UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,880	10/09/2021	10469614	HOLA-003-US5-EPR	4776

131926 7590 02/24/2022

May Patents Ltd. c/o Dorit Shem-Tov
P.O.B 7230
Ramat-Gan, 5217102
ISRAEL

EXAMINER	
ESCALANTE, OVIDIO	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
02/24/2022	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

LISTON ABRAMSON LLP (GENERAL)
THE CHRYSLER BUILDING
405 LEXINGTON AVE., 46TH FLOOR
NEW YORK, NY 10174

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,880* .

PATENT UNDER REEXAMINATION *10469614* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Office Action in Ex Parte Reexamination	**Control No.** 90/014,880	**Patent Under Reexamination** 10469614	
	Examiner OVIDIO ESCALANTE	**Art Unit** 3992	**AIA (FITF) Status** Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

a. ☑ Responsive to the communication(s) filed on 13 February 2022.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

b. ☐ This action is made FINAL.

c. ☑ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire 2 month(s) from the mailing date of this letter. Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).** If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☐ Notice of References Cited by Examiner, PTO-892.
2. ☑ Information Disclosure Statement, PTO/SB/08.
3. ☐ Interview Summary, PTO-474.
4. ☐ _____.

Part II SUMMARY OF ACTION

1a. ☑ Claims 1-29 are subject to reexamination.
1b. ☐ Claims _____ are not subject to reexamination.
2. ☐ Claims _____ have been canceled in the present reexamination proceeding.
3. ☐ Claims _____ are patentable and/or confirmed.
4. ☑ Claims 1-29 are rejected.
5. ☐ Claims _____ are objected to.
6. ☐ The drawings, filed on _____ are acceptable.
7. ☐ The proposed drawing correction, filed on _____ has been (7a) ☐ approved (7b) ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
   1 ☐ been received.
   2 ☐ not been received.
   3 ☐ been filed in Application No. _____.
   4 ☐ been filed in reexamination Control No. _____.
   5 ☐ been received by the International Bureau in PCT application No. _____.
   * See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.
10. ☐ Other: _____

cc: Requester (if third party requester)

DETAILED ACTION

1. This action is in response to the Request for Reexamination filed on October 9, 2021 and the Examiner's Order filed on November 17, 2021 which granted *ex parte* reexamination of claims 1-29.

2. A Patent Owner's statement has not been received in response to the Examiner's Order Granting Reexamination.

Status of the Claims

3. Original claims 1-29 are rejected.

Proceedings

IPR2020-01506:

On September 4, 2020 a Petition for Inter Partes Review of US Patent 10,469,614 was filed.

On February 17, 2021, the Patent Trial and Appeal Board, under 35 U.S.C. §314(a) denied all grounds based on their assessment of the *Fintiv* factors.

Claim Construction

During reexamination ordered under 35 U.S.C. 304, and also during reexamination ordered under 35 U.S.C. 257, claims are given the broadest reasonable interpretation consistent with the specification and limitations in the specification are not read into the claims (*In re Yamamoto*, 740 F.2d 1569, 222 USPQ 934 (Fed. Cir. 1984)).

The Examiner notes that the following terms were addressed in the Request for Reexamination and addresses the BRI standard and the District Court's construction:

"client device"

The court construed "client device" as a device operating in the role of a client by requesting services, functionality, or resources from the server".

"first server"

The court interpreted "first server" to mean "a device that is operating in the role of a server by offering information resources, services, and/or applications and that is not the client device.

The Request states that a "first server" may be regarding as a device that has at least a role in which it operates as a server in the system in question, provided only that it is not the same device as the device serving as the client device in carrying out the claimed steps.

The Examiner acknowledges the construction of the above terms and does not currently offer a different interpretation than that set forth by the court and in the Request.

Information Disclosure Statement

With respect to the Information Disclosure Statement filed, February 13, 2022 and December 20, 2021, the information cited has been considered as described in the MPEP. Note that MPEP 2256 and 2656 indicate that the degree of consideration to be given to such information will be normally limited by the degree to which the party filing the information citation has explained the content and relevant of the information. Information that does not appear to be "patents or printed publications" as identified in 35 U.S.C. 301 or information that have not been dated have been considered to the same extent (unless otherwise noted), but have been lined through and will not be printed on any resulting reexamination certificate.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a)(1) the claimed invention was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention.

5. Claim(s) 1, 9 and 13 is/are rejected under 35 U.S.C. 102(a)(1) as being anticipated by Shribman US Patent Pub. 2011/0087733.

Regarding claim 1:

A method for use with a resource associated with a criterion in a client device that communicates with a first server over the Internet, the client device is identified in the Internet using a first identifier and is associated with first and second state according to a utilization of the resource, the method comprising:

As shown in Fig. 3 of Shribman, agent 122 (client device) communicates with client 102/acceleration server 162 (first server) over the Internet. The client device is identified in the Internet by an IP address (first identifier) ¶[0072]. The availability state of the client device is reflected in a database according to responses from "keep-alive" signals. ¶¶ [0069] (connection status), [0099] (keep-alive monitoring).

The Examiner notes that in paragraph [0035], Shribman discloses that each communication device (in e.g. Figure 3) may serve as a client, peer, or agent, depending upon requirements of the network 100.

initiating, by the client device, communication with the first server over the Internet in response to connecting to the Internet, the communication comprises sending, by the client device, the first identifier to the first server over the Internet;

Upon its own startup (and thus "in response to connecting to the Internet"), the agent 122 (client device), acting as a client, registers with acceleration server 162, sending its IP address (first identifier):

> [0072] The initializer 222 is the first element of the communication device 200 to operate as the communication device 200 starts up (block 302). As the initializer 222 starts, it first communicates with the acceleration server 162 to sign up with the acceleration server 162. This is performed by providing the acceleration server 162 with the hostname, and all IP addresses and media access control (MAC) addresses of the interfaces on the communication device 200 having the initializer 222 thereon.

This combination of events also constitutes initiating by the client device of communication with the first server over the Internet comprising sending the first identifier to the first server. When a new client device goes online, the acceleration server shares a list of the origin IP addresses it will be handling with previously registered devices ¶ [0066], which a POSITA would understand as also identifying (by

the IP address, which is the only available identifier) the device responsible for handling the new list being communicated.

Additionally, the client device's IP address is also sent by the acceleration server to client 102 (the first server) ¶[[0080], and the IP address of the client device is reflected in the first server's local "cache" database, see item 310 in Fig. 7. The client device thus initiates these communications, in response to connecting to the Internet, in which it sends its IP address to the first server. In either operation, the client device IP address is relayed through the acceleration server, but thereby "to" the first server as well. The net result in each case being of a process that begins on startup of the first client device, where it sends its IP address, by way of the acceleration server, to another client device that corresponds to the claimed "first server."

when connected to the Internet, periodically or continuously determining whether the resource utilization satisfies the criterion;

Shribman discloses periodically determining whether agent devices are online. Shribman discloses:

> [0099] ...As shown by block 552, the acceleration server sends "keep alive" signals to the network elements, and keeps track within its database as to which network elements are online.

See also 166 in Fig. 7. Block 552 in Fig. 13 (which the full figure shows is repeatedly executed in a loop) reflects this as well. The referenced "database" makes the disclosed status determination persistent, as reflected in items 166 and 312 in Fig. 7. While item 166 in Fig. 7 reflects a record of the claimed determination in the database of the acceleration server, item 312 reflects a corresponding record in each communication device database as well, indicating that the communication devices perform corresponding monitoring of each other.

The Examiner notes that "keep-alive" signals are used, to "keep[] track of... which network elements are online" for example, by determining that they are off-line when the keep-alive sender fails to receive a responsive to its keep-alive signal after a specified timeout period. Thus, the "resource" may be viewed as the sum total of the capabilities of the device in its current operating condition to respond to a

signal. It is noted that a reasonable interpretation that a "resource" in the context of claim 1 can mean a combination of device facilities whose aggregate utilization at any given time affects the time it takes for the device to respond to a keep-alive signal.

The "criterion" in this context may be viewed as whether the device responds within a specified time period. The "first state" corresponds to the state of the disclosed database, in reflecting the device as being on-line or off-line as so determined.

responsive to the determining that the utilization of the resource satisfies the criterion, shifting to the first state or staying in the first state;

If the utilization of the resource satisfies the criterion, Shribman discloses marking the device on-line in the database (or continue such a marking if the database was already in this state), which constitutes shifting to (or staying in) the first state. This is based on the reasoning that having one of two mutually exclusive entries (on-line or off-line) written in a database represents a "state" of the system, and that changing such entries from one such indication (e.g., on-line) to the other (off-line) constitutes a "shitting" of the state.

Shribman discloses shifting to (or staying in) a first state based on whether the utilization of a resource (i.e., the device's collected capacity to respond to a keep-alive message satisfies a timing criterion). It is noted that the "state" is manifested in a database entry, which resides on the device doing the monitoring, which can be the acceleration server 162 or any of the communication devices 200 (see ¶ [0069], describing each communication device's "list of peers" and "the connection status 312 of the peer.")

responsive to the determining that the utilization of the resource does not satisfy the criterion, shifting to the second state or staying in the second state;

If the utilization of the resource does not satisfy the criterion (i.e., a keep-alive signal is sent to the device, but timely response is received), Shribman teaches to mark the device as off-line in the database (or continue such a marking if the database was already in this state), which constitutes shifting to for staying in) the second state.

Shribman discloses, it can be the acceleration server and/or any of the communications devices (which can serve as agents, peers, or clients) that shifts state as claimed, either of which is consistent with the claim language.

responsive to being in the first state, receiving, by the client device, a request from the first server

Shribman sends content retrieval tasks only to those agents marked as online: "choosing one or more client communication device out of all online client communication devices" (claim 32 in Shribman). See also ¶¶ 38, 66, 69, 80 (The list of agents is created by the acceleration server 162 by finding the communication devices of the communication network 100 that are currently online...), and ¶99.

Thus, the client device in Shribman 'receives the content retrieval request (the request) from the requesting client (first server) only responsive to being in the "first state," i.e., the online state.

In particular, Shribman 'discloses that the acceleration server 162 sends a list of agents to client 200 (first server) based in part "by finding the communication devices of the communication network 100 that are currently online" (¶ [0080], i.e., responsive to being in the first state, and that as a consequence, the agents (each a "client device"), receives a request from the client (first server):

> [0081] As shown by block 360, the client module 224 then sends the original request (e.g.. "GET http: / /www.aol.com/index.html HTTP/1.1") to all the agents in the list received from the acceleration server 162 in order to find out which of the agents in the list is best suited to be the one agent that will assist with this request.

However, as the foregoing reflects is also disclosed, only agents found to be in an on-line state are sent the request. Thus, Shribman discloses that, responsive to being in the first state, the client device receives a request from the first server. This occurs at block 360 in Fig. 9.

performing a task, by the client device, in response to the receiving of the request from the first server

As noted above, the agent/client device receives "the request" from the first server at block 360 of Fig. 9, as described at ¶ [0081]. Responsive thereto, the agent/client device performs a task, which begins

at block 382 on Fig. 10, and continuing through the remainder of Fig. 10 and from there through Fig. 11, as described in ¶¶ [0082]-[0089].

wherein the method is further configured for fetching over the Internet a first content identified by a first content identifier from a web server that is distinct from the first server

The "task" to be performed by the client device is proxied web content retrieval. Referring to block 390 and following in Fig. 10, Shribman discloses two possible responses by an agent to a content request as discussed above: the first response (blocks 392-596) is if the agent happens to already have information as to how the request may be satisfied locally (from cache), and the second (and alternate) response (blocks 400-406) is if it does not—in which case it fetches the content directly from the origin web server (i.e., "fetching over the Internet a first content identified by a first content identifier from a web server that is distinct from the first server"):

> [0089] If the selected agent discovers that it does not have information about this request, or if the selected agent discovers that the information it has is no longer valid, the selected agent needs to load the information directly from the server in order to be able to provide an answer to the requesting client. As shown by block 400, the selected agent then sends the request directly to the server. The selected agent then stores the information it receives from the server (both the headers of the request, as well as chunks of the response itself) in its database, for this particular response to the client, as well as for future use to other clients that may request this data (block 402}. The selected agent then prepares a response (list) for the client, where the response includes the protocol headers (if these are the first five chunks), and the checksums of the five chunks, and provides itself as the only peer for these chunks (block 404}. This list is then sent back to the client (block 406).

