-crawling techniques, as are known in the art. For example, search engine **36** may comprise a Google™ search engine, which is provided by Google, Inc., (Mountain View, Calif.), or the open-source Nutch search engine provides by the Apache Software Foundation. Search engines **36** may comprise different instances of a certain search engine (e.g., multiple Google Appliance boxes) and/or search engines of different types.

[0018] Each search engine **36** maintains a web-graph or equivalent data structure, which represents a set of pages that are currently known to the search engine and the links between them. The search engine searches for data in the set of pages, typically by (1) producing an index that maps words to the pages in which they appear, and (2) querying the index in response to user queries.

[0019] The search engine creates the web-graph in a progressive manner. The search engine is initially provided with a set of pages, e.g., a set of popular Web pages, which are referred to as a seed. The search engine "crawls" the Web by following links that appear in the seed pages and adding the linked pages to the web-graph. When a page is added to the web-graph, the search engine updates the index with the words that are found in this page. The crawling process continues in a progressive manner by following the links in the newly-added pages, so that the web-graph is expanded constantly. Since page content may change over time, the search engine typically performs re-crawling, i.e., revisits pages that already exist in the web-graph, in accordance with a certain re-crawling policy.

[0020] As can be appreciated, search engine **36** can index and search only pages that belong to its web-graph. Pages that do not exist in the web-graph will not be indexed and the data in these pages cannot be retrieved.

[0021] Embodiments of the present invention provide improved methods and systems for adding pages to the web-graphs of search engines **36**. As will be described in detail further below, routers **32** (or other network elements in network **24**) monitor data packets exchanged in the network, in order to detect requests from users to access Web pages **28**. When a router detects a request to access a certain page, it extracts an identifier of the requested page from the request, and forwards the identifier to the search engines. The search engines may choose to add the reported pages to their web-graphs. Thus, pages that are not linked to the seed pages, but are requested by users, can be reached, indexed and searched by the search engines.

[0022] In some embodiments, the routers send the identifiers to the search engines using a logical bus **38**. Bus **38** comprises a communication protocol that is supported by the network elements and the search engines. In some embodiments, the logical bus may be implemented using known mechanisms and protocols, such as using multicast packet transmission.

[0023] FIG. **2** is a block diagram that schematically illustrates router **32**, in accordance with an embodiment of the present invention. Router **32** in the present example comprises a network interface **40** for communicating with network **24**, and a processor **44** that carries out the methods described herein.

[0024] Processor **44** may be implemented using hardware components, using software, or using a combination of hardware and software elements. In some embodiments, the functions of detecting requests, extracting identifiers and sending them to the search engines are carried out by the same processor or group of processors that perform conventional routing functions of router **32**. Alternatively, request detection, identifier extraction and sending can be implemented using a separate, dedicated processor. Typically, the processor comprises a general-purpose processor, which is programmed in software to carry out the functions described herein. The software may be downloaded to the processor in electronic form, over a network, for example, or it may, alternatively or additionally, be provided and/or stored on tangible media, such as magnetic, optical, or electronic memory.

Seeding Search Engine Crawler Using Intercepted Network Traffic

[0025] FIG. **3** is a flow chart that schematically illustrates a method for seeding search engine **36**, in accordance with an embodiment of the present invention. The example of FIG. **3** refers to a search engine that searches Web pages on the Internet, and to routers or multilayer switches that identify Hyper-Text transfer Protocol (HTTP) requests that indicate Uniform Resource Locators (URL) of requested Web pages. In alternative embodiments, the method of FIG. **3** can be used with search engines that search other types of networks and/or other types of documents. The detected requests may comprise any other suitable type of request. The extracted identifier may comprise not only a URL, but also any other suitable type of identifier, that is a Uniform Resource Identifier (URI).

