

(19) World Intellectual Property Organization  
International Bureau



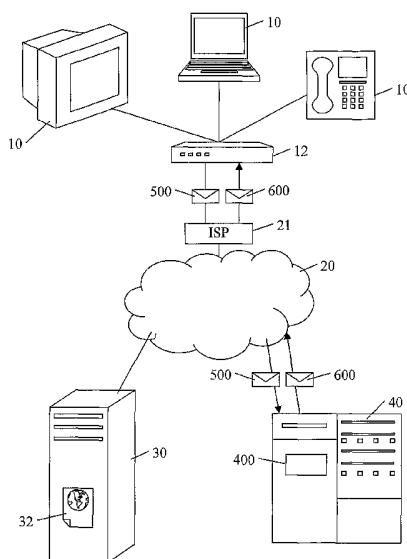
(43) International Publication Date  
16 March 2006 (16.03.2006)

PCT

(10) International Publication Number  
**WO 2006/027590 A1**

- (51) International Patent Classification<sup>7</sup>: **H04L 29/08**
- (21) International Application Number:  
PCT/GB2005/003465
- (22) International Filing Date:  
9 September 2005 (09.09.2005)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
0420024.2 9 September 2004 (09.09.2004) GB  
10/952,626 28 September 2004 (28.09.2004) US
- (71) Applicant (for all designated States except US): **SURF-CONTROL PLC** [GB/GB]; Riverside, Mountbatten Way, Congleton, Cheshire CW12 1DY (GB).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **SINCLAIR, John** [GB/GB]; 47 Marlborough Drive, Rhos-on-Sea, Conwy, Wales (GB). **PETTENER, Ian, James** [GB/GB]; 4 Sefton Gardens, Aughton, Ormskirk, Lancashire L39 6RZ (GB). **NASH, Alistair** [GB/GB]; 6 Bickerton Road, Altrincham, Cheshire WA14 4LA (GB).
- (74) Agents: **ROBINSON, Ian, Michael** et al.; Appleyard Lees, 15 Clare Road, Halifax HX1 2HY (GB).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— with international search report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM, METHOD AND APPARATUS FOR USE IN MONITORING OR CONTROLLING INTERNET ACCESS



(57) Abstract: An apparatus method and system are described for categorising Uniform Resource Locators (URLs) when accessing the Internet (20) from a client (12). A request message (500) is generated to request categorisation of a specified URL, and a category is provided in a reply message (600). The specified URL and the category are added to a category cache (125). In a second or subsequent instance of a URL request referring to the specified URL, the category is determined from the category cache (125). This reduces communication traffic in a network such as the Internet (20).

WO 2006/027590 A1

**SYSTEM, METHOD AND APPARATUS FOR USE IN  
MONITORING OR CONTROLLING INTERNET ACCESS**

The present invention relates in general to a system,  
5 method and apparatus for use in monitoring or controlling  
Internet access. In particular, the present invention  
relates to a system, method and apparatus for categorising  
Uniform Resource Locators (URLs) during Internet access.

10 The Internet is a global interconnection of computers  
and computer networks. One of the great benefits of the  
Internet is that many millions of users have access to  
shared information of the World Wide Web, whereby pages of  
text and graphic information in HTML or other formats are  
15 transmitted by a Hyper Text Transfer Protocol (HTTP). Each  
web page has a unique address, known as a Uniform Resource  
Locator (URL). The Internet and its supporting structures  
are discussed in detail in Requests for Comments (RFCs),  
available from [www.faqs.org](http://www.faqs.org) and elsewhere. Reference is  
20 made in particular to RFC760 (Internet Protocol) and  
RFC1738 (Uniform Resource Locators).

Although the Internet provides access to a vast amount  
of information, it is widely recognised that open access  
25 at all times to all forms of information is not  
appropriate. For example, many schools and businesses  
provide Internet access for their students and employees.  
However, the school or business is, at least in part,  
responsible for dissemination of information within that  
30 organisation and is usually under an obligation to prevent  
circulation of racist, sexist or other abusive materials.  
This is just one example situation where there is a strong  
need for a measure of control over Internet access. Other

examples include public spaces such as libraries or Internet cafes or public Internet kiosks. Another example is a home environment, where parents may wish to prevent their children accessing adult oriented web pages.

5

Prior art systems are available to address this need for monitoring or controlling access to the Internet. One example system is discussed at US 5,996,011, which describes making a linguistic analysis of a web page before delivering the web page or selected portions thereof to a user. Other approaches include comparing a requested URL against a previously-determined list of forbidden URLs, known as a "deny list". However, both of these approaches require relatively large resources, i.e. a computing platform with a relatively fast processor, a large memory, and plenty of storage space such as a hard disk. The World Wide Web currently contains over 200 million websites, with tens of thousands of new sites being added each week. Each site usually contains many individual web pages. As a result, any form of filtering using "deny lists" requires relatively large storage space. Even a linguistic analysis as in US 5,996,011 requires a relatively large space to store objectionable words or phrases, and requires intensive processor usage in order to maintain reasonable response times.

A further problem arises in that many computer users are not technically literate. Most computer users are not computer experts and would like to be able to use their computer with a minimum of fuss or problems. Hence, it is desired to provide an apparatus, method and system for monitoring or controlling Internet access which is simple, reliable and user friendly.

A Local Area Network (LAN) is often used to connect together computers located in one building or site. In this LAN environment access to the Internet is provided  
5 though a Proxy Server, which receives and services URL requests from within the LAN by communicating with the Internet. Some of the client computers in this LAN environment may have relatively limited resources, such as a dumb terminal or diskless workstation. Another example  
10 is a Personal Digital Assistant or other handheld computing device. In one preferred aspect of the present invention it is desired to provide an apparatus, method and system for monitoring or controlling internet access which is ideally simple, fast and reliable, in this LAN  
15 environment.

Many users, particularly in a small office or home office environment (SOHO) environment, connect to the Internet through an Internet Service Provider (ISP).  
20 Typically, the connection is established through dedicated hardware of an Internet gateway appliance such as a modem or a router. However, there is a strong price pressure on Internet gateway appliances and a strong desire to minimise equipment specification. This means minimising  
25 processor requirements, memory requirements, and storage requirements, all of which are directly contrary to known approaches for monitoring or controlling Internet access. In a preferred aspect of the present invention it is desired to provide an apparatus, method and system for  
30 monitoring or controlling internet access which is ideally simple, fast and reliable, when using an Internet gateway appliance.

Another emerging need relates to Internet appliances which are created to perform a specific dedicated function whilst also being connected to the Internet. One example is a web TV for displaying audiovisual signals. Such Internet appliances are generally intended for use by consumers who have little or no technical knowledge, by providing a simple and easy to use set of controls as opposed to the fully controllable interface of a regular computer. Again, most Internet appliances are designed to minimise processor, memory and storage requirements. In a preferred aspect of the present invention it is desired to provide an apparatus, method and system for monitoring or controlling internet access which is simple, fast and reliable, when using an Internet appliance.

15

An aim of the present invention is to address the disadvantages and problems of the prior art, as discussed above or elsewhere.

According to the present invention there is provided an apparatus, method and system as set forth in the appended claims. Preferred features of the invention will be apparent from the dependent claims, and the description which follows.

25

In a first aspect of the present invention there is provided a method of categorising Uniform Resource Locators (URLs) during Internet access, comprising the steps of: receiving a URL request denoting a specified URL; generating a request message to request categorisation of the specified URL; receiving a reply message denoting a category for the specified URL amongst a predetermined set of categories; adding the specified

30

URL and the category to a category cache; and in a second or subsequent instance of a URL request with respect to the specified URL, determining the category of the specified URL from the category cache.

5

In another aspect of the present invention there is provided a method for use in controlling or monitoring Internet access at a client device by categorising Uniform Resource Locators (URLs), comprising the steps of:  
10 receiving a specified URL; searching a category cache held at the client device using the specified URL as a search key, and returning a category code associated with the specified URL when a match is found for the specified URL; and generating a request message to request a  
15 category code for the specified URL, when a match is not found for the specified URL.

According to another aspect of the present invention there is provided a system for use in controlling or  
20 monitoring of Internet access by categorising Uniform Resource Locators (URLs), comprising: a client device arranged to monitor or control Internet access according to a category code of a specified URL, and including a categorisation module to provide the category code for the  
25 specified URL from a category cache stored at the client device or else generate a request message to request categorisation of the specified URL; and a categorisation server coupled to communicate with the client device and arranged to receive the request message and to send a  
30 reply message identifying a category code for the specified URL.

According to a further aspect of the present invention there is provided a client device, comprising: an interface module arranged to present a URL categorisation function, wherein the interface module is arranged to receive a specified URL from a client software and to return a category code; a category cache holding a plurality of stored URLs and associated category codes, such that matching the specified URL against one of the stored URLs provides the category code; and a communication module arranged to send an outgoing request message to a categorisation server when there is no match in the category cache and to receive and buffer incoming data including a corresponding reply message, wherein the request message comprises the specified URL and the reply message comprises the category code.

Further according to the present invention there is provided a cache structure, comprising: a hash array comprising one or more index elements, each index element comprising a host tree pointer and a hash key derived from a stored URL; and one or more host trees depending from the index elements of the hash array, each host tree comprising one or more tree nodes each holding URL data representing stored URLs and associated category codes; and an age list to list each of the tree nodes by age, wherein the age list comprises, within each tree node, a next pointer (827) and a previous pointer (828) which refer to a next older tree node and a previous newer tree node, respectively.

30

The preferred embodiments of the present invention may be implemented as hardware or software, or a combination of both. Terms such as 'module' or 'unit' used herein may

include, but are not limited to, a hardware device, such as a Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC), which performs certain tasks. Alternatively, elements of the invention may advantageously be configured to reside on an addressable storage medium and be configured to execute on one or more processors. Thus, functional elements of the invention may include, by way of example, components, such as software components, object-oriented software components, class components and task components, processes, functions, attributes, procedures, subroutines, segments of program code, drivers, firmware, microcode, circuitry, data, databases, data structures, tables, arrays, and variables. The functional elements such as the components, modules and units discussed herein may be combined into fewer elements or further separated into additional elements.

For a better understanding of the invention, and to show how embodiments of the same may be carried into effect, reference will now be made, by way of example, to the accompanying diagrammatic drawings in which:

Figure 1 is a schematic overview of a system and apparatus as employed in first preferred embodiments of the present invention;

Figure 2 is a schematic overview of a system and apparatus as employed in second preferred embodiments of the present invention;

Figure 3 shows an example of a uniform resource locator (URL);



Figure 4 shows part of a protocol stack appropriate for communication relating to the Internet;

5 Figure 5 is a schematic view of a preferred method for categorisation of URL requests;

Figure 6 shows a preferred format of a request message packet;

10

Figure 7 shows a preferred format of a reply message packet;

Figure 8 is a schematic overview of an example client gateway apparatus;

15

Figure 9 is a logical representation of a preferred structure of a category cache;

Figure 10 shows example data held within the category cache of Figure 9;

20

Figure 11 is a schematic overview of a preferred categorisation server apparatus;

25

Figure 12 is a schematic overview of a preferred licensing cache structure; and

Figure 13 is a schematic overview of preferred licensing systems.