The first content—the information that the agent in this scenario "load[s] directly from the server," is identified by a first content identifier: The request includes the URL as shown in ¶ [0081] and thus the first content identifier. The "server" mentioned is one that responds to HTTP requests and thus is a web server (see web server 152 in Fig. 3), and is also distinct from the first server.

receiving, by the client device, the first content identifier from the first server;

The first content identifier, as shown in ¶ [0081] of Shribman (referring to block 360 in Fig. 9), is sent by the client device (client 102, first server) to each agent (each a "client device" for purposes of

claim 1). Thus, Shribman discloses the client device receiving the first content identifier from the first server, as part of the task that the client device performs responsive to being in the first state.

sending, by the client device, the first content identifier to the web server:

Shribman at [0089] ("As shown by block 400, the selected agent then sends the request directly to the server").

receiving, by the client device, the first content from the web server in response to the sending of the first content identifier; and

Shribman at ¶ [0089] discloses ("The selected agent [client device] then stores the information it receives from the server (both the headers of the request, as well as chunks of the response itself) [first content] in its database, for this particular response to the client [first server]".

sending, by the client device, the received first content to the first server.

Shribman discloses in ¶ [0089] that in the scenario being described, the agent (client device) "provides itself as the only peer for these chunks [first content]" and that "[t]his list is then sent back to the client [first server]." Since the agent is the only local source for the requested content, It is all sent from the client device (the agent) to the first server (client 102) when the client requests these chunks from (in this case) the same agent, and the agent send them, per ¶ [0090]-[0092].

Regarding claim 9:

The method according to claim 1, wherein the client device is further storing, operating, or using, a client operating system.

Shribman ¶ [0045], discloses that each client device further stores, operates, and uses, a client operating system.

Regarding claim 13:

The method according to claim 1, further comprising executing an application, and wherein the application comprises a web browser.

Shribman teaches that the client device may run a web browser. ¶ ¶ [0045], [0046], [6057], [0058], [0078], and [0095].

Claim Rejections - 35 USC § 103

6. In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

The following is a quotation of 35 U.S.C. 103 which forms the basis for all obviousness rejections set forth in this Office action:

> A patent for a claimed invention may not be obtained, notwithstanding that the claimed invention is not identically disclosed as set forth in section 102, if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art to which the claimed invention pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries for establishing a background for determining obviousness under 35 U.S.C. 103 are summarized as follows:

1. Determining the scope and contents of the prior art.

2. Ascertaining the differences between the prior art and the claims at issue.

3. Resolving the level of ordinary skill in the pertinent art.

4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

7. Claims 2, 3, 15-19, and 21-27 is/are rejected under 35 U.S.C. 103 as being unpatentable over Shribman in view of Sigurdson US Patent Pub. 2007/0174246.

Regarding claim 2:

The method according to claim 1, wherein the determining is performed by the client device.

As set forth above, Shribman discloses that the "state" in the determining step, is manifested in a database entry, which resides on the device doing the monitoring, which can be the acceleration server

162 or any of the communication devices 200 (see ¶ [0069], describing each communication device's "list of peers" and "the connection status 312 of the peer."). Thus, the determining can be performed by the client device.

In addition, Sigurdsson teaches wherein the determining is performed by the client device. See paragraph [0061], lines 7-8, "monitoring polling by the client".

A POSITA would have been motivated before the effective filling date of the claimed invention to make the obvious a simple modification to incorporate the further features of Sigurdsson, of having the claimed determining being performed by the client device, into the framework of Shribman to obtain a more versatile tracking of the client device's state and availability.

Regarding claim 3:

The method according to claim 1, further comprising periodically or continuously sending, by the client device, the resource utilization to the first server, and wherein the determining is performed by the first server in response to a receiving, by the first server, the resource utilization from the client device.

Shribman teaches tracking the state of the agent device by another device, which sends keep-alive messages to the agent (periodically or continuously sending), determines if the resources (reflecting resource utilization) are timely (determining), and tracks the results in a database. ¶ [0099].

The Examiner notes that to the extent that it is considered that updating a database does not necessarily entail that the client device periodically or continuous sends the resource utilization to the first server, the Examiner notes that it would have been obvious to one of ordinary skill in the art to perform this function.

Sigurdsson teaches periodically or continuously sending, by the client device, the resource utilization to the server (see [0006], lines 8-16, which discloses a client transmitting content-related metadata to the server, and wherein the determining is performed by the server in response to a receiving, by the server, the resource utilization from the client device (see [0007], lines 1-8, which discloses

determining that a server is configured to transmit content to a client in response to receiving metadata from a client)."

A POSITA would have been motivated before the effective filling date of the claimed invention to make a simple modification to incorporate the further features of Sigurdsson, of periodically sending performance metadata to the server so that the server could determine client device availability based thereon, into the framework of Shribman to obtain a more versatile tracking of the client device's state and availability.

Regarding claims 15 and 16:

The method according to claim 1, wherein the client device comprises, or consists of, a portable or mobile device.

The method according to claim 15, wherein the mobile device comprises a smartphone.

Shribman, as set forth above, in claim 1, does not specifically discloses of a portable or a mobile device.

Nonetheless, Sigurdsson teaches that it was known for client device to comprise, or consist of, a portable or mobile device (fig. 4). In addition, Sigurdsson et al teaches wherein the mobile device comprises a smartphone (fig. 4).

Thus, it would have been obvious before the effective filing date of the claimed invention to do likewise, given that Shribman expressly discloses of various client type of devices such as computers. The Examiner notes that using a mobile device such as a smartphone would have yielded a predictable result to one of ordinary skill in the art since mobile phones such as described in Sigurdsson are configured to access the internet and communicate with servers.

Regarding claim 17:

The method according to claim 1, for use with a set threshold value, and wherein the criterion is satisfied when the resource utilization is above or below the threshold.

As discussed, a timeout value is used with a keep-alive signal as disclosed in Shribman. The timeout value acts as a threshold value. If it is considered that the timeout value is not a threshold value, the examiner notes that it would have been obvious to use a threshold value.

For example, Sigurdsson et al teaches for use with a set threshold value, and wherein the criterion is satisfied when the resource utilization is above or below the threshold (¶ [0008], lines 6-11 & [0012]), which disclose transmitting data between clients only in the event that a peer client doesn't have a bandwidth that exceeds a predetermined threshold).". Adding this feature would have been obvious toe one of ordinary skill in the art before the effective filing date given that a timeout threshold value for responses from the client was already implicit in the server's use of keep-alive messages, as disclosed in the Shribman '733 publication.

Regarding claim 18:

The method according to claim 1, wherein the resource comprises, or consists of, a hardware component in the client device.

While Shribman teaches a resource comprising the collected facilities of a device to respond to a message, Shribman does not specifically disclose the resource comprises a hardware component.

Nonetheless, Sigurdsson et al teaches wherein the resource comprises, or consists of, a hardware component in the client device (fig. 1 & ¶ [0060], Lines 1-3, which disclose determining available bandwidth on each client '102 & '104)."

Thus, it would have been obvious to one of ordinary skill in the art before the effective filing date to use discrete resources, such as available bandwidth, as disclosed in Sigurdsson, for the same purpose as set forth by Shribman. The Examiner notes that Shribman already discloses of using hardware components such as described in Figure 4 and thus, using hardware components for resources would have been predictable to one of ordinary skill in the art.

Regarding claim 19:

The method according to claim 18, wherein the hardware component comprises, or consists of, a processor or Central Processing Unit (CPU) operation in the client device.

Sigurdsson et al teaches wherein the hardware component comprises, or consists of, a processor or Central Processing Unit (CPU) operation in the client device (fig. 9)."

As in claim 18, the use of any single hardware resource whose usage would affect the ability to respond to a message or perform a task, as in the case of the use of CPU utilization for this purpose in Sigurdsson, would have been an obvious modification to Shribman. See also Figure 4 of Shribman.

As set forth above, it would have been obvious to one of ordinary skill in the art to use hardware components.

Regarding claim 21:

The method according to claim 18, wherein the hardware component comprises, or consists of, a memory in the client device, and wherein the resource utilization is based on, or comprises, an amount of used or unused location or space of the memory.

Sigurdsson et al teaches wherein the hardware component comprises, or consists of, a memory in the client device (¶ [0105], lines 1-2), and wherein the resource utilization is based on, or comprises, an amount of used or unused location or space of the memory (see [0008], lines 7-12).

For the reasons set forth above regarding claim 19 (CPU utilization}, it would have been equally obvious to use memory utilization, as also taught in Sigurdsson, as indicative of the client device's ability to respond to a message or perform a task.

As set forth above, it would have been obvious to one of ordinary skill in the art to use hardware components.

Regarding claim 22:

The method according to claim 1, wherein the resource comprises, or consists of, input or output capability.

Shribman, as applied above to claim 1, does not specifically disclose the resource comprises input or output capability.

Nonetheless, Sigurdsson et al teaches wherein the resource comprises, or consists of input or output capability see ¶ [0051], lines 10-11, which discloses available resource capacities at each client device.

Thus, it would have been obvious to a person of ordinary skill in the art before the effective filing date to understand that a resource such as available resource capacities can include input or output capabilities as described by Sigurdsson. As explained by Sigurdsson, these capability allows the resource capacities to the utilized the client device.

Regarding claim 23:

The method according to claim 22, wherein the resource comprises, or consists of, communication bandwidth of communication with another device over the Internet.

Sigurdsson et al teaches wherein the resource comprises, or consists of communication bandwidth of communication with another device over the Internet (see ¶ [0051], Line 11).

As with, e.g. claims 18 and 22, keeping track of communication bandwidth, as disclosed in Sigurdsson, goes to the same considerations of the ability of the client to respond to a message or perform a task, and would have been an obvious adaptation of what is already disclosed by Shribman.

Regarding claim 24:

The method according to claim 23, wherein the resource comprises, or consists of, communication bandwidth of communication with the first server over the Internet, or wherein the resource utilization is based on, or according to, IETF RFC 2914.

Sigurdsson et al teaches 'wherein the resource comprises, or consists of, communication bandwidth of communication with the server over the Internet (see ¶ [0079], lines 7-10), or wherein the resource utilization is based on, or according to IETF RFC 2914."

As with claim 23, concerning monitoring a resource that is communication bandwidth with "another device," it is a small and obvious step to focus on the situation where the other device is the "first server" (as taught in Sigurdsson), especially where, as here, (per claim 1) it is the first server that is parcelling out the content retrieval tasks.

Regarding claim 25:

The method according to claim 1, further comprising periodically sending, by the client device, a message that comprises a status of the client device, or is in response to the status of the client device.

Sigurdsson et al teaches periodically sending, by the client device, a message that comprises a status of the client device, or is in response to the status of the client device (see ¶ [0061], lines 6-8, which discloses determining online or offline status of each client, by each client)."

In par [0061], lines 6-8 of Sigurdsson, which discloses determining online or offline status of each client, by each client is obvious in light of sending. by the client device, a message that comprises a status of the client device because a status can be drawn to a plurality of client devices

Thus, it would have been obvious before the effective filling date of the claimed invention, in order for the acceleration server and client devices of that publication including the client device that acts as the first server), to follow the implementation of Sigurdsson, generating its database entries as to the status of client devices (e.g., entry 312 in Fig. 7 of Shribman), by receiving and recording responses to the disclosed keep alive messages, which would have been sent periodically in response to the periodically sent keep-alive messages to each client device, when the client device was online,

In view of Sigurdsson, the foregoing would have been a very obvious way to implement the on-line-off-line client device state tracking already disclosed by Shribman.

Regarding claim 26:

The method according to claim 25, wherein the status is associated with being in the first or second state, and wherein the message is sent over the Internet to the first server.

The obviousness showing above with regard to claim 25 applies to claim 26 as well. Claim 26 additionally specifies only that the status conveyed by the message from the client device is associated with being in the first or second state, and that the message is sent over the Internet to the first server. The disclosures relied upon above with regard to claim 25 already cover these additional limitations. When the client responds to the keep-alive message it sends a message associated with being in the first (online) state.

The message is sent over the Internet, as the client devices of Shribman are deployed in the Internet. See, e.g., ¶[0002], [0058]. The message is received by the first server, because the first server is also a client device, and every client device according to Shribman disclosure has a database 282 which notes the connection status 312 of other client devices with which it deals, which results from such messages (or which it would be obvious to implement in this manner in view of the disclosures of Sigurdsson cited in connection with claim 25).