[0026] Various techniques for detecting requests and for extracting URLs from requests are known in the art, and any suitable method can be used. Such techniques are used, for example, in Network Intrusion Detection Systems (NIDS). Some of these processes can be implemented at wire-speed, even for high-speed networks such as 10-Gigabit Ethernet networks, using suitable Application-Specific Integrated Circuits (ASICs) or Field-Programmable Gate Arrays (FPGAs). One exemplary process that can be used for detecting requests is commonly known as Deep Packet Inspection (DPI). A typical DPI process examines the data and/or header of a packet as it passes a certain inspection point. A DPI process can search for predefined criteria, such as for a HTTP request, and pass the corresponding packet to another process for extraction of the request URL.

[0027] Various methods and systems for implementing Deep Packet Inspection points within IP network nodes are known in the art. In some implementations, DPI functionality can be integrated into a network node. For example, Cisco Systems, Inc. (San Jose, Calif.) offers a series of network

switches called Catalyst 6500. DPI functionality can be integrated into such switches using a component called Cisco Catalyst 6500 Supervisor Engine **32** Programmable Intelligent Services Accelerator (PISA). In alternative implementations, DPI functionality can be carried out by a standalone component, e.g., by a device that is introduced into the traffic path between two network nodes or by mirroring the inbound or outbound traffic of a network node to such a device. A standalone device that implements DPI may comprise, for example, an SCE 2000 Series Service Control Engine, offered by Cisco Systems, Inc. Thus, the methods described herein can be carried out by one or more network elements, which may or may not be physically collocated. The processors of these network elements are collectively regarded herein as a processor that carries out the disclosed methods.

[0028] The method of FIG. **3** begins with router **32** detecting an HTTP request, at a request detection step **50**. The detected HTTP request is typically sent from a user of network **24**, requesting to access a certain Web page **28** that is stored in the network. The HTTP request comprises a URL of the requested page. The router extracts the URL from the request, at an identifier extraction step **54**.

[0029] In some embodiments, router **32** may apply filtering to the extracted URLs, at a filtering step **58**. In other words, the router may evaluate a certain condition with respect to the extracted URL, and send the URL to the search engine only when the condition is met. The condition may depend on the time that elapsed between the detection of the request and the detection of a previous request to access the same page (i.e., a previous request carrying the same URL). For example, the router may send a given URL to the search engine only if the page was not previously requested within a predefined time interval. This technique avoids sending duplicate reports of the same URL, and may assist in reducing the amount of traffic between the routers and search engine. In alternative embodiments, all extracted URLs are sent to the search engine without filtering.

[0030] Additionally or alternatively to filtering multiple requests of the same URL, the router may count the number of occurrences and report this number to the search engine. For example, the router may accumulate requests that carry a given URL over a certain period of time, and send a cumulative report to the search engine. The cumulative report indicates the URL in question, and the number of detected requests that carry this URL. As noted above, the router sends the URL to the search engine using logical bus **38**, at a URL reporting step **62**.

[0031] Search engine **36** may update its web-graph (i.e., to the set of searched pages) in response to the URL sent by router **32**, at a web-graph updating step **66**. In some embodiments, the search engine adds the page indicated by the URL to the web-graph, assuming the page does not already exist in the web-graph. From this stage, the crawling process will follow links that appear in the newly-added page. Thus, the newly-added page forms an additional seed page of the web-graph. The crawling process will eventually add the pages linked to the newly-added page to the web-graph, so that these pages are reachable to the search engine. Such pages may have been impossible to reach before the URL was reported, for example if the newly-added page was not linked to the pages of the web-graph in any way.

[0032] In some embodiments, the search engine decides if and when to revisit a page that already exists in the web-graph based on the reported URLs. For example, if the search engine identifies that a certain page is reported frequently, the search engine may conclude that the content of this page may have changed. The search engine may decide to revisit ("re-crawl") this page, and update the index to reflect the new content. Generally speaking, the search engine may decide to search pages that already exist in its web-graph in response to the reported URLs, irrespective of whether these pages have already been searched before. Additionally or alternatively, the search engine may apply any other suitable re-crawling policy in response to the reported URLs.

[0033] Generally speaking, the specific actions taken by the search engine are determined independently of the routers. In particular, each search engine may decide whether to add or revisit a page upon receiving a URL from the routers. Typically, the routers have no information as to whether or not a given page exists in the web-graph of a certain search engine.