30

Referring to Figure 1, a schematic overview is shown of a system and apparatus as employed in preferred

embodiments of the present invention. In this first example embodiment, a user machine 10 is connected to the Internet 20 through an Internet gateway appliance or client gateway 12.

5

The preferred embodiments of the present invention are primarily applicable to the World Wide Web, whereby a web page 32 is provided in response to a URL request sent under HTTP. In use, the user machine 10 provides a web browser application which initiates a URL request 11 in order to obtain content, i.e. a web page 32, from a content server or host 30. The web page 32 may take any suitable form, most commonly being text and graphics in HTML format. It will be appreciated however that the present invention is applicable to other forms of content provided over the Internet using URLs, such as file transfers under FTP or connection to a TELNET server.

It is desired to passively monitor and log the requested URLs for inspection later, or perform an active filtering function which determines whether the user machine 10 will receive or display the requested web page 32. To this end, it is useful to place URLs into categories. In a simple example, the categories are either "allow" or "deny". In a more sophisticated example, it is helpful to categorise URLs with greater granularity.

The preferred embodiments of the present invention place each requested URL into one of a predetermined set of categories. Specific downstream actions for controlling or monitoring Internet access, such as filtering or logging functions, are not particularly

30

relevant to the present invention and may take any suitable form.

The preferred embodiment provides eight core  
5 categories such as "adult/sexual explicit", "criminal  
skills", "drugs, alcohol, tobacco", "violence" or  
"weapons", as well as thirty two productivity-related  
categories such as "advertisements", "games", "hobbies and  
recreation" or "kids sites". Providing this predetermined  
10 set of categories allows a more sophisticated rules-based  
filtering or logging function. For example, a rule is used  
to alert an administrator when a request is made for any  
of the core categories, or to block selected productivity  
categories at particular times and allowing access only  
15 say at lunchtimes or outside work hours. To cater for all  
eventualities, the preferred categories may also include  
"don't know" or "not found" options.

The user machine 10 provides input and output  
20 interface functions appropriate for a human user, suitably  
including a display screen, speakers, and control keys or  
GUI. As shown in Figure 1, in one embodiment the user  
machine 10 is a computing platform such as a desktop  
computer, a laptop computer, or a personal digital  
25 assistant (PDA). In another embodiment, the user machine  
10 is a function-specific Internet appliance, such as a  
web-TV. In a third example, the user machine 10 is a  
public Internet kiosk, in this case also shown as  
including a voice telephone.

30

In one embodiment, the user machine 10 and the client  
gateway 12 are formed as physically separate devices and  
communicate by any appropriate wired or wireless link. In

other embodiments the client gateway 12 is integrated within the user machine 10.

As one preferred implementation which is useful particularly in a SOHO type environment, the client gateway 12 suitably includes a modem, such as an analogue, ISDN or ADSL modem, which connects to an Internet Service Provider (ISP) 21 over the plain old telephone system (POTS) or other wired or optical network to provide a network layer connection to the Internet 20. As another example, the client gateway 12 connects to the Internet 20 through a wireless network or cellular mobile network such as GSM or GPRS. In still other embodiments, the client gateway 12 connects to the Internet 20 through an intermediary such as a LAN or WAN, optionally over a virtual private network (VPN).

Referring to Figure 1, in a preferred embodiment the client gateway 12 acts as a router and forwards data packets between computers or computer networks. In this illustrated example of Figure 1, the client gateway 12 directs packets between the user machine 10 and the ISP 21. Routers typically use packet headers and forwarding tables to determine the best path for forwarding each data packet.

The client gateway 12 typically has relatively limited computing resources. In one example embodiment, the client gateway is a router having an Intel IXP422 processor, 64MB RAM and 16MB of Flash memory. There is no hard disk or other large-capacity storage device within the client gateway. The client gateway may also perform other functions, typically acting as a combined modem,

router, firewall, local network switch or VPN client, or any combination thereof. Hence, there is strong competition for resources in order to accommodate some or all of these functions within a single low-cost device.

5

It is desired to offer logging or filtering functions at the client gateway 12, because this is a natural control point between the upstream network of the ISP 21, and the downstream network of the user machine 10. The  
10 monitoring or controlling function relies, as an initial step, on placing requested URLs into categories. However, as just discussed, a problem arises in that the client gateway 12 typically has only limited available processor, memory and storage resources. Hence, there is a strong  
15 need to minimise resources used within the client gateway 12 when providing an Internet access controlling or monitoring function.

Figure 2 shows a second example system and apparatus  
20 as employed in an alternative embodiment of the present invention.

Referring to Figure 2, a client computer 12 is part of a Local Area Network (LAN) which also includes a proxy  
25 server 14 coupled to the Internet 20. The client computer 12 makes URL requests in order to receive web pages from a content server 30 available over the Internet 20. The URL requests are processed through the proxy server 14. It is desired to monitor or control Internet access at the  
30 client computer 12. The present invention is particularly applicable where the client computer 12 has relatively limited processor, memory or storage resources, such as a terminal or a diskless workstation.

Referring now to both Figures 1 and 2, the client 12 (i.e. the client gateway 12 of Figure 1 or the client computer 12 of Figure 2) sends a request message 500 to a server computer 40 hosting a categorisation service 400. The request message 500 identifies a specified URL, such as extracted from a HTTP URL request. This categorisation server 40 identifies one of the predetermined set of categories appropriate to the specified URL, and sends a reply message 600 to the client 12. The reply message 600 identifies the appropriate category, which the client 12 then employs to perform the desired monitoring or controlling function.

This arrangement reduces resource requirements at the client 12, and allows the categorisation server 40 to run on a large and powerful computing system with plenty of processing power, memory and storage space. This categorisation service 400 may take any suitable form. For example, upon receiving the URL categorisation request 500, the categorisation service 400 looks up an appropriate category for the specified URL using a category database. Additionally or alternatively, the categorisation service employs a linguistic or other analysis of the specified URLs to determine an appropriate category, with or without human intervention and review.

A problem arises in that it is desired to reduce delays when requesting a web page 32, while a URL is placed into a predetermined category. Also, in practical embodiments of the present invention, many tens, hundreds or thousands of clients 12 are able to communicate with the categorisation server 40. It is desired to minimise

communication traffic. Also, it is desired to minimise overheads both within the client 12, and within the central categorisation server 40.

## 5 Message Protocol

A first aspect of the present invention concerns an improved protocol for communication between first and second computing platforms, in this example between the client 12 and the categorisation server 40, when making requests to place URLs into categories.

Figure 3 shows the standard format of a uniform resource locator (URL), as described in detail in RFC1738. The URL 200 includes a host portion 202 and a page portion 204. The host portion 202 identifies a particular host (e.g. "www.host.com"), whilst the page portion gives a path to a specific web page (e.g. "/directory/page.html"). A root page (i.e. "www.host.com/") at the host is conveniently shown by giving the host portion 202 as "www.host.com" and the page portion 204 as "/".

Figure 4 shows part of a standard protocol stack appropriate for communication relating to the Internet, as described in more detail in RFC760 and elsewhere. The Internet Protocol (IP) interfaces to a local network protocol, and to higher level protocols for communication between network nodes or hosts. The basic function of the Internet Protocol is to move datagrams from a source address to a destination address.

Various host to host protocols exist, including the hypertext transfer protocol (HTTP) which is used to carry

URL requests and provide web pages 32 for the World Wide Web. However, HTTP has no mechanism to efficiently carry the request messages 500 and the reply messages 600 for categorisations of URLs as employed by the present  
5 invention.

Also, several messaging protocols have been defined. As examples, Figure 4 shows a Transmission Control Protocol (TCP) as defined for example in RFC761 and a User  
10 Datagram Protocol (UDP) as defined for example in RFC768. TCP is ideal for applications which require reliable delivery of data in a specified order. TCP sets up a connection between hosts, which is maintained open for the duration of a session. Whilst reliable, TCP has a  
15 relatively large overhead. By contrast, UDP is a fast and lightweight protocol, but is relatively unreliable. In particular, delivery and duplication protection are not guaranteed. UDP is connectionless, with no handshaking or acknowledgements between hosts. Hence, neither of these  
20 messaging protocols is suited to carrying requests and replies concerning URL categorisation.

Figure 5 is a schematic view of a preferred method for categorisation of URL requests, according to an embodiment  
25 of the present invention. A URL request is received at step 401, and a request message 500 is sent at step 402. A reply message 600 is received at step 403, and a URL category is determined at step 404.

30 In the present invention, the request message 500 and the reply message 600 are each sent as the payload of a UDP packet. Surprisingly, it has been found that the unreliable and limited messaging capability of UDP can be



employed to advantage in the context of categorisation of URLs. However, in order to use UDP, additional steps are taken by the present invention to adapt the protocol. More detailed explanation of the request message 500 and the  
5 reply message 600 now follows.

Figure 6 shows a preferred format of the request message packet 500, which includes an Ethernet packet header 501, an IP header 502, a UDP header 503, a UDP  
10 payload 504, and an Ethernet trailer 505. These are all formatted according to existing protocols.

As shown in Figure 6, the UDP payload 504 is divided to form a request message header section 510 and a request  
15 message data section 520.

The header section 510 comprises a sequence number 511 and a time stamp 512, and suitably a command identity 513, a data size 514, and a licensing field 515.  
20

The sequence number 511 allows the request message 500 to be uniquely identified and distinguished from other request messages. The sequence number 511 is generated upon creation of the request message 500 within the client  
25 12, suitably as an incremental value circling between 0 and 65535. Under UDP, each client-side socket exists only for the duration of a request-reply cycle and hence each request is assigned a different port value by the host process within, in this example, the client 12. However,  
30 there is a possibility that a reply could be passed back to a port of an incorrect waiting thread. The sequence number 511 allows a reply to be matched up with an originating request message 500.

The time stamp 512 enables calculation of timeouts. The client 12 originating the request message 500 waits a predetermined length of time for a reply message 600, and then re-tries for a predetermined number of times. Preferably, the timeout is increased after each resend, with an exponential back off (e.g. 2, 4 and then 8 seconds for a maximum retry count of 3).

10 The sequence number 511 and the time stamp 512 together provide excellent reliability, whilst adding only minimal overhead.

The command ID field 513 allows the request message to perform different command functions. In most cases, the command ID is set to "1" in order to request categorisation of a URL. Also, the request message uses a command ID of "2" to request that the categorisation server 40 provide a current list of categories, or a command ID of "3" to confirm a current list version and determine whether an update is required. Other commands can be defined as appropriate. Hence, the command ID field 513 brings increased flexibility and allows the system to perform additional functions.