Regarding claim 27:

The method according to claim 25, wherein the message comprises, or is based on, an 'heartbeat' message, and wherein the time period between multiple messages sent is at least 10 milliseconds, 20 milliseconds, 30 milliseconds, 50 milliseconds, 100 milliseconds, 1 second, 2 seconds, 3 seconds, 5 seconds, 10 seconds, 20 seconds, 30 seconds, 50 seconds, or 100 seconds, 1 minute, 2 minutes, 3 minutes,

minutes 5, or 10 minutes.

Sigurdsson teaches wherein the message comprises, or is based on, an "heartbeat" message (¶ [0061], lines 11-13), and wherein the time period between multiple messages sent is at least 10 milliseconds, 20 milliseconds, 30 milliseconds, 50 seconds, 30 seconds, 50 seconds, or 100 seconds, 1

minute, 2 minutes (¶ [0045], lines 14-15}, which discloses implementing set intervals for posting of client content of 3 minutes, 5 minutes, or 10 minutes. Setting a specified time period between multiple keep-alilve messages, to the extent not inherent in Shribman, would have been an obvious implementation to incorporate into the techniques taught by Shribman in view of the cited disclosures of Sigurdsson.

The frequency of such messages, including the specific frequencies listed in claim 27, would have been a routine design choice and obvious.

8. Claims 4-6, 8, 10, 20, and 29 is/are rejected under 35 U.S.C. 103 as being unpatentable over Shribman in view of Chauhan et al. US Patent Pub. 2010/0262650.

Regarding claims 4 and 5:

The method according to claim 1, wherein the communication by the client device with the first server is based on, or is according to, TCP/IP protocol or connection.

The method according to claim 4, further comprising establishing a connection with the first server, and wherein the communication with the first server is over the established connection, and wherein the communication by the client device with the first server uses TCP, and wherein the connection is established by performing 'Active OPEN' or "Passive OPEN'.

As noted as AAPA in the "614 patent, "Web browsers typically use TCP when they connect to servers on the World Wide Web." See the '614 Patent at 1:61-62.

Moreover, Shribman "teaches a further embodiment in which a communication device/client, mapped as above as the "first server" in claim 1, creates an established TCP connection with an agent (mapped as above as the "first server). ¶[0106)-[0107] & Fig. 14, For example, ¶ [0107] states: "To establish a connection, as shown by block 608, the client [first server] issues a TCPIP connect with the primary agent or one of the other agents [each client devices] if the primary agent does not succeed, to create a connection with the agent."

The Examiner notes that if it is determined that the communication is not in accordance to TCP/IP protocol or connection, the Examiner notes that Chauhan et al. teaches wherein the

communication by the client device with the server is based on, or is according to, TCP/IP protocol or connection (see [0286])."

TCP is also a generally applicable Internet standard protocol, further supplying a motivation to combine. As also discussed above, ActiveOPEN and PassiveOPEN are the only two choices for opening a connection under the TCP standards, and it therefore would likewise have been obvious—indeed, mandatory) to use one of the two available methods and thus represents a finite amount of available choices.

Regarding claim 6:

The method according to claim 5, wherein the communication by the client device with the first server is based on, or is according to, a Virtual Private Network (VPN) and the established connection is using a tunneling protocol.

Chauhan et al teaches wherein the communication by the client device with the server is based on, or is according to, a Virtual Private Network (VPN) see [0050], lines 8-9) and the established connection is using a tunneling protocol (see [(0107], lines 11-12))."

It would have been obvious to a POSITA before the effective filling date of the claimed invention to combine Shribman with Chauhan because incorporating the secure data transmission provisioning mechanism of Chauhan within the content transmission system of Shribman would provide the predictive result of allowing for more effective network security and secure transmission of data between peer clients and a network server when implementing an encrypted channel for an increased level of security via transmission through implemented network tunnels.

Regarding claim 8:

The method according to claim 1, wherein the first identifier comprises an IP address that is in IPv4 or IPv6 form.

Shribman discloses that the first identifier (i.e. Of the agent) is an IP address. ¶¶ [0069], [0072] and Fig. 7, showing each peer device having an IP address 310. The Examiner notes that IPv4 or IPv6 forms are the only two possibilities.

Nonetheless, Chauhan et al teaches wherein the first identifier comprises an IP address that is in IPv4 or IPv6 form (see [0207], lines 7-8)." It would have been obvious to have chosen one of the other of IPv4 and IPv6 as disclosed in Chauhan, since Chauhan addresses similar problems of Internet communication and IPv4 and IPv6 are the only two possibilities under the applicable Internet standards.

Regarding claim 10:

The method according to claim 9, wherein the client operating system consists or, comprises of, or is based on , one out of Microsoft Windows 7, Microsoft Windows XP, Microsoft Windows 8, Microsoft Windows 8.1, Linux, and Google Chrome OS.

Chauhan et al teaches wherein the client operating system consists or comprises of or is based on, one of Microsoft Windows 7, Microsoft Windows XP (see [0089], lines 14-16). Microsoft Windows 8, Microsoft Windows 8.1, Linux, and Google Chrome OS."

A POSITA before the effective filling date of the claimed invention could readily have used any of these operating systems in conjunction with the disclosures of Shribman, with predictable results. These are all widely deployed and well-known operating systems and, given the "614 patent's contemplation that general purpose devices could serve as clients, it would have been obvious to a POSITA to have selected one of these common operating systems for the client device.

Regarding claim 20:

The method according to claim 19, wherein the resource utilization is based on, or comprises, the processor or CPU time of executing one or more threads or processes, wherein the

resource utilization is based on, or comprises, the processor or CPU idling time, or wherein the resource utilization is based on, or comprises, the processor or CPU executing a system idle process.

Chauhan et al teaches wherein the resource utilization is based on, or comprises, the processor or CPU time of executing one or more threads or processes, wherein the resource utilization is based on, or comprises, the processor or CPU idling time (see[0155], lines 8-10, 'terminating an execution of a virtual machine), or wherein the resource utilization is based on, or comprises, the processor or CPU executing a system idle process." See 296.

Claim 20 thus reflects the implementation of claim 19, which could obviously be accomplished by using the same mechanisms as taught in Chauhan, with predictable results.

Regarding claim 29:

The method according to claim 1, wherein the web server uses HyperText Transfer Protocol (HTTP) and responds to HTTP requests via the Internet, and wherein the sending of the first content identifier to the web server comprises a HTTP request, or wherein the communication with the web server is based on, or uses, HTTP persistent connection, and wherein the first content includes, consists of, or comprises, a part or whole of files, text, numbers, audio, voice, multimedia, video, images, music, or computer program, or wherein the first content includes, consists of, or comprises, a part of, or a whole of, a web-site page.

As reflected ¶ [0081], the web server uses HTTP protocol and responds to HTTP requests via the Internet. Per ¶ [0089], the client device (agent) sends a content request (referred to as "the request") "directly to the [web] server." The antecedent of "the request" in ¶ [0089] is the HTTP request disclosed in Par. [0081] (the example given is the HTTP request "GET http: //www.aolcom/index.html HTTP/1.1"). The fact that a "web" server is involved in made clear in ¶ [0080], referring to "Web server 152."

9. Claims 11, 12 and 14 is/are rejected under 35 U.S.C. 103 as being unpatentable over Shribman in view of Chauhan and further in view of Bhatia US Patent Pub. 2013/0171964.

Regarding claim 11:

The method according to claim 9, wherein the client operating system is a mobile operating system.

As addressed above in connection with claim 9, the combination of Shribman and Chauhan render it obvious to implement the device using a mobile device.

Bhatia further discloses wherein the client operating system is a mobile operating system, See ([0018], lines 17-20 ("The mobile device 102, in illustrative embodiments, runs the ANDROID operating system."). Bhatia concerns a device layout wherein the mobile device acts as an intermediary to pass message packets from a device to which the mobile device is connected (such as a notebook computer or tablet), to a variety of application servers accessible to the mobile device over a broadband wide-area network (e.g., Internet).

It would have been obvious before the effective filling date of the claimed invention, in view of Bhatia that a mobile operating system could be used to implement the claimed functionality of the client device, with predictable results. Since it would have been obvious to do so on a mobile device (as discussed in connection with claim 9), the incremental further provision of running a mobile operating system on the mobile device (as in claim 11) would have also been obvious.

Regarding claim 12:

The method according to claim 11, wherein the mobile operating system is one out of Android version 2.2 (Froyo), Android version 2.3 (Gingerbread), Android version 4.0 (Ice Cream Sandwich), Android Version 4.2 (Jelly Bean), Android version 4.4 (KitKat), Apple iOS version 3, Apple iOS version 4, Apple iOS version 5, Apple iOS version 6, Apple iOS version 7, Microsoft Windows® Phone version 7, Microsoft Windows® Phone version 8, Microsoft Windows® Phone version 9, and Blackberry® operating system.

Bhatia teaches to use Android, without specifying the version. However, as of the claimed date of the invention (August 2013) using any one of the specified versions of Android, which is all the claim requires, would have been obvious, as a matter of current commercial availability and desire to address as large a market as possible. Since as discussed with regard to claims 9 and 11 it would have been obvious to implement claim 1 on a mobile device using a mobile operating system, the choice of such an OS, such as Android as in Bhatia, also would have been obvious.

Regarding claim 14:

The method according to claim 13, wherein the web browser is a mobile web browser.

As set forth above, Shribman in view of Chauhan and Bhatia discloses that the client runs a web browser as an application, and also teaches {as claimed in claim 11) that the client may run a mobile operating system.

10. Claim 28 is/are rejected under 35 U.S.C. 103 as being unpatentable over and further in view of Ebata et al. US Patent 6,513,061.

Regarding claim 28:

The method according to claim 1, further comprising sending, by the client device, a physical geographical location to the first server, and wherein the physical geographical location corresponds to the actual physical geographical location of the client device.

Shribman ¶[0088] describes sorting peers by geographic proximity to the requesting client, reflecting the relevance of geographic location.

Ebata discloses selecting proxies based upon location so as to improve response speeds when requesting content. See Ebata, 4:23-45, Like the Shribman "733 publication, Ebata concerns the selection of proxies to access a resource. See Ebata, 4:23-28 ("a main object of the present invention to provide a comfortable working environment to a client in which only by specifying a logical node name of a server for pro- viding a target service the client would like to reach, the most approximate proxy server to the client can be automatically selected').

Ebata discloses storing a list of IP addresses of proxies and their related location information. See Ebata, Fig. & 11:27-30 (the SPS information list (625) includes the IP address, the location information and the load condition of each of the SPSs [service proxy servers] 2 and 7 distributively located on the network.

It would be obvious before the effective filling date of the claimed invention to associate each of the IP addresses to the list of IP addresses as taught in Shribman with a corresponding geographical location, as disclosed in Ebata, as doing so would allow the client to select a proxy server based upon location so as to select the closest proxy and thus reduce response times, as suggested in Ebata. See Ebata, at 3:49-52 ("(The user on the LAN can reuse the resource or the data cached in the proxy cache server that is a local server, so that the user may enjoy a comfortable response to the access.").

A POSITA would have a reasonable expectation of success of making such a combination since Shribman teaches selection of a proxy according to certain criteria, which a POSITA understands could reasonably include geographic location, and since the combination is straightforward, yielding predictable results without significantly altering or hindering the functions performed by the system of Shribman. See

This parallels the mechanism already described in Shribman 'for making the agent selection based on "multiple factors, such as, for example, factoring the speed the reply by each agent," or in other words, screening out agents that do not respond quickly, i.e., within a small amount of time (5 ms).See ¶ [0084]; see also claim 29 ("wherein the list of one or more peer communication devices includes only the peer communication devices that are the most beneficial to the requesting client communication device in terms of speed of receiving information from the peer communication devices"), A POSITA would understand that merely adding an additional or different attribute to the selection mechanism would not pose any kind of impediment for implementation.

11. Claims 1, 2, 4, 7-13, 15-23, 25, 26 and 29 is/are rejected under 35 U.S.C. 103 as being unpatentable Mithyantha US Patent 8,972,602 in view of Hypertext Transfer Protocol—HTTP/1.1 (hereinafter RFC 2616).

Regarding claim 1:

A method for use with a resource associated with a criterion in a client device that communicates with a first server over the Internet, the client device is identified in the Internet using a first identifier and is associated with first and second state according to a utilization of the resource, the method comprising:

Mithyantha discloses a method that is used with a resource associated with a criterion in a client device. As set forth in col. 56, line 61-col. 57, line 10, Mithyantha uses a status monitor which is incorporated in each appliance device 200 concerns resources associated with the health of the device, which the monitor compares with relevant criteria. See also col. 23, lines 12-15 and col. 57, lines 11-19.