[0034] Note that a given search engine may update its web-graph with respect to a given page (e.g., add the page to the web-graph or decide to re-crawl the page) in response to reports sent from the same router or from different routers. Different search engines may exercise different policies and may produce different web-graphs based on the same URL reports from the routers.

[0035] Although the embodiments described herein mainly address seeding of web-crawling search engines, the principles of the present invention can also be used for additional applications, such as for controlling the re-crawl frequency for a given Web page.

[0036] It will thus be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and sub-combinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

1. A method, comprising:

monitoring data packets exchanged in a computer network over which documents having respective location identifiers are distributed, so as to detect a request to access a given document;

extracting a location identifier of the given document from the request; and

providing the location identifier to a search engine that searches for data in a set of the documents, so as to cause the search engine to add the given document to the set.

2. The method according to claim **1**, and comprising, after the given document is added to the set, applying the search engine to search for the data in the set including the given document.

3. The method according to claim **1**, wherein one or more documents in the set contain at least one link pointing to at least one additional document, and wherein the search engine incrementally expands the set by following the at least one link in the at least one document and adding the at least one additional document to the set.

4. The method according to claim **3**, wherein providing the location identifier comprises adding to the set at least one document that is reachable by following one or more links in the given document but that cannot be reached by following the links in the documents in the set.

5. The method according to claim **1**, wherein the given document is already in the set when the location identifier is

provided to the search engine, and wherein providing the location identifier comprises causing the search engine to search for the data in the given document irrespective of whether the given document has already been searched.

6. The method according to claim **1**, wherein providing the location identifier comprises evaluating a condition related to the extracted location identifier, and providing the location identifier only responsively to meeting the condition.

7. The method according to claim **6**, wherein evaluating the condition comprises defining the condition responsively to a time interval between detection of the request and the detection of a previous request to access the given document.

8. The method according to claim **1**, wherein monitoring the data packets comprises detecting multiple requests to access the given document, and wherein providing the location identifier comprises reporting an amount of the multiple requests to the search engine.

9. The method according to claim **1**, wherein the documents comprise Web pages.

10. The method according to claim **1**, wherein the request comprises a Hyper-Text Transfer Protocol (HTTP) request.

11. The method according to claim **1**, wherein the location identifier comprises a Uniform Resource Locator (URL).

12. Apparatus, comprising:

an interface, which is operative to communicate with a computer network over which documents having respective location identifiers are distributed; and

a processor, which is configured to monitor data packets exchanged in the computer network so as to detect a request to access a given document stored in the computer network, to extract a location identifier of the given document from the request, and to provide the location identifier to a search engine that searches for data in a set of the documents so as to cause the search engine to add the given document to the set.

13. The apparatus according to claim **12**, wherein the given document is already in the set when the location identifier is provided to the search engine, and wherein the processor is configured to cause the search engine to search for the data in the given document irrespective of whether the given document has already been searched.

14. The apparatus according to claim **12**, wherein the processor is configured to evaluate a condition related to the extracted location identifier, and to provide the location identifier only responsively to meeting the condition.

15. The apparatus according to claim **14**, wherein the processor is configured to define the condition responsively to a time interval between detection of the request and the detection of a previous request to access the given document.

16. The apparatus according to claim **12**, wherein the processor is configured to detect multiple requests to access the given document, and to report an amount of the multiple requests to the search engine.

17. The apparatus according to claim **12**, wherein the documents comprise Web pages.

18. The apparatus according to claim **12**, wherein the request comprises a Hyper-Text Transfer Protocol (HTTP) request.

19. The apparatus according to claim **12**, wherein the location identifier comprises a Uniform Resource Locator (URL).

20. A system, comprising:

a network element, which is configured to monitor data packets exchanged in a computer network over which documents having respective location identifiers are distributed, to detect in the monitored data packets a request to access a given document stored in the computer network, to extract a location identifier of the given document from the request and to send the extracted location identifier over the computer network; and

a search engine, which is coupled to search for data in a set of the documents, to accept the location identifier from the network element and, responsively to the location identifier, to add the given document to the set.

* * * * *