25

The data section 520 contains data representing a specified URL 200. The URL data 520 includes a host portion 202 and, where appropriate, a URL path portion 204. The request data 520 is encrypted, preferably with a secret-key block encryption algorithm such as RC2 which is described in detail at RFC2268. Encryption of the data section 520 improves security and privacy. However, encrypting only the data section 520 minimises both

encryption workload and transmission overhead. The size of the encrypted data section 520 is stored as the data size field 514 in the request header 510

5       The licensing field 515 optionally transmits a licence identity relevant to the originator of the request message 500. The licence identity is suitably associated with the client 12 or optionally the user machine 10.

10       Figure 7 is a schematic representation of a reply message 600 as generated by the categorisation server 40 and sent to the client 12. The reply message 600 includes a UDP payload comprising a response header 610 and a response data section 620. The response header 610  
15       comprises a sequence number 611 and a time stamp 612, preferably with a command ID 613, all copied from a corresponding received categorisation request message 500. A data size 614 gives a size of the following response data section 620. A status code 615 denotes a status.  
20       This is usually simply "success", but occasionally relates to one of a predetermined set of error statuses.

      The response data 620 is formatted according to the relevant command ID 613 and is preferably encrypted, such  
25       as with RC2. In response to a request to categorise URL, the response data 620 comprises a category 621, a match length 622, and an exact flag 623. The category 621 identifies one amongst a predetermined set of categories for the URL sent in the request data 520, suitably as a  
30       numerical value (e.g. category "27" is say sports related web pages). The exact flag 623 determines whether the requested URL 520 was matched exactly. If only a partial match was obtained, such as a match with only the host

portion 202 or only part of the URL path 204, then a match length is given in the match length field 622. The match length determines a number of characters of the specified URL 520 which were matched with a stored URL at the server 40. The character count is taken along the host portion 202 or the path portion 204, or both. In the preferred embodiment, the count is taken along the path portion 204 only. A match on the root page "/" counts as one character.

10

In response to other command types, the response data 620 contains other data such as a category list specifying a predetermined list of categories, or a version identity which identifies a current version of the category list being used by the categorisation server 40. These other command types can be used to trigger software or configuration updates at the client 12.

As shown in Figures 6 and 7, the request message 500 and reply message 600 each use the payload section of a UDP packet, which usually has a maximum size of 65Kb as defined by the MTU (Maximum Transmission Unit) of the network. By contrast, the Ethernet physical layer packet has a maximum size of just 1500 bytes. Even so, in the present invention almost all of the request and reply messages 500,600 for categorisation of URLs fit within the very limited size constraints of a single Ethernet packet, thus avoiding fragmentation.

Figure 8 shows the client 12 in more detail, including an interface module 121, a communication module 122, a protocol module 123 and an encryption module 124. The interface module 121 presents the URL categorisation

function to a client application, such as to a web browser or a HTTP function (not shown). The interface is suitably an API (application programming interface) to the client software. The interface module 121 is passed a URL from the client software, and returns a categorisation code 621, preferably with a match length 622 and an exact flag 623. The communication module 122 sends outgoing data to the categorisation server 40 and receives and buffers incoming data, including making retransmission requests as necessary. The protocol module 123 interprets the incoming and outgoing data according to the protocol discussed above with reference to Figures 5, 6 & 7 and makes encryption/decryption calls to the encryption module 124. The encryption module 124 encrypts and decrypts data.

15

In the preferred embodiment, the communication module 122 calculates a retransmission timeout for every sent request. To be effective, it is desired that the timeout interval take account of vastly varying network conditions, and adapt accordingly. This helps to eliminate both unnecessary retransmissions and unrealistically high timeout periods. Optionally, the number of retries is configurable such as through a user interface.

25

The preferred method for calculating the retransmission timeout "rto" includes (a) measuring the round-trip time "mt" for each request, (b) maintaining a estimate of the smoothed round-trip time "srtt", and (c) maintaining an estimate of the smoothed mean deviation "smd". The estimates are calculated as:

$$\text{srtt}' = \text{srtt} + (\text{abs}(\text{mt} - \text{srtt}) / 8)$$

$$\text{smd}' = \text{smd} + ((\text{abs}(\text{mt}-\text{srtt}) - \text{smd})/4)$$

From these estimates, the timeout value is calculated as:

5

$$\text{rto} = \text{srtt} + 4(\text{smd})$$

Advantageously, this formula is quickly calculated using fixed-point arithmetic and bit shifts.

10

If any time-out period rto expires, then next timeout is exponentially increased by:

$$\text{rto}' = \text{rto} * 2$$

15

The preferred embodiment of the present invention has many advantages, including in particular minimising overhead when requesting categorisation of URL requests and minimising workload at the gateway appliance 12. The preferred embodiment employs UDP for speed and simplicity, whilst adding a sequence number and time stamp to improve reliability.

20

### Cache

25

In another aspect of the present invention, it is desired to further reduce network traffic over the Internet 20 when placing requested URLs into categories.

30

Figure 8 shows that the client 12 preferably comprises a category cache 125. The category cache 125 stores URL categories by storing response data 620 from each categorisation request 500. Since users often navigate to

a limited set of favourite web pages time and again, the category cache 125 significantly reduces traffic over the Internet 20 by avoiding duplication of requests for categorisation of the same URL or a child page from the same host or directory.

Figure 9 is a logical representation showing a preferred structure of the category cache 125. The cache is structured for both lookups of stored URLs, and also for aging of the cache to ensure that the cache remains within a predetermined maximum memory size. These two functions, namely lookup and aging, are combined so that both share the same nodes in the cache structure, which reduces cache size requirements. As will be discussed in more detail below, the cache 125 is compact and so occupies only a relatively small footprint within the memory of the client 12, whilst still recording valuable data in a manner that is readily searchable and updateable.

20

Referring again to Figure 5, the method of the present invention preferably includes the step 405 of adding the determined URL category to the category cache 125.

In Figure 9, the cache structure comprises a hash array 810, and combined host trees and age list 820. The host portion 202 of each URL is hashed to produce an index 811 in the hash array 810. Many hosts may produce the same hash index 811, and each array element is a pointer to a root tree node of a host tree 820. Hosts with the same hash are searched through the host tree 820, which is preferably a balanced red-black tree where each node has a red/black bit to colour the node red or black. There are

30

n internal nodes and the tree 820 has a height of at most  $2\log_2(n+1)$  so that no leaf is more than twice as far from the root as any other. This is just one example tree structure and many other tree structures are applicable in  
5 embodiments of the present invention.

Each node 821 comprises a host string 822 holding a host portion 202, and optionally an array of pages 823 for the specified host 822. Left and right pointers 825, 826  
10 are used for searching the tree 820. Each node also includes next and previous pointers 827, 828 which refer to a next (older) node and a previous (newer) node, respectively, for aging. Also, each node includes a parent node pointer 824 to allow for fast node deletions.

15

As also shown in Figure 9, the next and previous node pointers 827, 828 allow the nodes to be arranged in order by age. New nodes are added to the head of the age list, and old nodes are removed from the tail. When the cache  
20 is full and has reached a predetermined maximum size, the oldest node is removed to make room for a new URL to be added in a new host node. Conveniently, the age list is refreshed, in order to keep the most recently accessed nodes at the head of the age list.

25

In a preferred embodiment, the memory footprint of the category cache 125 is configured in bytes, in order to determine the maximum size occupied by the hash array 810 and tree list 820. The size may be configured in use  
30 through a control panel, or determined automatically according to needs of the client and thereby balance available resources amongst neighbouring functions.



The hash array 810 has a predetermined length, which is ideally a prime number for better hash distribution. The hash array length is suitably dynamically configurable, such as by being a variable which is input  
5 from a control panel during use. A longer hash array yields faster categorisations, but uses more memory. As examples, the hashing algorithm is suitably MD4 or MD5.

In use, a URL host portion 202 and a URL path 204 are  
10 extracted from a URL request 11 within HTTP or equivalent. The host portion 202 is hashed to determine an index 811 in the hash array 810, and the respective host tree 820 is searched to locate a node 821 matching the host portion 202. The URL path portion 204 is then searched against  
15 the page array 823.

Figure 10 shows example data held in the host string 822 and the page array 823. The host string 822 includes the host portion 902. In some embodiments, a category  
20 code 906 and a children flag 908 are provided for the host, or else these can be presented in a root page. The page array includes, for the or each page, a page string 904, a category code 906 for that page or directory, and a children flag 908.

25

In this example of Figure 10, the host is "www.host.com" and a searched URL path is "/directory\_1/page\_1". The entry for the page string 904  
30 "/directory\_1" has a children flag 908 of "yes" which shows that specific category codes are available for children of this path. The cache shows that "/directory\_1/page\_9" has already been cached, but there is currently no entry for the searched page string

"/directory\_1/page\_1". In this example, the cache 125 has failed to provide a category for the requested URL. A request message 500 is generated to determine the code for the specified URL, i.e. for host "www.host.com" and the path "/directory\_1/page\_1".

As a second example, assume that the children flag 908 for the page "/directory\_1" is set to "no", which allows a cache result to be returned with confidence for the searched page based on a partial match. For example, if the children flag for "/directory\_1" is set to "no", then a confident category code is returned for the requested "/directory\_1/page\_1" based on a partial match with "/directory\_1" as a parent of the requested child page.

15

The cache 125 is suitably built by storing data from request messages 500 and reply messages 600. The request message 500 identifies the specified URL with the host portion 202 and the page portion 204 conveniently provided as a delimited character string. The host portion 202 forms the host string 902. The exact flag 623 determines the children flag 908. The match length field 622 determines a truncation point for the specified URL as a number of characters. The truncated URL is then added to the category cache. For example, the specified URL "www.host.com/directory\_1/page\_1/sub\_page3" is truncated with an exact match at 19 characters to be stored as host = "www.host.com" and page string = "/directory\_1/page\_1". The category code field 621 provides the category code 906.

20  
25  
30

Referring again to Figure 8, the gateway appliance 12 preferably further includes a custom cache 126 alongside

the category cache 125. The custom cache 126 records a customised list of categorisations. In preferred embodiments, the custom cache 126 is used to override other categorisations, or to add supplementary URLs. In  
5 the preferred embodiment, the custom cache 126 is structured identical to the category cache 125. Searches are preferably conducted in order through the custom cache 126, then if necessary the category cache 125, and finally if necessary by generating a request message 500 to the  
10 categorisation server 40. Preferably, the custom cache 126 does not perform any URL aging, so that a user has full control over the size and content of the custom cache 126. In this case, the previous and next pointers 827,828 are not required or are left unused.

15

In the preferred embodiment, the category cache 125 and/or the custom cache 126 can be cleared completely and then rebuilt with fresh data, such as after a reset operation. Preferably, each cache 125,126 may also be  
20 given a partial clear out, such as deleting all hosts 822 or pages 823 with a specified category code. The cache structure described with reference to Figures 8 and 9 enables convenient cache management, whilst being efficient to operate.

25

Figure 11 is a schematic view of the categorisation server 40 including a main module 410, a communication module 420, a protocol module 430 and an encryption module 440. The main module 410 initialises the categorisation  
30 service and creates worker threads. The communication module 420 receives and buffers data and responds to categorisation requests including generation of reply messages 600. The protocol module 430 unmarshals incoming

data into a comprehensible command format and marshals outgoing data into a transmittable format, and makes encryption/decryption calls to the encryption unit 440 where required. The encryption unit 440 encrypts and  
5 decrypts data, preferably according to the RC2 algorithm.