Mithyantha discloses appliance(s) 200 that serve as intermediaries and make Internet web requests on behalf of client(s) 102 when the appliance(s) 200 are in an active state. Appliance/device 200' of Mithyantha is considered a "client device" since Mithyantha itself expressly contemplates appliance/device 200 operating in a client role. See 6:31-35.

Appliance/device 200 is the first server because it operates in the role of a server, and is not the same physical device as the client device. Mithyantha discloses that appliance/device 200 in the above example is a "server" as commonly understood because appliance/device 200 accepts a connection from a client 102 in order to send back a response.

In addition, the Examiner notes, client device corresponds to appliance/device 200', first server corresponds to appliance/device 200 (between client(s) 102 and appliance 200'), resource corresponds to one of CPU, memory, or bandwidth (see col. 56, line 61 – col. 57, line 4), criterion corresponds to the amount of usage (threshold) of the resource (see col. 57, lines 4-14), First identifier corresponds to IP address associated with the client device (see col. 56, lines 17-29), First state corresponds to active/available and second state corresponds to inactive/unavailable (see col. 56, lines 61 – col. 57, line 14; col. 23, lines 12-36 ("Health monitoring program 216 may call any application programming interface (API) to determine a state ...of any portion of the appliance 200.")

initiating, by the client device, communication with the first server over the Internet in response to connecting to the Internet, the communication comprises sending, by the client device, the first identifier to the first server over the Internet;

Mithyantha discloses that the client device (appliance/device 200') sends a first identifier (the IP address associated with the appliance/device 200') to an upstream router 700 over the Internet (over networks 104/104'). See 55:61-56:60; 56:17-45; 25:24-31 (explaining an IP address can be a network identifier). The upstream router 700 maintains a routing table of the identifiers of the intermediary devices that comprise the network. See 57:19-21, 36-47, 58:22-26.

The Examiner notes that a POSITA would understand that, devices communicating over the Internet would send an identifier to facilitate communication. See 2:44-62, 25:24-33, 26:38-46, 55:44-58. In the multi-proxy embodiment disclosed in Figure 1B and discussed above, the client device (appliance/device 200') would send the first identifier to the first server (appliance/device 200) before it was delivered to upstream router 700, which is located between client(s) 102 and the first server (appliance/device 200) in the network path. See 53:5-19, 56:7-10, 17-27, Fig. 1B, Fig. 7A.

Mithyantha further discloses that the client device (appliance/device 200') sends the IP address "over the Internet in response to connecting to the Internet." First, both "client device" appliance/device 200' and "first server" appliance device 200 communicate with each other and other devices on the disclosed networks via TCP/IP over the Internet. See 56:67-57:4, 5:3-7 (networks 104 and 104' connecting each device in the disclosed system can be Internet networks), 19:4-8 (appliance/device 200' communicates via TCP/IP), 21:21-38 (appliance/device 200' compresses TCP/IP protocols, HTTP, and HTML protocols sent "bi-directionally" on the network). Second, Mithyantha discloses that intermediary devices advertise "a corresponding IP address . . . as devices become active or inactive." See at 58:38-61, Fig. 7C. Third, Mithyantha discloses that a device re-advertises its route to upstream router 700 upon return to the cluster. See 57:32-61. One reason a device 200' may leave the network is because it loses its connection to the Internet. See 57:36-39, 57:62-58:2. When device 200' regains its connection, it re-advertises its route (which, at a minimum, inherently includes its identifier) to the upstream router 700.

See 57:32-61, claim 10; 1:39-44 (routes inherently include IP addresses). Again, both the advertisement and the re-advertisement will necessarily be sent, over the network 104', to "first server" appliance/device 200 (acting as proxy between upstream router 700 and "client device" appliance/device 200'), and both the advertisement and re-advertisement comprise sending the client device's IP address. The disclosure of Mithyantha states that "[t]he client 102, server 106, and appliance 200 may be deployed as and/or executed on any type and form of computing device," providing general purpose "computing device 100"—which may be "any workstation, desktop computer, laptop or notebook computer, server, handheld computer, mobile telephone, [and] any other computer'"—as such an example. See 13:65-14:14, 16:51-64, Fig. 1E, Fig. 1F. Appliance/device 200 is a "server" and appliance/device 200' is a "client device."

To the extent it is consider that Mithyantha does not specifically disclose the initiating, by the client device limitation, the Examiner notes that it would have been obvious to a person of ordinary skill in the art before the effective filling date of the claimed invention to determine that the device in Mithyantha could initiate communication with other devices by sending a communication with the IP address or identifier of the device. It is noted that Mithyantha discloses that "client device" appliance/device 200' and "first server" appliance/device 200 communicate over Internet networks 104 and 104' using "any TCP/IP based protocol," including HTTP. See 5:3-7, 19:4-8, 21:21-38, 28:55-58, Fig. 1B. In addition, in Mithyantha appliance /device 200' sends its IP address to appliance/device 200 (the first server), this would have been obvious to a POSITA, in that Mithyantha expressly discloses that the IP address is sent to upstream router 700 through appliance/device 200, that the IP address thereby also goes to appliance/device 200, which relays it to upstream router 700. This is particularly clear, in view of the described embodiments in which each appliance /device 200 runs (i.e., has within it) vServer 702, which specifically has the capability for "receiving, intercepting, processing, and/or forwarding packets to and from nodes of the cluster, servers, clients, or any combination of devices." See 56:17-36.

It is therefore obvious if not implicit that in the routing illustrated in Mithyantha, the IP address of appliance/device 200 goes (or would go) from appliance /device 200' to appliance/device 200. The stated purpose of appliance/device 200 is optimizing network services for the other devices in the network (See

8:4-24). A POSITA would have been motivated to maximize this improvement by using well-known methods to incorporate this functionality upon startup of the respective devices, in order to make the improved capability provided by Mithyantha available immediately and automatically for all of the users' communications.

when connected to the Internet, periodically or continuously determining whether the resource utilization satisfies the criterion;

Mithyantha discloses that "a device 200a-200n "may comprise a status monitor 704a704m, referred to generally as status monitor(s) 704," for monitoring the health of devices on the network, which can "comprise monitoring CPU usage or load, memory usage, bandwidth . . . or any other status of the device, Vserver, network, or other entity." See at 56:61-57:4. "[A] status monitor of a device may periodically poll other devices for health status." See at 57:4-6. Further, the "status monitor 704 may determine that a monitored status or value exceeds a threshold, such as a CPU utilization rate past a threshold for a predetermined time." See 57:11-14.

Status monitor 704 functions when "connected to the Internet," as it is not only able to monitor the "client device" (appliance/device 200') on which it resides, but other network devices connected via the Internet as well. See 56:67-57:4; 5:3-7 (networks 104 and 104' are Internet networks); 19:4-8 (appliance/device 200' communicates via TCP/IP).

Likewise, Mithyantha discloses that appliance 200 executes a monitoring agent 197. See 27:51-52. Among other things, "the monitoring agent 197 measures and monitors the memory, CPU and disk usage and performance of the appliance 200." See 27:65-67, 12:5-7 (bandwidth). Monitoring agent 197 functions "when connected to the Internet," as the agent can "perform[] monitoring and performance measurement of any network resource or network infrastructure element, such as a client server, server farm, appliance 200, appliance 205, or network connection." See 11:62-66; 11:50-52 (monitoring web page requests and HTTP responses). "In some embodiments, the monitoring agent 197 monitors, measures and collects data on a predetermined frequency. In other embodiments, the monitoring agent

197 monitors, measures and collects data based upon detection of any type and form of event." See 11:46-50.

responsive to the determining that the utilization of the resource satisfies the criterion, shifting to the first state or staying in the first state;

Status monitor 704 monitors the resources of appliance/device 200', as well as other devices on the network. When status monitor 704 of "client device" appliance/device 200' determines that utilization on one of its resources does not exceed a threshold, device 200' shifts to or stays in a first state (active/available). Alternatively, if the status monitor determines that utilization does exceed the threshold, appliance/device 200' shifts to or stays in a second state (inactive/unavailable). Further:

> a status monitor 704 may determine that a monitored status or value exceeds a threshold, such as a CPU utilization rate past a threshold for a predetermined time. Responsive to the determination, the status monitor 704 may direct the vServer 702 of the device to withdraw its advertised route from the upstream router 700. This may be done to reduce or eliminate incoming traffic while the device resolves any issues or errors, to allow the device to reboot, or any other such reason. Once received, the router 700 may update its multipath routing table and continue routing traffic to other, active nodes of the cluster.

See 57:11-21; 59:13-50. Appliance/device 200' may shift between the first state and second state when, for example, appliance/device 200' exceeds a threshold and thereafter returns to utilization below the threshold: "if the unavailable node returns, it may re-advertise its route with the routing metric at its original value 706, such that the router 700 may begin including it in load balancing and multipath routing without requiring all devices to re-advertise their routes." See 57:57-61.

responsive to the determining that the utilization of the resource does not satisfy the criterion, shifting to the second state or staying in the second state;

As discussed, Mithyantha teaches appliance/device 200' staying in or shifting to the second state (see above discussion regarding vServer's withdrawn of an advertised route when a monitored status or value exceeds a threshold). See 57:11-21; 59:13-50.

responsive to being in the first state, receiving, by the client device, a request from the first server; and

When in the first state, appliance/device 200' receives requests from client(s) 102, via "first server" appliance/device 200. See 57:11-21; 59:13-50; 4:64-65. Such requests may include, for example, requests from a web browser on client(s) 102, sent via upstream router 700, appliance/device 200, and appliance/device 200', to web server(s) 106.

This is "responsive to being in the first [available] state," insofar as the state-determination mechanism in Mithyantha, by which a device may determine that it is unavailable, is designed to enable the upstream router 700 to "withdraw" the unavailable device from the set of routes to which tasks will be allocated, and "and continue routing traffic 20 to other, active [i.e., available] nodes of the cluster."

performing a task, by the client device, in response to the receiving of the request from the first server,

As discussed above, when in the first state, appliance/device 200' receives requests from client(s) 102, via "first server" appliance/device 200. See 57:11-21; 59:13-50; 4:64-65. The "task" performed is further described below with respect to the method claim steps concerning the "task."

wherein the method is further configured for fetching over the Internet a first content identified by a first content identifier from a web server that is distinct from the first server, and the task comprising:

Mithyantha discloses a system that can be used to fetch web content from a web server over the Internet. An HTTP request for web content can be made from a "web browser" on client 102. See 10:45-51, 60-62. That initial HTTP request is then sent to upstream router 700 that directs the request first to "first server" appliance/device 200, and thereafter "client device" appliance/device 200'. See 3:45-47; 4:64-65; 7:1-18; 56:7-10, 17-23; Fig. 1B; Fig. 7A; 11:50-52 (monitoring agent 197, found on appliance/device 200 and 200', detects "request[s] for a web page or receipt of an HTTP response"). "Client device" appliance/device 200', acting "as a proxy or access server to provide access to the one or more servers 106," then forwards the HTTP request to "web server" 106. See 6:52-53; 9:15-17, 53-55; 11:13-18; 23:55-58; 56:17-23. Each of these requests are routed via networks 104 and 104', which can be

Internet/World Wide Web networks, using "any TCP/IP based protocol," including HTTP. See 5:3-7; 19:4-8; 21:21-38; 28:55-58; Fig. 1B.

The Examiner notes that to the extent it is considered that Mithyantha does not disclose a first content identifier, a POSITA would have understood that HTTP requests involved content identifiers for web content, typically a URL.

For example, **RFC 2616**, which concerns HTTP 1.1, discloses "GET" requests, which include URLs, i.e., content identifiers. See RFC 2616 at 35. The HTTP request includes the content identifier (e.g., URL) so that the web content may be identified on the web server and returned to client 102. Facilitating such transactions is a primary point of Mithyantha. In the context of a claim that already expressly recites "fetching...a first content ...from a web server," it would have been obvious to a POSITA before the effective filling date of the claimed invention to include a content identifier (e.g., URL) in passing a content request to the Web browser. Similarly, again to such extent as may not otherwise have been completely implicit, it would have been obvious to any POSITA familiar with a web browser that the HTTP web browser requests disclosed in Mithyantha, see e.g., See Mithyantha at 10:45-61, 11:50-52, 23:57-58, would cause a web server to respond with first content (e.g.. a web page) to be delivered to the web browser where the HTTP request originated.

The incorporation of the content identifiers for the claimed first content, would have predictively implemented a known technique (RFC 2616's GET request) in a known web browser/Internet system, yielding predictable results.

receiving, by the client device, the first content identifier from the first server;

Mithyantha discloses a network of devices capable of handling HTTP web requests for web pages sent between a web browser and a web server. This claim limitation is met when "client device" (appliance/device 200') receives the HTTP web request from "first server" (appliance/device 200). See 3:45-47; 4:64-65; 7:1-18; 9:15-17; Fig. 1B. As set forth above, a POSITA would understand that the HTTP web browser requests disclosed in Mithyantha would include a content identifier (e.g., a URL) that identifies first content (e.g., a web page). See 10:45-61; 11:50-52; 23:57-58.

sending, by the client device, the first content identifier to the web server;

After the client device receives the first content identifier, "client device" appliance/device 200', serving as proxy, sends the HTTP request (containing the first content identifier, e.g., URL) to any of "web servers" 106a-c. See 4:64-65, 6:52-53, 9:15-17, 53-55, 23:57-58, Fig. 1B.

A set forth above POSITA would understand that the HTTP web browser requests disclosed in Mithyantha would include a content identifier (e.g., a URL) that identifies first content (e.g., a web page). See 10:45-61; 11:50-52; 23:57-58; see also See 5:38-41(explaining web requests include content identifiers such as URL), 20:4-27, 63:26-29. Mithyantha further explains that a content identifier is sent to the web server: "[T]he server 106 may run an application, which for example, may be...a web or Internet server[.]" See 11:13-17. Server 106 may provide an HTTP service, and appliances 200 and 200' may "receive[], intercept[] or otherwise process|[] communications between a client 102 and a server 106[.]" See 23:57-64. Appliance 200 (or appliance 200' in Fig. 1B embodiment) may establish a connection to a service 270 of a server 106 that comprises a "web server" or an "HTTP server." See 24:7-14. Client 102 may make a request for a webpage, via upstream router 700, which is intercepted by appliance 200 (and thereafter device 200' in Fig. 1B embodiment). See 27:43-45, 56:23-25 (device 200 receives incoming client traffic from router 700), 7:1-18, Fig. 1B, Fig. 7A.

receiving, by the client device, the first content from the web server in response to the sending of the first content identifier; and

After the client device sends the first content identifier (e.g., URL) to "web server" 106, the "client device" appliance/device 200' receives, in response and still serving as proxy, the first content (e.g., a web page) from "web server' 106. See 4:64-65; 9:15-17; 53:26-28; 56:17-23; Fig. 1B; 20:9-16 (appliance/device 200 caching markup language content); 11:50-52 (appliance/device 200 monitoring HTTP response); 21:21-38 (appliance/device 200 compressing bi-directionally "any markup languages').

After the client device sends the first content identifier (e.g., URL) to "web server" 106, the "client device" appliance/device 200' receives, in response and still serving as proxy, the first content (e.g., a web page) from "web server' 106. See 4:64-65; 9:15-17; 53:26-28; 56:17-23; Fig. 1B; 20:9-16

(appliance/device 200 caching markup language content); 11:50-52 (appliance/device 200 monitoring HTTP response); 21:21-38 (appliance/device 200 compressing bi-directionally "any markup languages').

sending, by the client device, the received first content to the first server.

After the client device receives the first content (e.g., a web page), the client device (appliance/device 200'), still serving as proxy, sends that content to the first server (appliance/device 200) so that the first content (e.g., a web page) can be forwarded to the originator of the HTTP request, the web browser of client 102. See 4:64-65; 7:1-18; 9:15-17; 53:26-28; 56:17-23; Fig. 1B.

Thus, the HTTP web browser requests disclosed in Mithyantha would cause a web server to respond with first content (e.g., a web page) to be delivered to the web browser where the HTTP request originated. See 10:45-61; 11:50-52; 23:57-58.

Regarding claim 2:

The method according to claim 1, wherein the determining is performed by the client device.

Mithyantha discloses that the "client device" (appliance/device 200') "may comprise a status monitor 704a-704m, referred to generally as status monitor(s) 704." See 56:61-67; Fig. 7A. As discussed above with respect to claim 1, the status monitor(s) 704 satisfy the "determining" limitation of claim 1 via their monitoring of resource utilization, including CPU, memory, and bandwidth utilization

Mithyantha similarly discloses that the "client device" appliance/device 200' "executes the monitoring agent 197," which performs similar functionality. See 27:51-52; Fig. 2B. As a further example, Mithyantha states, "In one embodiment, a status monitor 704 may determine that a monitored status or value exceeds a threshold, such as a CPU utilization rate past a threshold for a predetermined time. Responsive to the determination, the status monitor 704 may direct the vServer 702 of the device to withdraw its advertised route from the upstream router 700." See 57:11-16; 56:61-63 ("In some embodiments, a device 200a-200n may comprise a status monitor 704a-704n, referred to generally as status monitor(s)').

Regarding claim 4:

The method according to claim 1, wherein the communication by the client device with the first server is based on, or is according to, TCP/IP protocol or connection.

Mithyantha discloses that the "client device" appliance/device 200' communicates with the "first server" appliance/device 200 via network 104'. See 7:1-18; Fig. 1B. Network 104' can be the Internet, over which appliance/device 200' communicates with appliance/device 200 using "any TCP/IP based protocol." See 5:3-7; 19:4-8; 21:21-38; 28:55-58; Fig. 1B. Furthermore, appliance/devices 200 and 200' each "comprise[] one network stack 267, such as a TCP/IP based stack, for communicating with the client 102 and/or the server 106." See 19:4-6. Appliance/devices 200 and 200' also contain multi-protocol compression logic 238. See 19:27-30; Fig. 2A. That compression engine "compresses bi-directionally between clients 102a-102n and servers 106a-106n any TCP/IP based protocol." See 21:24- 27. Similarly, client 102 comprises network stack 310 that "may comprise a transport control protocol (TCP) over the network layer protocol of the internet protocol (IP), generally referred to as TCP/IP." See 28:31-47. As discussed above, appliance/device 200' can communicate with client 102 via appliance/device 200. See 5:39-43; 7:1-18; Fig. 1B. Thus, communications between appliance/device 200' and client 102, passing through appliance/device 200, may be based on or according to the TCP/IP protocol.

Regarding claim 7:

The method according to claim 1, wherein the steps are sequentially executed.

As discussed with respect to claim 1, appliance/device 200' must send its identifier to upstream router 700, via appliance/device 200, for the router to add its advertised path to the routing table. As also discussed, once connected to the system, status monitor 704 (and/or monitoring agent 197) periodically or continuously determines resource utilization. In the event appliance/device 200' leaves the network, and thereafter returns, there will be a shift from first state, to second state, and back to first state. Following such a shift, appliance/device 200' may receive a request from the first server, and thereafter perform the tasks, in sequential order, associated with fetching first content (e.g., a webpage) over the Internet.

Further, this claim refers in part to an order of steps that involve the HTTP protocol. Such use of HTTP was well understood, routine, and conventional. The sending of a content request to a web server and retrieval of web content was a standard usage of HTTP, as known to a POSA and as specified in RFC 2616. For example, RFC 2616 § 1.4 explains that "[t]he HTTP protocol is a request/response protocol" and that the request chain from client to server and back may involve intermediaries, such as proxies, to which the request is first sent.

Regarding claim 8:

The method according to claim 1, wherein the first identifier comprises an IP address that is in IPv4 or IPv6 form.

Mithyantha discloses that the "client device" (appliance/device 200') sends a first identifier (the IP address associated with the appliance/device 200) to the "first server" appliance/device 200 over the Internet. See 56:27-36, 56:53-57. This is evidenced, in part, by the fact that upstream router 700 maintains a routing table of the identifiers of all devices in the network. See 57:19-21, 36-47; 58:22-26. Further, Mithyantha expressly discloses both IPv4 and IPv6. See 47:20-33.

Additionally, one means of routing traffic disclosed in Mithyantha is via "flow distributor 550," which "can comprise a Receive Side Scaler (RSS) or Receive Side Scaling (RSS) module 560," and which may be located on "any type and form of one or more systems, appliances, devices or components." See 43:11-25. "Flow distributor 550" may therefore be located on the upstream router 700. And Mithyantha discloses that the RSS module "can apply any type and form hash function .. . including TCP/IPv4, TCP/IPv6, IPv4, and IPv6 headers." See 47:20-33. Thus, in this embodiment, the first identifier would comprise an IP address that is in IPv4 or IPv6 form.

Regarding claim 9:

The method according to claim 1, wherein the client device is further storing, operating, or using, a client operating system.

Mithyantha discloses "client device" (appliance/device 200') that stores, operates, or uses a client operating system. For example, Mithyantha discloses that "the operating system [of appliance 200'] may

be any type and/or form of Unix operating system," as well as any version of "Microsoft® Windows operating systems... Unix and Linux operating systems... Mac OS® for Macintosh computers . .. any operating systems for mobile computing devices or network devices, or any other operating system capable of running on the appliance 200[']and performing the operations described herein." See 18:41-56; Fig. 2A. Additionally, appliance/device 200 may run on computing device 100, which "can be running any operating system," including every version of Windows, Unix, Linux, and Mac operating systems, and "any operating systems for mobile computing devices." See 14:3-5; 16:30-50; Fig. 1E.

Regarding claim 10:

The method according to claim 9, wherein the client operating system consists or, comprises of, or is based on, one out of Microsoft Windows 7, Microsoft Windows XP, Microsoft Windows 8, Microsoft Windows 8.1, Linux, and Google Chrome OS.

See above discussion in claim 9.

Regarding claim 11:

The method according to claim 9, wherein the client operating system is a mobile operating system.

See above discussion in claim 9.

Regarding claim 12:

The method according to claim 11, wherein the mobile operating system is one out of Android version 2.2 (Froyo), Android version 2.3 (Gingerbread), Android version 4.0 (Ice Cream Sandwich), Android Version 4.2 (Jelly Bean), Android version 4.4 (KitKat)), Apple iOS version 3, Apple iOS version 4, Apple iOS version 5, Apple iOS version 6, Apple iOS version 7, Microsoft Windows® Phone version 7, Microsoft Windows® Phone version 8, Microsoft Windows® Phone version 9, and Blackberry® operating system.

See above discussion in claim 9.

Regarding claim 13:

The method according to claim 1, further comprising executing an application, and wherein the application comprises a web browser.

Mithyantha discloses a web browser application operating on client 102. See 10:45-51, 60-62, 28:20-24.

Regarding claim 15:

The method according to claim 1, wherein the client device comprises, or consists of, a portable or mobile device.

Mithyantha discloses "client device" (appliance/device 200') that may run on computing device 100, which "can be any .. . handheld computer, mobile telephone, any other computer, or other form of computing or telecommunications device." See 14:3-5; 16:58-64. "[I]n one embodiment the computer 100 is a Treo. . . smart phone." See 16:53-55.

Regarding claim 16:

The method according to claim 15, wherein the mobile device comprises a smartphone.

See above citations set forth in claim 15

Regarding claim 17:

The method according to claim 1, for use with a set threshold value, and wherein the criterion is satisfied when the resource utilization is above or below the threshold.

As discussed above with respect to the "determining" and "shifting . . . or staying" steps, Mithyantha discloses that "a status monitor 704 may determine that a monitored status or value exceeds a threshold, such as a CPU utilization rate past a threshold for a predetermined time." See 57:11-14; 2:15-19; 3:4-8; claim 4.

Regarding claim 18:

The method according to claim 1, wherein the resource comprises, or consists of, a hardware component in the client device.