### **Licensing**

In a further aspect of the present invention, the  
10 categorisation service 400 running on the categorisation server 40 performs a licensing process.

In particular, it is desired to confirm that the request message 500 is valid and comes from a valid client  
15 device 10,12. This licensing process controls access to the categorisation service, such as for security and to enable paid-for subscription based implementations.

The licensing process employed in the preferred  
20 embodiments of the present invention is highly flexible and is readily integrated with other existing licensing mechanisms.

As shown above in Figure 6, the header 510 of each  
25 request message 500 preferably includes a licensing field 515 which carries data such as a licence key.

In the preferred embodiment, the licensing field 515 is subdivided into a partner ID field 516 and a client ID  
30 field 517. The partner ID field 516 allows a plurality of different licensing schemes to exist in parallel, each having different requirements or validation processes.

Referring again to Figure 11, the categorisation service 400 comprises a licensing module 450 associated with the main module 410, which performs validation of the supplied licensing field 515. In the preferred embodiment, 5 the licensing module 450 receives the licensing field 515 and returns a "licence valid" or "licence invalid" status which controls whether or not the categorisation server 40 will respond to a categorisation request message 500. Suitably, the licensing module 450 runs as a dynamically 10 linked library (DLL).

In a further preferred embodiment, the categorisation service 400 includes a plurality of licensing DLLs 450, one of which is called to validate the licensing field 515 15 according to the partner ID field 516. This allows different licensing schemes to be applied for different clients.

In the preferred embodiment, the partner ID field 516 20 is 4 bytes long, giving up to 65535 licensing partner identities. The client ID field 517 is suitably up to 60 printable characters long, allowing room for any appropriate secure licensing mechanism.

25 It is important to validate licenses relatively quickly, since the system is operating in real time and a user is waiting for their requested web page. As show in Figure 11, the categorisation server 40 preferably comprises a license cache 455 to store recently 30 encountered license fields 515. The licensing process comprises first checking whether the received licensing field 515 is stored in the licensing cache 455, and then calling the licensing validation DLL 450. Suitably, the

result of each licensing call is then added to the licensing cache 455 and is then available for subsequent requests from that client 12. Since clients tend to access the Internet in short burst of activity, it is likely that one categorisation request 500 will be followed by another soon after. The license cache 455 significantly improves response speed for second and subsequent requests.

Figure 12 is a schematic overview of the structure of the licensing cache 455. The structure is similar to that of the category cache 125 as discussed above with reference to Figure 9.

As shown in Figure 12, the licensing cache 455 comprises a hash array 1210 and one or more combined license trees and age list 1220. The hash array 1210 comprises index elements 1211 as a hash of license keys from the licensing field 515, each of which is a pointer to a licence tree list 1220.

20

Each tree node 1221 comprises a license string 1222 holding a license key and a corresponding license result (e.g. valid or invalid). The cache can hold solely valid keys, solely invalid keys, or, as in this example, a mixture of both, according to the circumstances of a particular implementation.

Further, each tree node 1221 comprises parent, left and right pointers 1223,1224,1225 defining the tree structure. This example shows a balanced red/black tree using a red/black flag 1228.

30

The license trees 1220 also functions as an age list to list each of the tree nodes 1221 by age. The age list comprises, within each tree node 1221, a next pointer 1226 and a previous pointer 1227 which refer to a next older tree node and a previous newer tree node, respectively.

Ideally, the license cache 455 is actively managed to reside within a predetermined memory size. Older tree nodes 1221 are deleted from a tail of the age list by referring to the next and previous pointers 1226,1227, whilst new nodes are added to the head of the age list. Optionally, the age list is updated after each access to keep recently accessed nodes at the head of the list.

In order to maintain valid content, the license cache is preferably flushed, in whole or in part, such as at scheduled regular timed intervals or following triggering events such as a reset.

Figure 13 shows example licensing schemes in more detail.

The categorisation service 400 makes calls to a license interface DLL 1350, which in turn makes calls one of a plurality of partner licence DLLs 1360.

The license interface DLL 1350 optionally includes the license cache 455. Preferably, the licence interface DLL first consults the licence cache 455 and then, if necessary, request licence validation by one of the partner licence DLLs 1360.

In this preferred embodiment, the license interface DLL 1350 resolves the partner ID field 516 by referring to a partner map database 1352, which links the partner ID 516 to a partner DLL name and preferably provides  
5 configuration information for making calls into that DLL.

In Figure 12, the partner licence DLLs 1360 include a no license DLL 1361 which simply indicates that any licence key is valid. This allows the system to run a  
10 default "no problem" licence mode prior to implementation of licence schemes which actively validate licence keys.

As one option, a no database DLL 1362 performs a mathematical, algorithmic or cryptographic validation of  
15 the licence key.

As another option, a hosted licensing DLL 1364 is provided which forwards licensing requests to a remote licensing server 1370 for validation. As examples, the  
20 licensing requests are sent over a local area network (LAN), or are forwarded using a SOAP-based web service over the Internet 20.

As yet another option, a database licensing DLL 1366  
25 connects directly into an ODBC database 1380 using a stored procedure to validate the licence key. The database 1380 suitably stores the partner ID field 516, licence code 517, and expiry date of valid licenses and hence can offer validation for a plurality of partner licence  
30 schemes. A licence management interface 1382 is provided to manage the content of the licence database 1380.



This aspect of the present invention has many advantages, as discussed above. Licensing is very useful in the context of controlling or monitoring Internet access by categorisation of URLs, and opens up many useful  
5 commercial and technical implementations of this technology. Further, the use of a licensing cache reduces time and resources for each validation and increases throughput. The cache is structured to be compact and is easily managed. The use of a partner ID field allows great  
10 flexibility and convenience to choose between available licensing schemes.

Although a few preferred embodiments have been shown and described, it will be appreciated by those skilled in  
15 the art that various changes and modifications might be made without departing from the scope of the invention, as defined in the appended claims.

Attention is directed to all papers and documents  
20 which are filed concurrently with or previous to this specification in connection with this application and which are open to public inspection with this specification, and the contents of all such papers and documents are incorporated herein by reference.

25

All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination,  
30 except combinations where at least some of such features and/or steps are mutually exclusive.

Each feature disclosed in this specification (including any accompanying claims, abstract and drawings) may be replaced by alternative features serving the same, equivalent or similar purpose, unless expressly stated  
5 otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.

The invention is not restricted to the details of the  
10 foregoing embodiment(s). The invention extends to any novel one, or any novel combination, of the features disclosed in this specification (including any accompanying claims, abstract and drawings), or to any novel one, or any novel combination, of the steps of any  
15 method or proc

**Claims**

1. A method of categorising Uniform Resource Locators  
5 (URLs) during Internet access, comprising the steps of:

receiving a URL request denoting a specified URL;

generating a request message (500) to request  
10 categorisation of the specified URL;

receiving a reply message (600) denoting a category  
for the specified URL amongst a predetermined set of  
categories;

15

adding the specified URL and the category to a  
category cache (125); and

in a second or subsequent instance of a URL request  
20 with respect to the specified URL, determining the  
category of the specified URL from the category cache  
(125).

2. The method of claim 1, wherein the category cache  
25 (125) comprises:

a hash array (810) wherein the host portion of the  
specified URL is hashed to produce an index element in the  
hash array (810); and

30

a combined host tree (820) and age list (820) wherein  
each tree node (821) of the host tree (820) comprises URL  
data (822) representing a specified URL and a

corresponding category, and each tree node (821) further comprises a next pointer (827) and a previous pointer (828) which refer to a next older tree node (821) and a previous newer tree node (821), respectively, to form the  
5 age list (820).

3. The method of claim 2, comprising adding new tree nodes to a head of the age list (820).

10 4. The method of claim 2 or 3, comprising removing an oldest tree node (821) from a tail of the age list (820) when the category cache (125) has reached a predetermined maximum size.

15 5. The method of claim 2, 3 or 4, comprising moving any host that is re-visited to a head of the age list (820).

6. The method of claim 2, 3, 4 or 5, wherein the hash  
20 array (810) has a predetermined length.

7. The method of claim 6, wherein the predetermined length of the hash array (810) is set by a variable which is input from a control panel during use.

25

8. The method of any of claims 2 to 7, comprising:

searching the category cache (125) for the specified URL; and

30

returning a category code for the specified URL from the category cache (125), or else generating the request

message (500) to request categorisation of the specified URL.

9. The method of any of claims 2 to 8, wherein each  
5 specified URL comprises a host portion (202) and a path  
portion (204), and each tree node (821) comprises a host  
string (822) holding the host portion (202), and a page  
array (823) of one or more page strings (823) each  
representing a path portion (204) depending from the host  
10 portion (202).

10. The method of claim 9, comprising:

extracting a URL host portion (202) and a URL path  
15 portion (204) from the specified URL, hashing the host  
portion (202) to determine an index element in the hash  
array (810), searching the host tree (820) identified by  
the index element to locate a tree node (821) matching the  
host portion (202), matching a URL path portion (204) with  
20 a page string in the page array (823), and returning a  
stored category associated with the specified URL.

11. The method of claim 9 or 10, wherein the category  
cache (125) comprises, within each tree node (821), a  
25 children flag (908) which identifies whether category  
codes are available for children of a specified URL host  
or URL path.

12. The method of any preceding claim, comprising  
30 providing a custom cache (126) alongside the category  
cache (125), wherein the custom cache (126) records a  
customised list of stored URLs and associated category  
codes.

13. The method of claim 12, wherein the custom cache (126) comprises:

5 a hash array (810) wherein the host portion (202) of a specified URL is hashed to produce an index element in the hash array (810); and

10 a host tree (820) wherein each tree node (821) of the host tree (820) comprises URL data (822) representing a specified URL and a corresponding category code.

14. The method of claim 12, comprising, in order, the steps of searching a specified URL in the custom cache (126) and returning a category code, or else searching the specified URL in the category cache (125) and returning a category code, or else generating a request message (500) to request categorisation of the specified URL and receiving a reply message (600) returning a category code.

20

15. A method for use in controlling or monitoring Internet access at a client device (12) by categorising Uniform Resource Locators (URLs), comprising the steps of:

25 receiving a specified URL;

searching a category cache (125) held at the client device (12) using the specified URL as a search key, and returning a category code associated with the specified URL when a match is found for the specified URL; and

30

generating a request message (500) to request a category code for the specified URL, when a match is not found for the specified URL.

5 16. The method of claim 15, comprising the steps of:

receiving a specified URL from a client software function within the client device (12); and

10 passing the category code to the client software function in return.

17. The method of claim 15 or 16, comprising:

15 sending the request message (500) to a remote categorisation service;

receiving a reply message (600) from the remote categorisation service denoting a category code for the  
20 specified URL amongst a predetermined set of categories.

18. The method of claim 17, further comprising:

adding the specified URL and the category code to the  
25 category cache (125).