Mithyantha discloses that "a status monitor 704 may determine that a monitored status or value exceeds a threshold, such as a CPU utilization rate past a threshold for a predetermined time." See 57:11-

14; 2:15-19; 3:4-8; claim 4. Status monitor 704 can monitor "devices, vServers, network interfaces, processors, memory or storage devices, network connections, or other entities. Monitoring health may comprise monitoring CPU usage or load, memory usage, bandwidth. . . or any other status of the device, vServer, network, or other entity." See 56:63-57:4. Similarly, "the monitoring agent 197 measures and monitors," among other things, "the memory, CPU and disk usage and performance of the appliance 200," as well as transport layer connections, user sessions, and bandwidth utilization. See 27:65-67; 11:46- 12:7; 12:12-19; 45-49; 13:3-21; 34-44; 33:4-7. Thus, the "resource" of claim 1, one of CPU, memory, or bandwidth, may comprise or consist of a hardware component (e.g., CPU, memory, and/or network interfaces/input/output devices) in appliance/device 200'. See 14:15-18; 14:28-44; 15:26-39, 16:18-26; 13:65- 14:7; Fig. 1E; Fig. 1F. Further, the "resource" of claim 1, may comprise or consist of input or output capability, including, as one example, bandwidth of communication with another device over the Internet. See 13:65-14:7; 15:26-39; 16:18-26; Fig. 1E; Fig. 1F. Mithyantha also teaches monitoring CPU usage or load, and determining whether such CPU usage or load exceeds a threshold when the CPU utilization rate is past a threshold for a predetermined time. See 26:17-33. Mithyantha further discloses that "[i]n other embodiments, the appliance 200 executes the monitoring agent 197. In one embodiment, the monitoring agent 197 measures and monitors the performance of any application, program, process, service, task or thread executing on the appliance 200... . [I]n another embodiment, the monitoring agent 197 measures and monitors the performance of any transport layer connections of the appliance 200. In some embodiments, the monitoring agent 197 measures and monitors the performance of any user sessions traversing the appliance 200. See 27:51-62.

Regarding claim 19:

The method according to claim 18, wherein the hardware component comprises, or consists of, a processor or Central Processing Unit (CPU) operation in the client device.

See above discussion set forth in claim 18.

Regarding claim 20:

The method according to claim 19, wherein the resource utilization is based on, or comprises, the processor or CPU time of executing one or more threads or processes, wherein the resource utilization is based on, or comprises, the processor or CPU idling time, or wherein the resource utilization is based on, or comprises, the processor or CPU executing a system idle process.

See above discussion set forth in claim 18.

Regarding claim 21:

The method according to claim 18, wherein the hardware component comprises, or consists of, a memory in the client device, and wherein the resource utilization is based on, or comprises, an amount of used or unused location or space of the memory.

See above discussion set forth in claim 18.

Regarding claim 22:

The method according to claim 1, wherein the resource comprises, or consists of, input or output capability.

See above discussion set forth in claim 18.

Regarding claim 23:

The method according to claim 22, wherein the resource comprises, or consists of, communication bandwidth of communication with another device over the Internet.

See above discussion set forth in claim 18.

Regarding claim 25:

The method according to claim 1, further comprising periodically sending, by the client device, a message that comprises a status of the client device, or is in response to the status of the client device.

As discussed above with respect to the "determining" and "shifting . . . or staying" steps in claim 1, Mithyantha discloses periodic or continuous determining concerning resource utilization, as well as shifting of states. Mithyantha further discloses that when upstream router 700 refreshes its routing table at

a predetermined time interval, if "client device" (appliance/device 200') is in the first state (i.e., in response to the status of appliance/device 200'), appliance/device 200' sends a message to upstream router 700, via "first server" (appliance/device 200), re-advertising its path. "In some embodiments, after a predetermined period of time during which the upstream router 700's routing table may expire and be refreshed, the active vServers 702a-702c, 702n may re-advertise their routes with routing metric values restored or increased to their original values 706a-c, 706n." See 57:47-52; 57:52-61; 58:47-61; Fig. 7C.

Regarding claim 26:

The method according to claim 25, wherein the status is associated with being in the first or second state, and wherein the message is sent over the Internet to the first server.

See the above discussion set forth in claim 25.

Regarding claim 29:

The method according to claim 1, wherein the web server uses HyperText Transfer Protocol (HTTP) and responds to HTTP requests via the Internet, and wherein the sending of the first content identifier to the web server comprises a HTTP request, or wherein the communication with the web server is based on, or uses, HTTP persistent connection, and wherein the first content includes, consists of, or comprises, a part or whole of files, text, numbers, audio, voice, multimedia, video, images, music, or computer program, or wherein the first content includes, consists of, or comprises, a part of, or a whole of, a web-site page.

Mithyantha discloses client(s) 102 that communicate with "one or more servers 106a-106n," for example, when a web browser on a client 102 sends a request for a web page to a web server 106. See 4:58-64; 6:52-53; 10:45-51; 11:1-24, 50-52; 23:55-58. Server 106 can (1) be a web or HTTP server, (6:25-26; 52-53; 11:13- 17, 50-52; 21:21-34; 23:57-58; 24:12-14); (2) receive requests for webpages made over HTTP, (11:50-52; 23:57-58; 24:12-14; 27:43-45; 56:23-25); and (3) serve webpages in response to requests for the same (9:15-17; 11:13-17; 20:9-16; 21:21-38; 23:57-64; 24:9-14; 27:43-45). Thus, server 106 uses HTTP, receives HTTP requests with first content identifiers (URLs), and serves

content that consists of webpages. 10:54-67 (regarding video and/or audio), 21:21-38 (regarding voice (VoIP)), 32:10-29 (regarding files and/or computer program).

12. Claims 5 and 6 is/are rejected under 35 U.S.C. 103 as being unpatentable over Mithyanta in view of RFC 2616 and further in view of Transmission Control Protocol (hereinafter RFC 793).

Regarding claim 5:

The method according to claim 4, further comprising establishing a connection with the first server, and wherein the communication with the first server is over the established connection, and wherein the communicating by the client device with the first server uses TCP, and wherein the connection is established by performing 'Active OPEN' or 'Passive OPEN'.

The Examiner notes the '614 patent acknowledges that TCP "is one of the core protocols of the Internet protocol suite," and that "Active OPEN" and "Passive OPEN" are standard connection establishment approaches of the TCP protocol. See 1:54-57, 2:41-65.

A POSA would understand that the disclosure of the use of TCP/IP protocol, as described above, inherently discloses the standard TCP "Active OPEN" and "Passive OPEN" means of connecting.

Nonetheless, both "Active OPEN" and "Passive OPEN" would have been obvious based on Mithyantha combined with the general knowledge of a POSITA and RFC 2616, including knowledge of RFC 793 (TCP specification).

Mithyantha discloses that appliance/device 200' may communicate with the first server using TCP. Section 8.1.3. For the same reasons, RFC 793, similarly reflects TCP communication details, such as Active and Passive open, with which a POSITA would have been familiar. See Section 1.4; 2.7, Fig. 6.

Therefore, it would have been obvious to one of ordinary skill in the art that the TCP/IP disclosures of Mithyantha and RFC 2616 would establish a connection using 'Active OPEN' or "Passive OPEN' since those connection types are based on well-known connection types of TCP as described by RFC 793. This teaching would have yielded a predictable result when determining the functions associated with TCP.

Regarding claim 6:

The method according to claim 5, wherein the communication by the client device with the first server is based on, or is according to, a Virtual Private Network (VPN) and the established connection is using a tunneling protocol.

Mithyantha discloses that appliance/devices 200' and 200 can "provide[] a secure virtual private network connection from a first network 104 of the client 102 to the second network 104' of the server 106, such as an SSL VPN connection," and therefore can operate as tunneling devices. See 9:18-21. For example, "the encryption engine 234" on appliance/devices 200' and 200 "uses a tunneling protocol to provide a virtual private network between a client 102a-102n and a server 106a-106n." See 21:13-16. And "the vServer 275" on appliance/device 200' and 200 "includes any logic, functions, rules, or operations to perform any embodiments of the techniques described herein, such as SSL VPN 280," including the establishment of "end-to-end secure transport layer connection." See 24:3-5; Fig. 2B; 25:20-23. Mithyantha also discloses that the appliance/devices 200' and 200 "establish[] a VPN or SSL VPN connection based on the policy engine's 236 [(within appliance/devices 200' and 200)] authentication and/or authorization of a remote user or a remote client 102." See 24:44-48. Mithyantha further discloses that "[i]n one embodiment, the encryption engine 234 uses a tunneling protocol to provide a virtual private network between a client 102a-102n and a server 106a-106n." See 12:13-16. In addition, "the appliance 200 provides one or more of the following services, functionality or operations: SSL VPN connectivity 280[.]" See 23:49-51,

13. Claims 1, 2, 4, 5, 7-10, 13, 17-20, 22, 23, 25, 26, and 29 is/are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over MorphMix in view of RFC 2616.

Regarding claim 1:

A method for use with a resource associated with a criterion in a client device that communicates with a first server over the Internet, the client device is identified in the Internet using a first identifier and is associated with first and second state according to a utilization of the resource, the method comprising:

As discussed below, the claim recites a "client device" that, when it is in an active state, serves as an intermediary for a content request from a "first server" to a "web server." As further discussed below, MorphMix discloses nodes that serve as intermediaries and make Internet web requests on behalf of other nodes when the intermediary nodes are in an active state.

The Examiner notes the following:

> Client device: last node (*c*).
> First server: intermediate node (*b*).
> Web server: server (*s*).
> Resource: either (1) tunneling resources or (2) communication resources.
> Criterion: either (1) sufficient resources or insufficient resources or (2) sufficient CREDIT or insufficient CREDIT.
> First state: either (1) available for tunneling or (2) able to relay communications.
> Second State: either (1) not available for tunneling or (2) not able to relay communications.

In this example, the "first server" is the node that provides the first content identifier to the first client device (which is final node *c*). Node *b* is the first server because it: (1) operates in the role of a server, and (2) is not the same physical device as the first client device. MorphMix discloses that node *b* in the above example is a "server" as commonly understood because node *b* accepts a connection from node *a* in order to send back a response. The "first server" is the prior node *s*, which had "forwarded" the payload" to node *c* (the first client device). See p. 105.

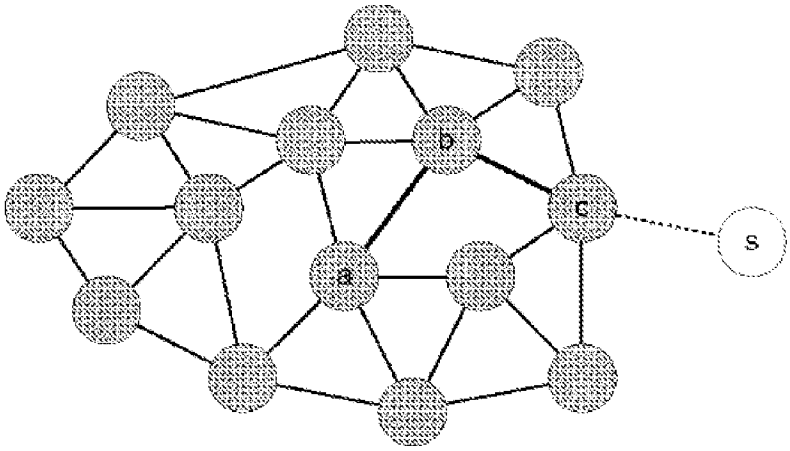

Figure 5.1: *Basic idea of MorphMix.*

As MorphMix notes, "[s]ending data back from the server to the client works exactly in the opposite way." See p. 105. MorphMix makes clear that every node "can itself establish circuits via other nodes to access a server anonymously, but can also be part of circuits established by other nodes and relay data for them at the same time." See p. 98.

initiating, by the client device, communication with the first server over the Internet in response to connecting to the Internet, the communication comprises sending, by the client device, the first identifier to the first server over the Internet;

MorphMix discloses that the client device (node *c*)) sends a first identifier (node (*c*)'s IP address) to the first server (node (*b*)) over the Internet. MorphMix makes clear that every node "can itself establish circuits via other nodes to access a server anonymously, but can also be part of circuits established by other nodes and relay data for them at the same time." See p. 98.

An early step in building the MorphMix network is establishing "neighbours"—nodes connected by "virtual links." See pp. 98-99, Fig. 5.1; pp. 267-68.

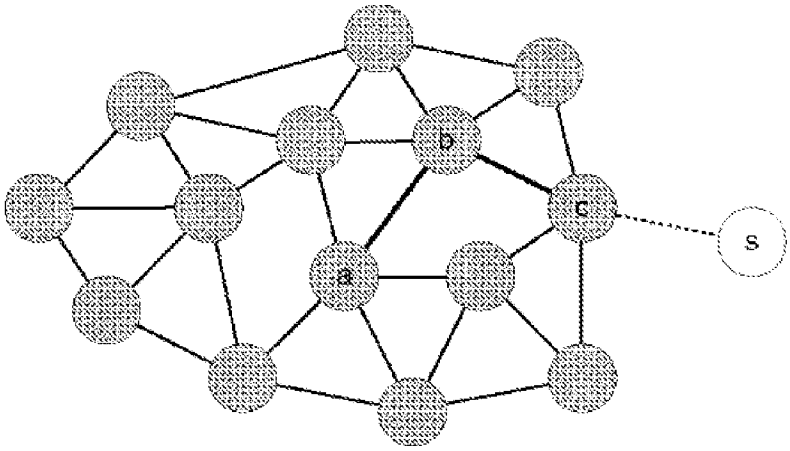

Figure 5.1: *Basic idea of MorphMix.*

Continuing with the example path described regarding the preamble, a virtual link can be created between node (*c*) and node (*b*) by node (*c*) "first establish[ing] a TCP connection with *b* using ip_b and p_{mmb}," node (*b*)'s IP address and port. See p. 267. Thus, this communication is over the Internet. Node (*c*) then sends a link request message ("Link REQ") that contains, among other things, node (*c*)'s IP address and port. See p. 267. Once node (*b*) accepts the connection and replies to node (*c*), the two nodes are "neighbours" via a virtual link and can commence with the circuit-based communications disclosed in MorphMix and described below. See pp. 98-99, pp. 267-68.

MorphMix acknowledges that "[n]odes can join and leave the system at any time... ." See p. 98. MorphMix further acknowledges that "nodes may be shut down by their operators." See p. 199. In the event of an operator shutdown, for example, when a MorphMix user shuts down his/her computer, "the TCP connections from and to the computer will be correctly terminated [41], which tells the neighbours immediately that the connections and all tunnels using them are no longer usable." See p. 199. Upon a restart of the computer and reconnection to the Internet, the initial step of establishing "neighbour" status would then take place gain. Both the initial set-up procedure, taking place after connection to the Internet, as well as the restart scenario just described, are examples of the "client device" initiating communication with the "first server" "in response to connecting to the Internet."

MorphMix utilizes a "peer discovery mechanism that enables MorphMix nodes to learn about other nodes." See p. 131. "[T]he MorphMix protocol includes peer discovery messages .. . which allow a node to ask another MorphMix node for information about other nodes, i.e. their IP addresses, ports on which they are listening for connection requests, public keys, and node levels." See p. 132, pp. 269-71 (discussing peer discovery messages and the need for a pre-established virtual link), See pp. 267-68 (discussing LINK REQ messages (that include the IP address of the sending node) used to establish virtual links). "[A] node always includes its own node information when establishing a virtual link to another node…[T]his is necessary because otherwise, a node could never inform other nodes about itself and consequently, no other node would establish a virtual link to it." See p. 133. Thus, MorphMix discloses that a node (e.g., node (*c*)), when joining MorphMix, sends its node information to other nodes (e.g., node (*b*)), including its IP address.

Further, communications between nodes (including a communication from node (*c*) to node (*b*)) use HTTP and TCP communications. See at pp. 102-03 ("In general, MorphMix can anonymize any TCP-based application but in its first version, we focus on web browsing and MorphMix supports HTTP (both versions 1.0 and 1.1) and HTTPS."), p. 99 (MorphMix stating that "there is a TCP connection" between nodes), p. 101 (nodes *a*, *b*, and *c* "are communicating directly with each other across TCP connections"), p. 267 ("Whenever two nodes in MorphMix communicate directly with each other, they establish a virtual link. To do so, *a* establishes a TCP connection with *b* using ip_b and p_{mmb} ....").

when connected to the Internet, periodically or continuously determining whether the resource utilization satisfies the criterion;

A first example of a determination, as described in this limitation, takes place during an anonymous tunnel setup. Nodes are added to tunnels "hop-by-hop." See p. 116. Further, it is assumed in this example (disclosed in MorphMix) that a tunnel already exists between node (*a*) and node (*s*), and that a virtual link already exists between node (*s*) and node (*c*) (as explained above regarding the preceding claim limitation). See p. 116. The tunnel setup involves, in relevant part, node (*c*) receiving a tunnel set-up message and "check[ing] if it is indeed willing to accept an anonymous tunnel from *s*." See pp. 116-18

(step 6). If yes, node (*c*) responds OK and participates in key encryption exchanges to build the tunnel. See pp. 116-18 (steps 6-10). However, "[i]f *c* does not accept being the next node in the tunnel or if there is another problem, *c* replies with a NEXT FAIL message... ." See p. 269. Node (*c*) may decline, for example, because willingness to be appended to an anonymous tunnel depends in part on whether node (*c*) determines it has sufficient "spare resources" to support the tunnel. See p. 114, p. 148, p. 232 ("A node can easily communicate with its neighbours to learn which of them are currently participating in MorphMix and have spare resources to accept new anonymous tunnels."). An example of such "spare resources" includes the resources needed to perform "the computational overhead imposed by public-key cryptography operations" associated with the tunnel. See at pp. 140-41, p. 217, p. 221. By client device node (*c*) determining whether it meets the criterion (whether it has sufficient "spare resources" to support the tunnel), MorphMix discloses this limitation.

Further, this "determining" is performed "periodically or continuously" by node (*c*) due to the policy that "anonymous tunnels are only used for a limited time"—specifically, ten minutes. See p. 114. The ten-minute time limit, combined with MorphMix's expectation that any node can utilize five tunnels at a given time, "results in setting up a tunnel every two minutes on average." See p. 114. Thus, in the example above, node (*c*) will be forced to "periodically or continuously" determine whether it has sufficient "spare resources" to support requested tunnels.

A second example of a determination, as described in this limitation, takes place as a result of "flow control messages" utilized by MorphMix, which allow any node (including node (*c*)) to avoid congestion. See p. 272. MorphMix's flow control utilizes "a credit based scheme" that "works bidirectionally between two neighbouring nodes along an anonymous tunnel." See p. 272. MorphMix discloses:

> A node *a* gets an initial credit, which corresponds to the number of cells it is allowed to send to one specific neighbour *b* for a particular anonymous tunnel identified with atID ID on the virtual link between a and *b*. This initial credit is set to 50 cells, which means that a is allowed to send 50 cells belonging to this anonymous tunnel to *b* before it must wait. Node *b* counts itself the cells with atID ID it has received from a and already forwarded to the next node along the tunnel (or to the client application(s) if the *b* is the initiator or to the host(s) if the *b* is the final node). Whenever *b* has forwarded 30 cells, it sends a CREDIT message back to *a* which sets the credit

> for the tunnel with atID ID back to 50... . [I]f *b* cannot forward the cells from a anymore, it simply stops sending back CREDIT messages, which prevents *a* from sending cells along the tunnel identified with atID ID after its credit it used up. See pp. 272-73.

In the exemplary nodes (*a*) to (*b*) to (*c*) to server (*s*) path discussed above, each of nodes (*a*), (*b*) (first server), and (*c*) (client device) employs this same credit-based flow control, "periodically or continuously" determining whether their resource utilization satisfies a criterion. For example, each node can determine whether it is able to forward message cells. If so, it sends back CREDIT messages to the preceding node. If not, it does not send CREDIT messages, and stops receiving additional cells.

The "determining" takes place "when connected to the Internet," as both tunnel setup, and flow control, utilize messages sent from node to node, indicating activity taking place when connected to the Internet. See at p. 96 (goal of MorphMix is anonymous Internet access), p. 96 (Internet access required to join and use MorphMix), pp. 98-100 (nodes communicate via IP addresses, ports, and TCP connections).

responsive to the determining that the utilization of the resource satisfies the criterion, shifting to the first state or staying in the first state;

With respect to the above discussion on tunnel setup, node (*c*), based on the sufficiency of its resources to support a tunnel, will either accept or decline a tunnel setup request. If there are sufficient resources, the criterion will be determined satisfied and the tunnel setup will be accepted. See pp. 116-17, p. 269. Thus, MorphMix discloses the client device shifting to or staying in the first state (available for tunneling). If sufficient resources are not available, the criterion will be determined not satisfied and the tunnel setup will be declined. See p. 269, p. 114, p. 148, p. 232.

MorphMix also discloses the client device shifting to or staying in the second state (not available for tunneling). In the second example, focused on flow control, node (*c*), based on the amount of cells sent to server (*s*), will either send a CREDIT message back to node (*b*), or it will not. See pp. 272-73. If the balance of cells warrants a CREDIT message, node (*c*) will determine that the criterion has been satisfied and a CREDIT message will be sent to node (*b*). See pp. 272-73. This will result in node (*c*) shifting to or staying in the first state (able to relay communications).

If the balance of cells does not warrant a CREDIT message, node (*c*) will determine that the

criterion has not been satisfied and no CREDIT message will be sent to node (*b*). See pp. 111 272-73. This will result in node (*c*) shifting to or staying in the second state (not able to relay communications).

responsive to the determining that the utilization of the resource does not satisfy the criterion, shifting to the second state or staying in the second state

MorphMix teaches the "client device," node (*c*), staying in or shifting to the second state. MorphMix therefore discloses this claim limitation. See p. 237

responsive to being in the first state, receiving, by the client device, a request from the first server; and

MorphMix discloses that node (*c*) receives a request from node (*b*) when in the first state (available for tunneling and/or able to relay communications). See pp. 103-05, Fig. 5.4. Such a request may include, for example, HTTP requests for web content from initiator node (*a*) to server (*s*), sent using "an anonymous tunnel from a via b to c." See pp. 103-05, Fig. 5.4, p. 207 (downloading web pages from a web server using HTTP), p. 218 ("[S]ending a web request as an initiator means that all other nodes along the anonymous tunnel must send and receive this request, too. Similarly, the reply sent back by the web server must be sent and received by the final and all intermediate nodes along the tunnel."). For example, step 9 of MorphMix's communication path description explains that node (*b*) forwards the HTTP request to node (*c*), and steps 10-12 explain how node (*c*) receives and processes that request. See p. 105. MorphMix therefore discloses this claim limitation.

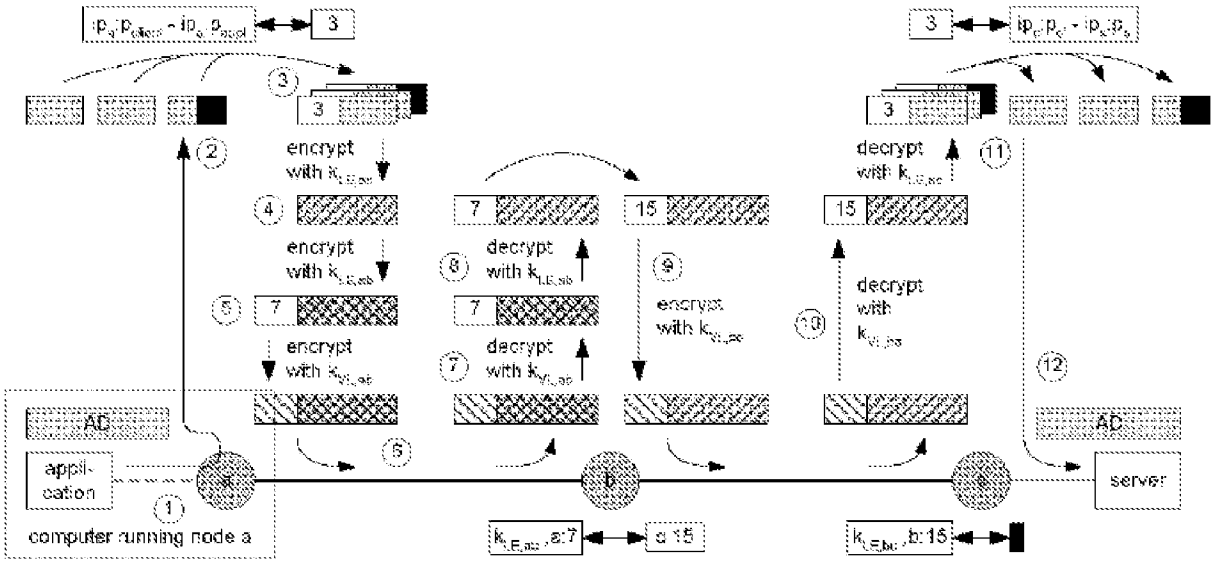

Figure 5.4: *Anonymous connections and cell forwarding.*

performing a task, by the client device, in response to the receiving of the request from the first server,

As set forth above, when in the first state, "client device" node (*c*) receives requests from node (*a*), via "first server" node (*b*). See pp. 103- 05, Fig. 5.4, p. 207. The "task" performed is further described below with respect to the method claim steps concerning the "task."

wherein the method is further configured for fetching over the Internet a first content identified by a first content identifier from a web server that is distinct from the first server, and the task comprising:

As discussed above, MorphMix discloses a system that can be used to fetch content from a web server over the Internet. See p. 96 (goal of MorphMix is anonymous Internet access), p. 103 (MorphMix's focus is "web browsing" and it supports HTTP and HTTPS), p. 200 (performance of MorphMix is analyzed using web browsing as the example application), p. 204 (the final node in a tunnel connects to "the web server"). MorphMix discloses that "in the case of web browsing, embedded objects in a web page are automatically requested by the browser, which results in quickly sending back data to the web server once the web browser has received the index file of a web page." See p. 107.