19. The method of any of claims 15 to 18, wherein the method comprises:

30 hashing at least part of the specified URL to reference an index element for a hash array (810) within the category cache (125); and

using the index element as a pointer to a combined host tree (820) and age list (820) comprising one or more tree nodes, wherein each tree node (821) comprises URL data (822) representing a URL and one or more  
5 corresponding category codes, and each tree node (821) further comprises an age list next pointer (827) referring to an adjacent older tree node (821) and an age list previous pointer (828) referring to an adjacent newer tree node (821); and

10

searching the combined host tree (820) and age list for a tree node (821) containing URL data (822) which matches the specified URL.

15 20. The method of claim 19, wherein each tree node (821) comprises a host string (822) holding a host portion (202) of a URL and an associated category code, and a page array (823) of zero or more page strings (823) each page string holding a page portion and an associated category  
20 code, and the method comprises selecting a tree node (821) by a host portion (202) of a specified URL, and selecting a page string in the page array (823) matching all or part of a page portion of a specified URL, to thereby retrieve a category code for the specified URL.

25

21. The method of claim 20, wherein each page string further comprises a children flag (908) indicating whether category codes are available for children of a stored path portion (204).

30

22. The method of claim 19, 20 or 21, comprising the steps of determining an oldest tree node (821) at a tail of the age list (820) and removing the oldest tree node



(821) when the category cache (125) has reached a predetermined maximum size.

23. The method of claim 22, comprising the steps of  
5 adding a new tree node (821) and updating the age list (820) to place the new tree node (821) at a head of the age list (820).

24. The method of any of claims 19 to 23, comprising,  
10 when the searching step determines a tree node (821) matching the specified URL, updating to place the matched tree node (821) at the head of the age list (820).

25. A system for use in controlling or monitoring of  
15 Internet access by categorising Uniform Resource Locators (URLs), comprising:

a client device (12) arranged to monitor or control Internet access according to a category code of a  
20 specified URL, and including a categorisation module to provide the category code for the specified URL from a category cache (125) stored at the client device (12) or else generate a request message (500) to request categorisation of the specified URL; and

25

a categorisation server (40) coupled to communicate with the client device (12) and arranged to receive the request message (500) and to send a reply message (600) identifying a category code for the specified URL.

30

26. The system of claim 25, wherein the client device (12) and the categorisation server (40) communicate over the Internet.

27. The system of claim 25 or 26, wherein the client device (12) comprises a user machine (10) to provide a human-oriented user interface, and a client internet gateway appliance (12) to couple the user machine (10) to the Internet through an Internet Service Provider (ISP).

28. The system of claim 27, wherein the user machine (10) and the client internet gateway appliance (12) are formed within a single integrated device.

29. The system of claim 27, wherein the user machine (10) and the client internet gateway appliance (12) are separate devices arranged to communicate over a local network.

30. The system of any of claims 25 to 29, wherein the client device (12) and the categorisation sever are arranged to communicate over a local area network.

20

31. A client device (12), comprising:

an interface module (121) arranged to present a URL categorisation function, wherein the interface module (121) is arranged to receive a specified URL from a client software and to return a category code;

a category cache (125) holding a plurality of stored URLs and associated category codes, such that matching the specified URL against one of the stored URLs provides the category code; and

a communication module (122) arranged to send an outgoing request message (500) to a categorisation server (40) when there is no match in the category cache (125) and to receive and buffer incoming data including a corresponding reply message (600), wherein the request message (500) comprises the specified URL and the reply message (600) comprises the category code.

32. The client device (12) of claim 31, wherein the client device (12) is arranged to add the specified URL and the category code to the stored URLs and associated category codes in the category cache (125).

33. The client device (12) of claim 31 or 32, further comprising:

a custom cache (126) which holds a customised list of stored URLs and associated category codes, such that the custom cache (126) is searchable instead of or as well as the category cache (125).

34. A cache structure, comprising:

a hash array (810) comprising one or more index elements (811), each index element (811) comprising a host tree pointer and a hash key derived from a stored URL; and

one or more host trees (820) depending from the index elements (811) of the hash array (810), each host tree (820) comprising one or more tree nodes (821) each holding URL data (822) representing stored URLs and associated category codes; and

an age list (820) to list each of the tree nodes (821) by age, wherein the age list (820) comprises, within each tree node (821), a next pointer (827) and a previous pointer (828) which refer to a next older tree node (821) and a previous newer tree node (821), respectively.

35. The cache structure of claim 34, wherein each tree node (821) comprises a host string (822) holding a host portion (202) of a stored URL and an associated category code, and a page array (823) of zero or more page strings (823) each page string holding a page portion and an associated category code.

36. The cache structure of claim 35, wherein each page string comprises a children flag (908) denoting whether category codes are available for children of a stored URL path portion (204).

37. The cache structure of claim 34, 35 or 36, wherein each tree node (821) further comprises a parent node pointer, a left node pointer, and a right node pointer.

38. The cache structure of claim 36 or 37, wherein the tree list is a balanced red-black tree and each tree node (821) further comprises a red-black flag.

39. A program storage medium having computer executable instructions stored thereon for performing the method of any of claims 1 to 14.

30

40. A program storage medium having computer executable instructions stored thereon for performing the method of claims 15 to 24.

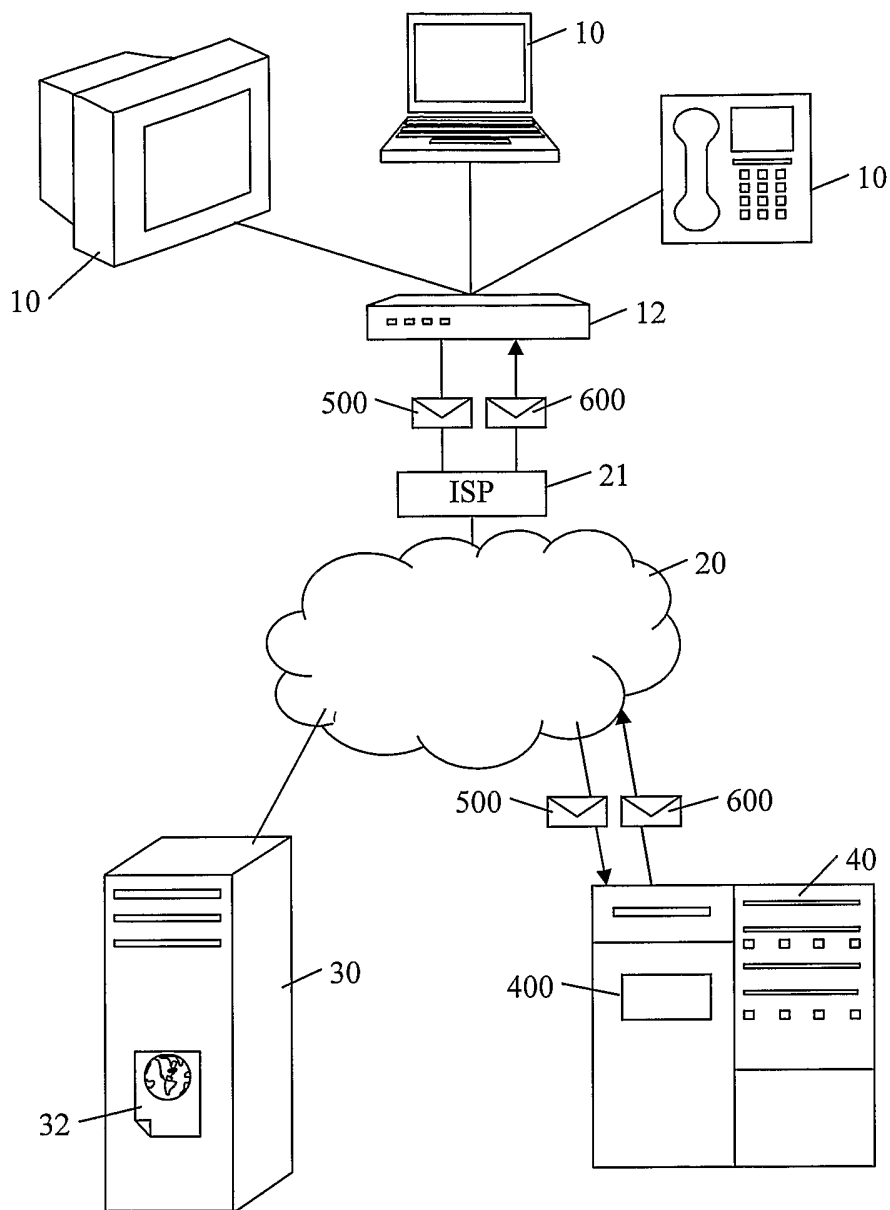


Fig. 1

2/12

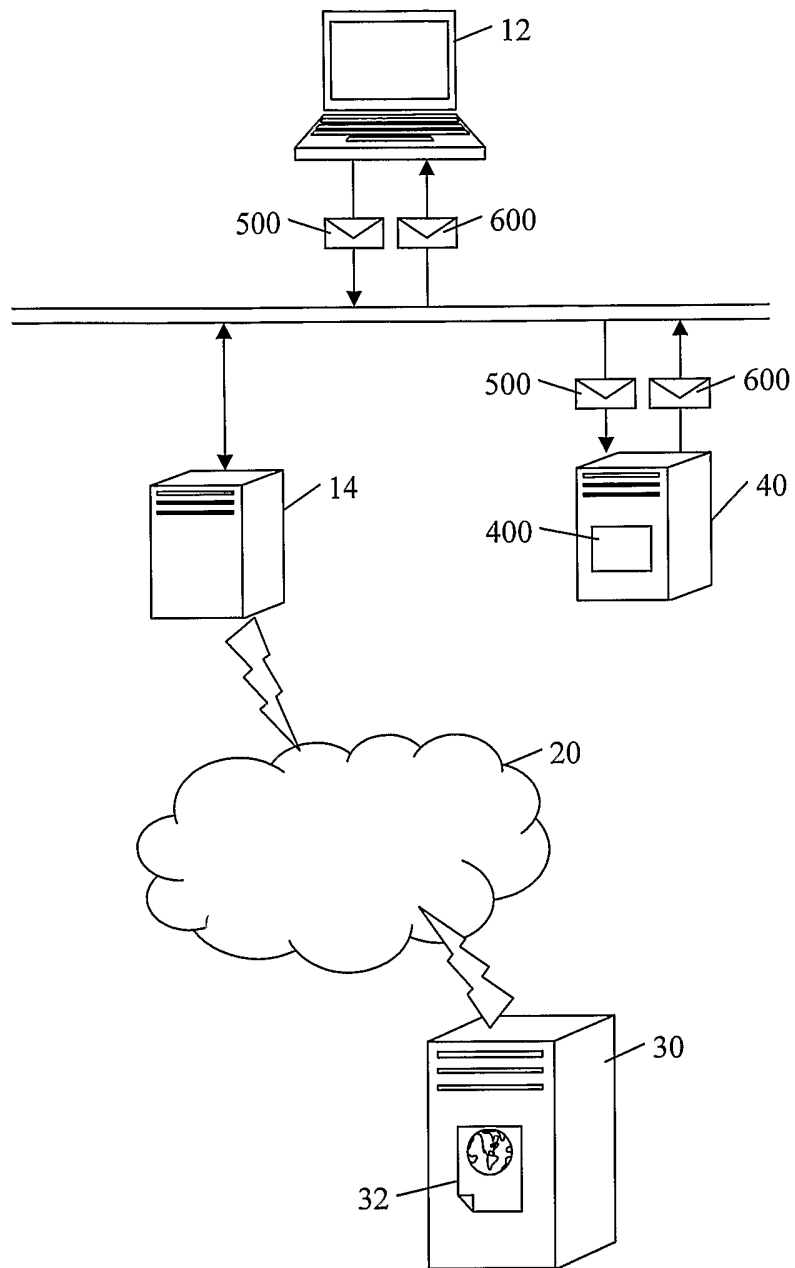
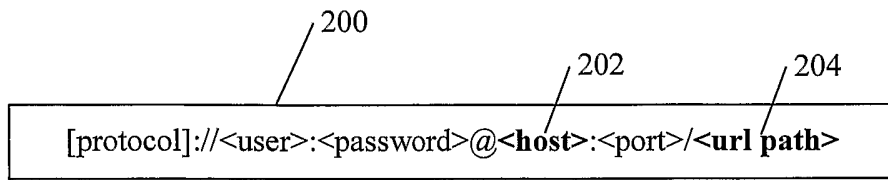
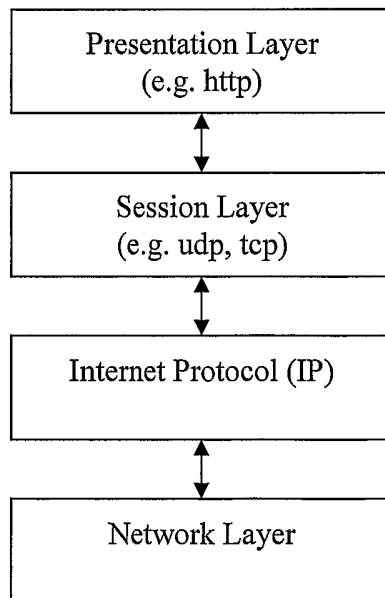


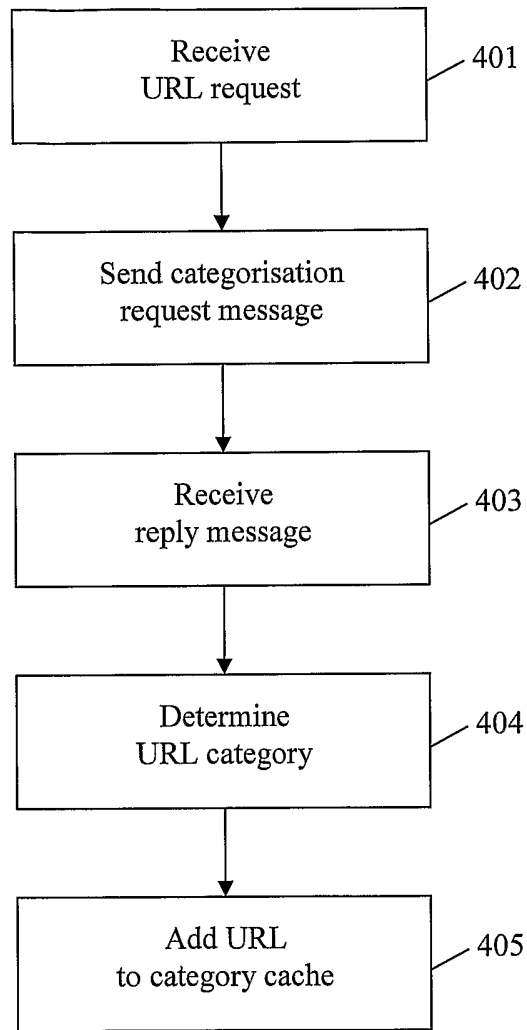
Fig. 2



**Fig. 3**



**Fig. 4**



**Fig. 5**



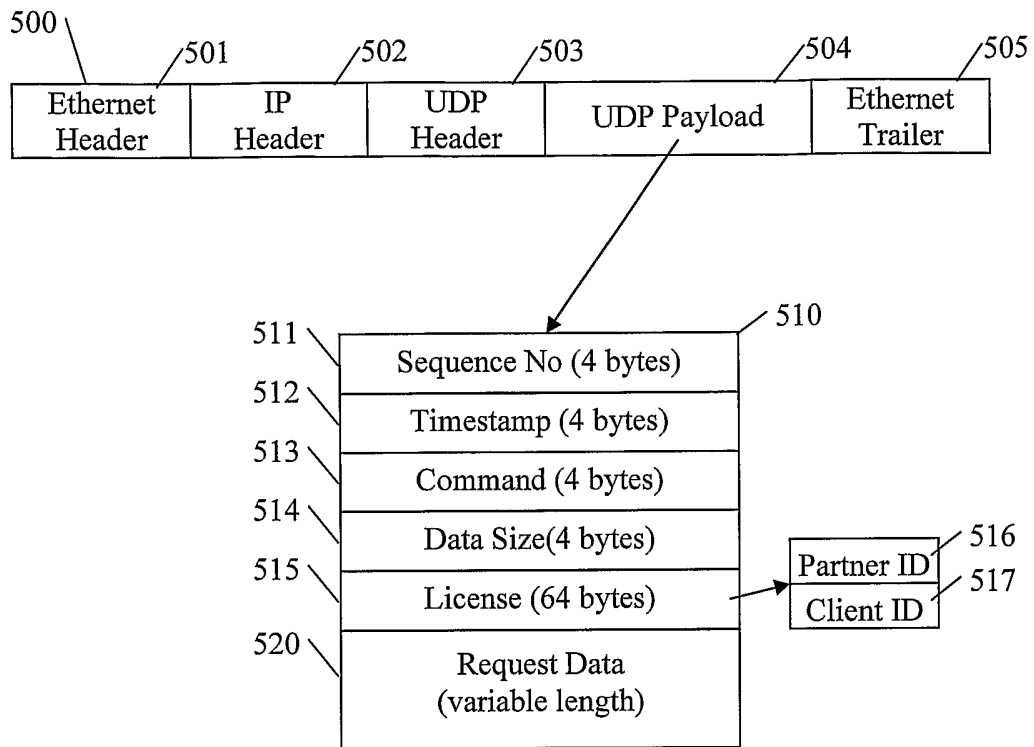


Fig. 6

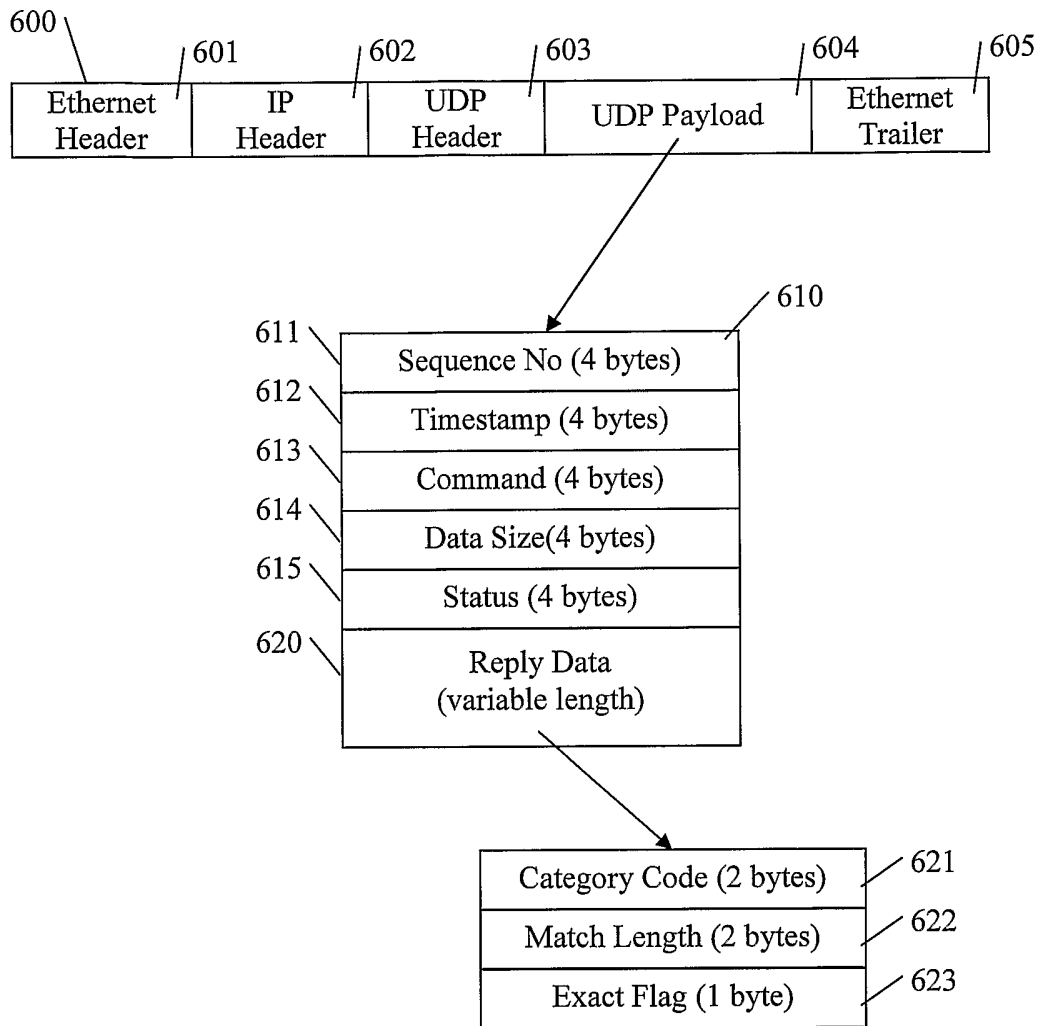


Fig. 7

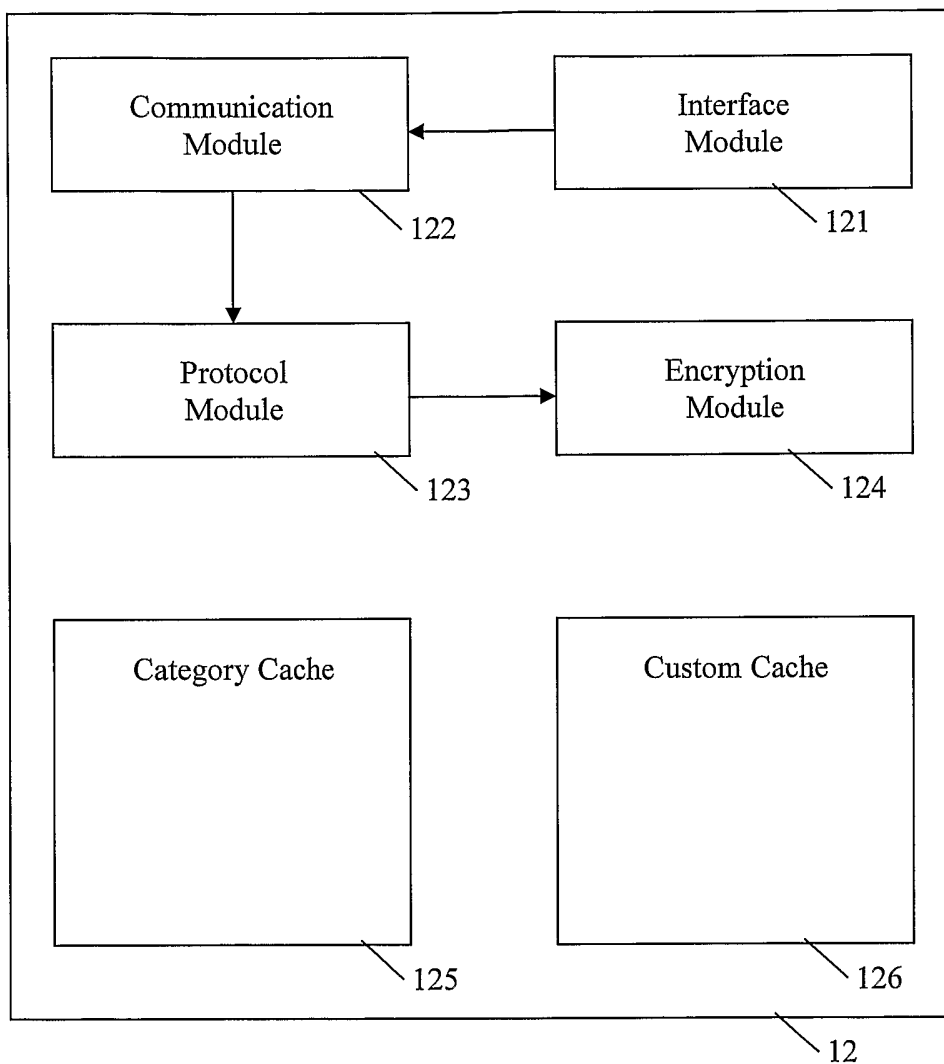


Fig. 8

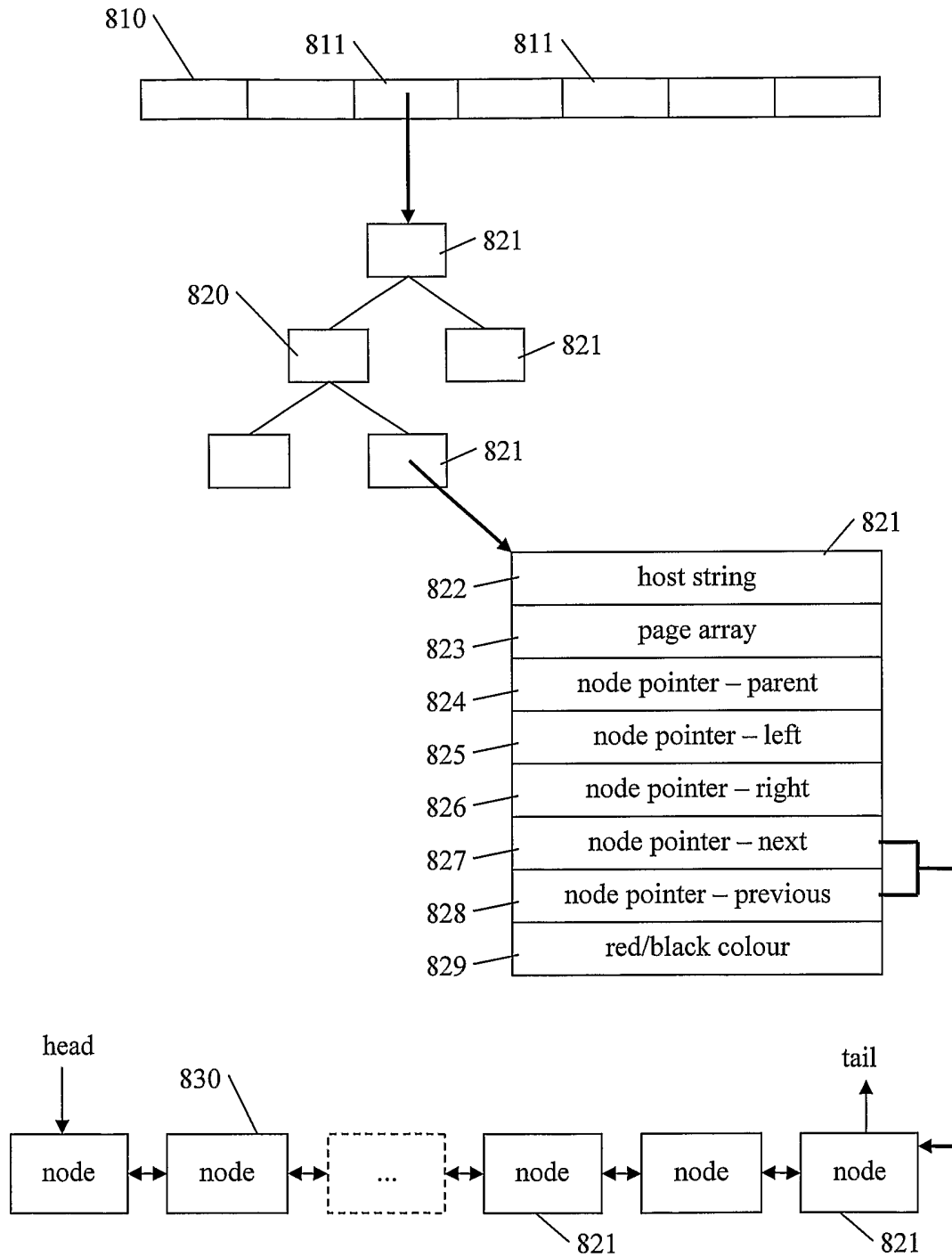


Fig. 9

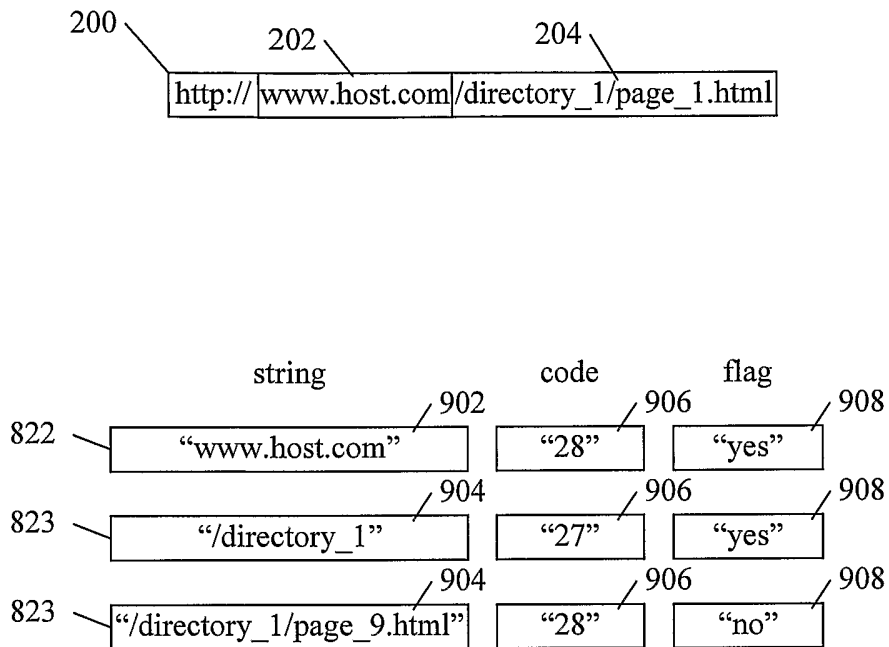


Fig. 10

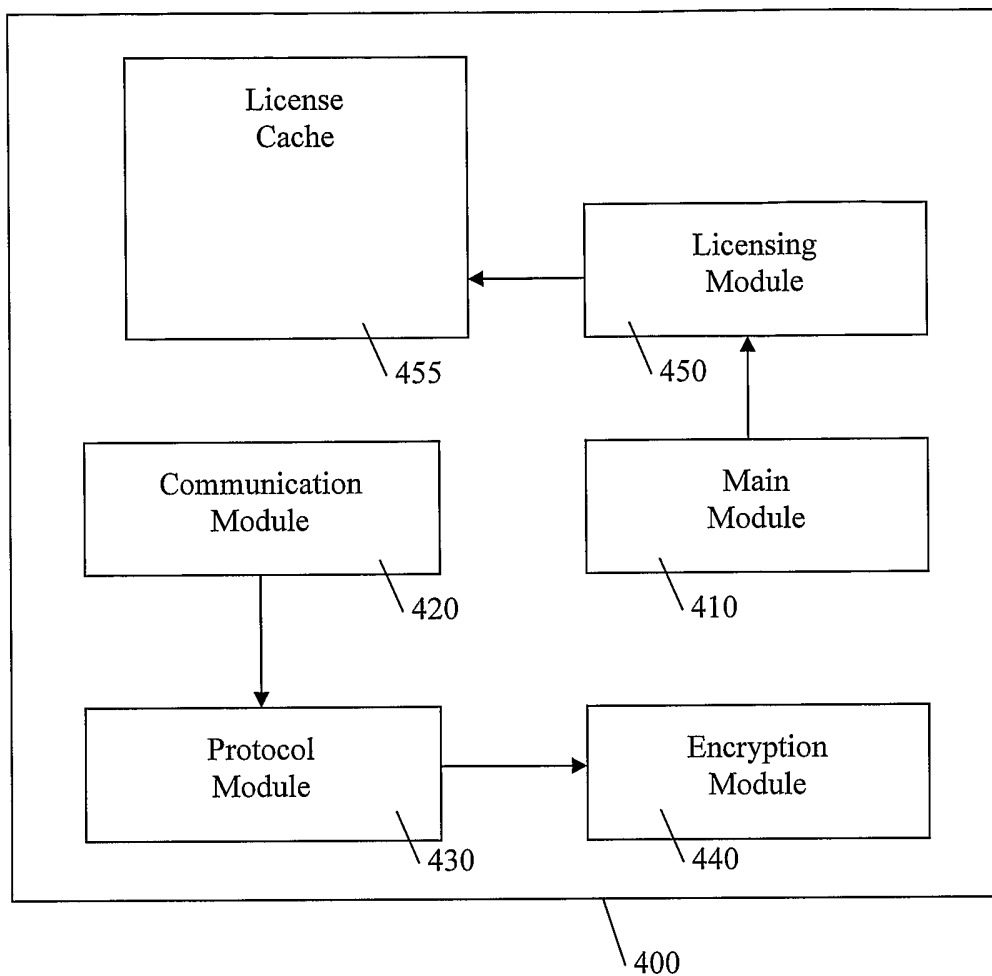


Fig. 11

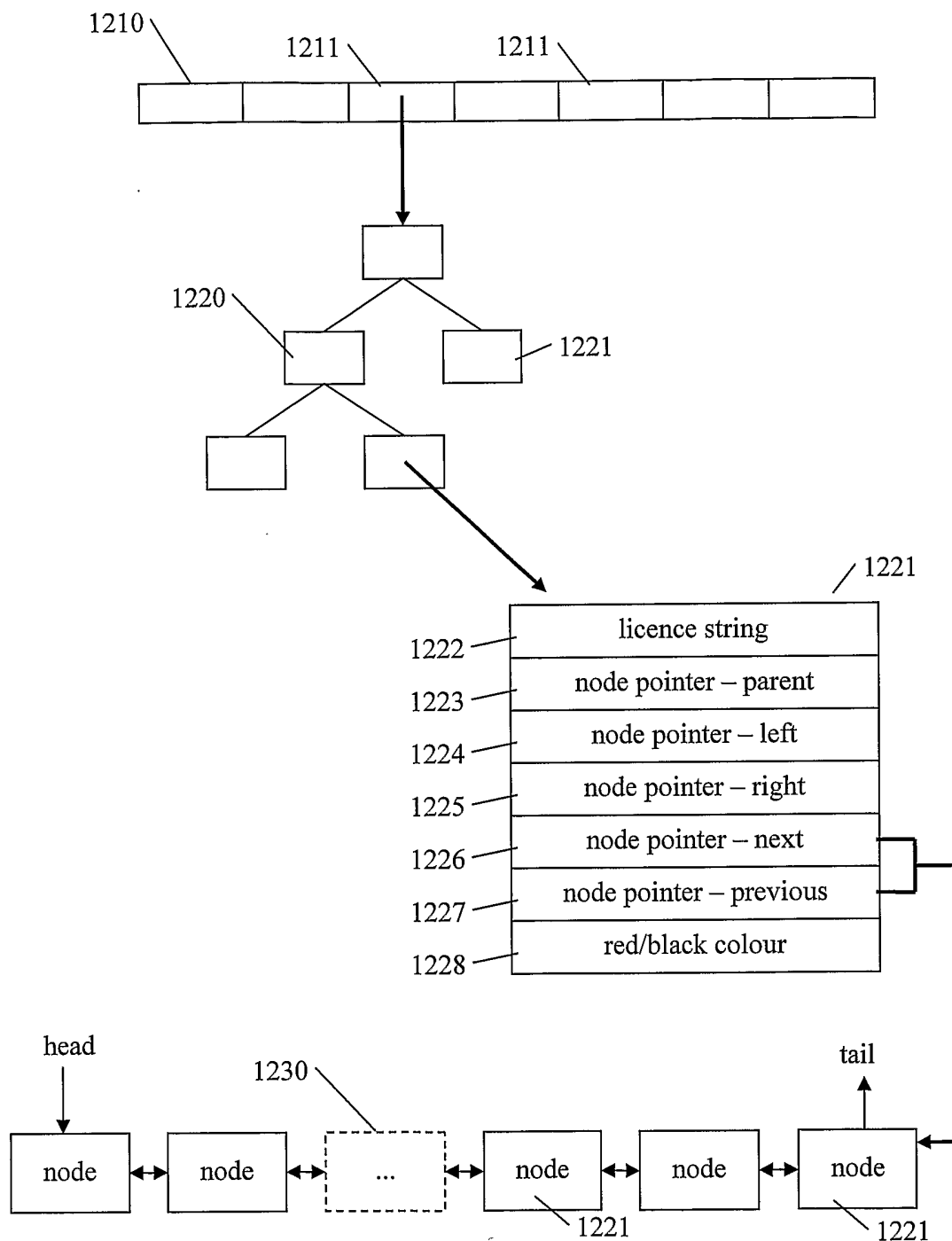


Fig. 12

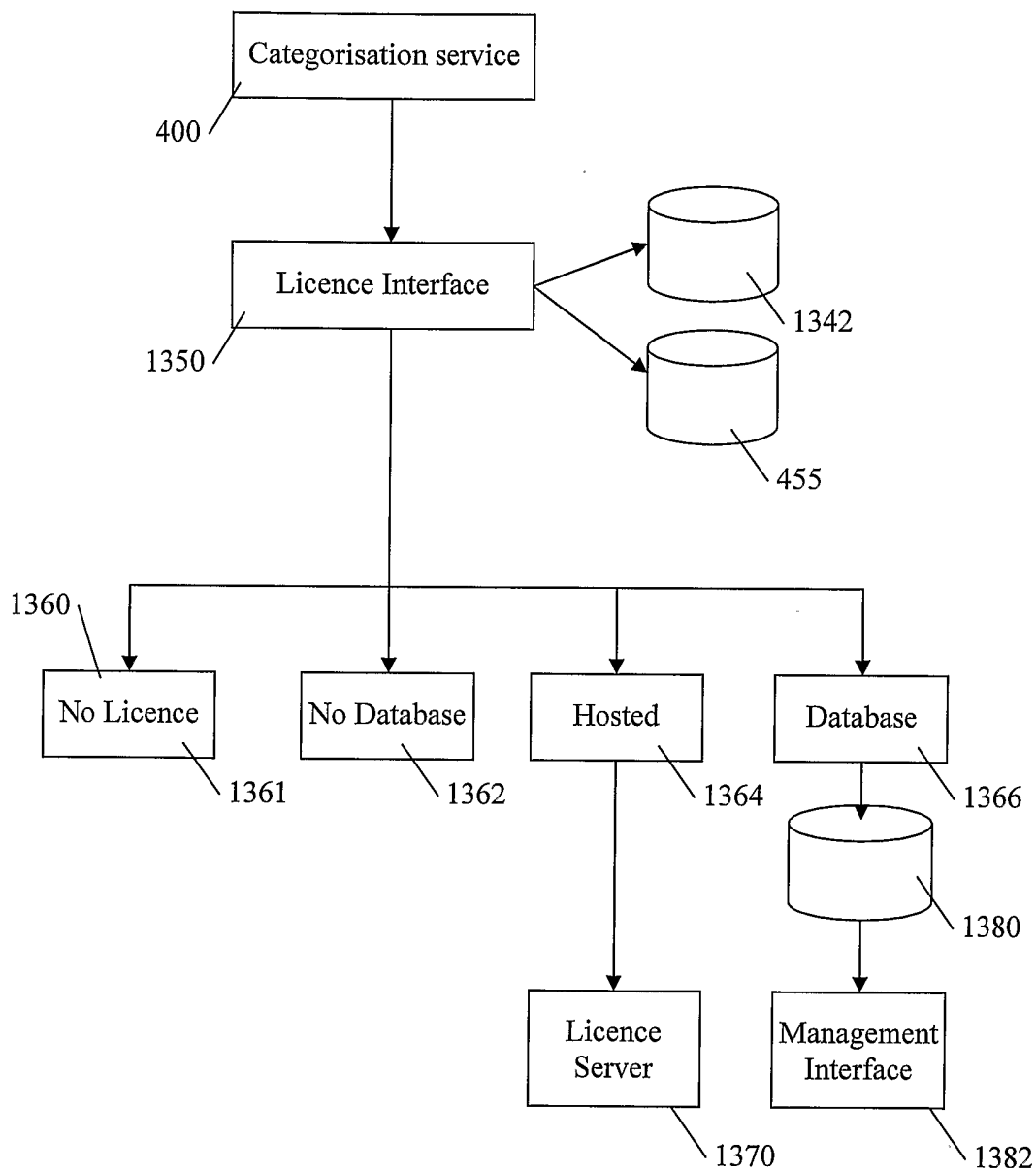


Fig. 13



# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/JP2005/003465

<b>A. CLASSIFICATION OF SUBJECT MATTER</b> H04L29/08				
According to International Patent Classification (IPC) or to both national classification and IPC				
<b>B. FIELDS SEARCHED</b>				
Minimum documentation searched (classification system followed by classification symbols) H04L				
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched				
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, WPI Data, INSPEC				
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>				
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
X	US 2004/006621 A1 (BELLINSON CRAIG ADAM ET AL) 8 January 2004 (2004-01-08) the whole document	1-40		
X	US 2002/120754 A1 (ANDERSON TODD J ET AL) 29 August 2002 (2002-08-29)  abstract paragraphs '0030! - '0033!, '0037! - '0047!, '0060!, '0061!, '0067!; figures 1a,3,4,10	1,8, 14-18, 25-33, 39,40		
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;"> <input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.                 </td> <td style="width: 50%; border: none;"> <input checked="" type="checkbox"/> Patent family members are listed in annex.                 </td> </tr> </table>			<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.	<input checked="" type="checkbox"/> Patent family members are listed in annex.
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.	<input checked="" type="checkbox"/> Patent family members are listed in annex.			
° Special categories of cited documents :				
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;">                     *A* document defining the general state of the art which is not considered to be of particular relevance                      *E* earlier document but published on or after the international filing date                      *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)                      *O* document referring to an oral disclosure, use, exhibition or other means                      *P* document published prior to the international filing date but later than the priority date claimed                 </td> <td style="width: 50%; border: none;">                     *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention                      *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone                      *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.                      *&amp;* document member of the same patent family                 </td> </tr> </table>			*A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *&* document member of the same patent family
*A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *&* document member of the same patent family			
Date of the actual completion of the international search  <p style="text-align: center; font-size: 1.2em;">30 November 2005</p>		Date of mailing of the international search report  <p style="text-align: center; font-size: 1.2em;">08/12/2005</p>		
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer  <p style="text-align: center; font-size: 1.2em;">Milano, M</p>		

# INTERNATIONAL SEARCH REPORT

Int'l Application No  
 PU. 2005/003465

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/105863 A1 (HEGLI RONALD BJORN ET AL) 5 June 2003 (2003-06-05)  paragraphs '0018!, '0025!, '0033!, '0051! - '0053!; figure 1 -----	1,8, 14-18, 25-33, 39,40
A	GREENFIELD P ET AL: "Access Prevention techniques for Internet Content Filtering" CSIRO, December 1999 (1999-12), XP002265027 the whole document -----	1-40

# INTERNATIONAL SEARCH REPORT

In                      al Application No  
 FC1/UB2005/003465

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004006621    A1	08-01-2004	AU 2003204349 A1	22-01-2004
		BR 0301972 A	31-08-2004
		CA 2429883 A1	27-12-2003
		CN 1470999 A	28-01-2004
		EP 1385097 A2	28-01-2004
		JP 2004030678 A	29-01-2004
		RU 2003119093 A	10-01-2005
US 2002120754    A1	29-08-2002	EP 1388091 A2	11-02-2004
		WO 02069114 A2	06-09-2002
US 2003105863    A1	05-06-2003	AU 2002351237 A1	17-06-2003
		CA 2465662 A1	12-06-2003
		EP 1461929 A2	29-09-2004
		JP 2005512211 T	28-04-2005
		WO 03049403 A2	12-06-2003