Analyzing the data transmitted along the anonymous tunnel, MorphMix discloses that:

> sending a web request as an initiator means that all other nodes along the anonymous tunnel must send and receive this request, too. Similarly, the reply sent back by the web server must be sent and received by the final and all intermediate nodes along the tunnel. The effective web replies received at the initiators... . See p. 218.

An HTTP request for web content can be made from a client application (e.g., a web browser), which sends the request to node (*a*) in the ongoing example. See p. 104. That HTTP request is then sent to a first server (node (*b*)) that then directs the request to a client device (node (*c*)). See pp. 104-05. The client device (node (*c*)) then forwards the HTTP request to web server (*s*). See pp. 104-05. The web server(s) disclosed in MorphMix are distinct from the first server (node (*b*)). The sought-after first content (e.g., a webpage) is returned via the same path: "Sending data back from the server to the client works exactly in the opposite way." See p. 105. As discussed above, node (*b*) and node (*c*) are different, communicate with one another, and are not the same device.

The "first content" and "first content identifier,' as well as the reminder of the method steps that "comprise" this claim element, are discussed in detail below.

To the extent that it is considered that MorphMix does not specifically disclose "first content identifier", the Examiner notes that this claim limitation would have been obvious based on the disclosure of MorphMix combined with the general knowledge of a POSA and RFC 2616.

MorphMix discloses the use of HTTP and requests sent from a web browser to a web server. See p. 103, p. 107, p. 168, p. 204, p. 207. A POSA would have understood that HTTP requests rely on content identifiers to identify desired web content, typically in the form of a URL (Uniform Resource Locator) or which is more generally known as a URI (Uniform Resource Identifier). RFC 2616, which concerns HTTP 1.1 and which is cited in the Patent, discloses use of a URL 20 times and a URI 235 times, both which identify web content. As RFC 2616 states, "URIs have been known by many names: [including] Uniform Resource Locators (URL)... . [A]s far as HTTP is concerned, Uniform Resource Identifiers are simply formatted strings which identify . . . a resource." RFC 2616 at § 3.2. As explained above, node (*c*) receives an HTTP request from node

(*b*). MorphMix pp. 103-05, Fig. 5.4. That HTTP request includes the web content identifier (e.g., URL) so that the web content may be identified on the web server and the web content may be returned to the requesting browser. Facilitating such transactions is a primary point of MorphMix. Even if the HTTP request is determined not to include a first content identifier, it would have been obvious to a POSA before the effective filling date of the claimed invention to include a URL or other such content identifier in that request.

receiving, by the client device, the first content identifier from the first server;

As discussed above MorphMix discloses node (*c*) acting as a proxy between nodes (*a*) and (*b*), on the one side, and server (*s*), on the other side. This is shown in Figure 5.1. See p. 98. The "first server" (node (*b*)), forwards the "first content identifier" (included in the application data within the payload) to the receiving "client device" (the final node (*c*)). See pp. 104-05.

MorphMix discloses that the "application data," which is being sent to the web server, includes the data identifying the requested web page. For example, as background, MorphMix discloses that an HTTP request includes both a "Host field" in an HTTP header and a "GET field" in the HTTP request line that together constitute the URL identifier of the content being requested from the web server. See pp. 4-6. The headers in HTTP requests that may be sent using MorphMix also include HTTP referer [sic] fields, which "tells a web server .. . the uniform resource locator (URL) of the web object the user has downloaded before." See p. 3.

MorphMix discloses its applicability to web browsing: "For instance, in the case of web browsing, embedded objects in a web page are automatically requested by the browser, which results in quickly sending back data to the web server once the web browser has received the index file of a web page." See p. 107. Analyzing a web browsing scenario, MorphMix discloses that "web requests sent and replies received by initiators" corresponds to the "application data sent and received if the web server is contacted directly." See p. 214. Thus, a primary focus of MorphMix is on "web browsing," which involves the sending of HTTP requests. See p. 103, p. 207 (downloading web pages from a web server using HTTP).

Figure 5.4 of MorphMix shows that application data (AD) within the payload is received by node (*c*) (client device) from node (*b*) (first server). See Fig. 5.4; pp. 103-05 (specifically, step 9). This application data includes an HTTP request, which includes the URL of the content being requested from a web server.

MorphMix discloses that "a node" simply must have an IP address and port number. See at p. 98 (node is "identified with its public IP address"). MorphMix discloses, for example, running MorphMix on both typical computer software, such as Linux (kernel 2.4.17), as well as on "Linux-VServer" software capable of operating a node. See pp. 109, 129. Node *b* is the first server because it: (i) operates in the role of a server, and (ii) is not the same physical device as the first client device. MorphMix discloses that node (*b*) in the above example is a "server" as commonly understood because node (*b*) accepted a connection from node (*a*) in order to send back a response, because "[s]ending data back from the server to the client works exactly in the opposite way." See p. 105. Every node "can itself establish circuits via other nodes to access a server anonymously, but can also be part of circuits established by other nodes and relay data for them at the same time." See p. 98.

sending, by the client device, the first content identifier to the web server

After the client device (node (*c*)) receives the first content identifier, node (*c*), serving as proxy, sends the application data (AD) (which contains the first content identifier, e.g., URL) to the web server (server (*s*)). See pp. 103-105 (specifically, steps 11-12), p. 218 ('[S]ending a web request as an initiator means that all other nodes along the anonymous tunnel must send and receive this request, too. Similarly, the reply sent back by the web server must be sent and received by the final and all intermediate nodes along the tunnel."). See Fig. 5.4.

As explained immediately above a POSITA would understand that the "application data" included in the exchanged payload includes an HTTP request that includes a content identifier (e.g., URL) identifying the target web page and related content. MorphMix discloses and experimentally evaluates "the performance of MorphMix using web browsing as the example application... . We usually evaluate the time it takes to completely download a web page." See p. 205. MorphMix discloses the download

times of web content when the web server is contacted anonymously through MorphMix, using both HTTP 1.0 and HTTP 1.1, where HTTP 1.0 requires a new connection between the client device (node (*c*)) and the web server (server (s)) for each HTTP request, while HTTP 1.1 could reuse an existing connection between the client device and web server. See p. 207. To request these web content, the client device sends an HTTP request including a content identifier to the web server.

receiving, by the client device, the first content from the web server in response to the sending of the first content identifier; and

After the client device sends the first content identifier (e.g., URL) to web server (s), the client device (node (*c*)) receives, in response and still serving as proxy, the first content (e.g., web page content) from the web server (s). See pp. 103-05 (specifically, step 12) ("Sending data back from the server to the client works exactly in the opposite way."). Because the final node (client device node (*c*)) is the only node connected to the server, the web server's response with first content must be received by the client device. See p. 100 (one TCP connection between final node and web server), p. 103 (same), p. 207 (same), Fig. 5.1, Fig. 5.4, p. 218 ("[S]ending a web request as an initiator means that all other nodes along the anonymous tunnel must send and receive this request, too. Similarly, the reply sent back by the web server must be sent and received by the final and all intermediate nodes along the tunnel.").

A POSA would understand that a web server responds to an HTTP request by sending the web page content (the first content) specified by the request's URL (content identifier) to the requesting client (node (*c*)), as the client device is a proxy or tunnel device that establishes a TCP connection with the web server over which it sends an HTTP request. As such, a POSA would understand that, subsequent to the client device (node (*c*)) sending a HTTP request to the web server, it receives the requested web content. See p. 100, p. 103, p. 107 ("For instance, in the case of web browsing, embedded objects in a web page are automatically requested by the browser, which results in quickly sending back data to the web server once the web browser has received the index file of a web page."), p. 168 ("[I]n the case of web browsing, if downloading a web page is interrupted, the page can simply be downloaded again using another tunnel... ."), p. 204 (discusses a "web server... sending a web reply after having completely received a

web request'), 205 (downloading a web page is only completed after "receiving the last byte of the complete page).

sending, by the client device, the received first content to the first server.

After the client device receives the first content (e.g., web page content), the client device (node (*c*)), still serving as proxy, sends that content to the first server (node (*b*)), which in turn provides the first content (e.g., web page content) up the same path until finally delivered to the requesting web browser. See pp. 103-05 (specifically, step 12) ('Sending data back from the server to the client works exactly in the opposite way."), 104 ("There exists an anonymous tunnel from *a* via *b* to *c*."), Fig. 5.1, Fig. 5.4, p. 218 ("[S]ending a web request as an initiator means that all other nodes along the anonymous tunnel must send and receive this request, too. Similarly, the reply sent back by the web server must be sent and received by the final and all intermediate nodes along the tunnel.").

A POSA would understand that the HTTP web browser requests disclosed in MorphMix would cause a web server to respond with first content (e.g., web page content) to be delivered to the web browser where the HTTP request originated. See pp. 100, 103, 107.

Regarding claim 2:

The method according to claim 1, wherein the determining is performed by the client device.

With respect to the tunnel setup disclosure set forth above, when client device node (*c*) is being appended to a tunnel, it is node (*c*) that receives a tunnel set-up message and node (*c*) that "checks if it is indeed willing to accept an anonymous tunnel from *b*." MorphMix at pp. 116-18 (step 6). Further, it is node (*c*) that either responds OK or declines the request, for example, because of insufficient "spare resources" to support the tunnel. See p. 114, pp. 116-18 (step 6), p. 148, p. 232, p. 269. This may occur, for example, when node (*c*)'s CPU has insufficient computational overhead available to devote to the cryptography operations necessary during tunnel creation. See pp. 140-41, p. 221. MorphMix discloses

this limitation at least because client device (node (*c*)) determines whether it meets the criterion (whether it has sufficient "spare resources" to support the tunnel).

As to the flow control disclosure, every node (including node (*c*)) uses flow control messaging to avoid congestion. See p. 272. Thus, client device node (*c*) uses the "credit based scheme" for flow control disclosed in MorphMix, periodically or continuously determining whether its resource utilization satisfies a criterion. For example, node (*c*) can determine whether it is able, or not able, to forward message cells, and sends, or does not send, CREDIT messages in response to that determining. See pp. 272-73.

Regarding claim 4:

The method according to claim 1, wherein the communication by the client device with the first server is based on, or is according to, TCP/IP protocol or connection.

MorphMix discloses that the client device (node (*c*)) communicates with the first server (node (*b*)) via the TCP protocol or a TCP connection. See p. 98 ('To be contacted by other nodes, a node listens on TCP port p_{mmi}.. . ."), p. 99 (MorphMix stating that "there is a TCP connection" between nodes), p. 101 (nodes *a*, *b*, and *c* "are communicating directly with each other across TCP connections"), p. 267 ("Whenever two nodes in MorphMix communicate directly with each other, they establish a virtual link. To do so, a establishes a TCP connection with *b* using ip*b* and p*mmb* ....").

As discussed above, node (*c*) establishes a virtual link with node (*b*). "Whenever two nodes in MorphMix communicate directly with each other, they establish a virtual link." MorphMix at p. 267. For example, when node (*c*) "establishes a TCP connection with b using ip_b and p_{mmb}," it does so using node (*b*)'s IP address and port. MorphMIx p. 267. "As a result, [t]here are virtual links between nodes...*b* and *c* (VL*bc*) that are communicating directly with each other across TCP connections." MorphMix at p. 101.

Regarding claim 5:

The method according to claim 4, further comprising establishing a connection with the first server, and wherein the communication with the first server is over the established connection,

and wherein the communicating by the client device with the first server uses TCP, and wherein the connection is established by performing 'Active OPEN' or 'Passive OPEN'.

The Examiner notes that the underlying '614 patent acknowledges that TCP "is one of the core protocols of the Internet protocol suite," and that "Active OPEN" and "Passive OPEN" are standard connection establishment approaches of the TCP protocol. See '614 Patent at 1:54-57.

Thus, a POSA would understand that the disclosure of the use of TCP/IP protocol, as described above, inherently discloses the standard TCP "Active OPEN" and "Passive OPEN" means of connecting.

MorphMix directly cites RFC 793, the applicable Internet Standard for the TCP protocol published in 1981, which defines these two approaches — "Active OPEN" and "Passive OPEN" — for opening and establishing a TCP connection. See MorphMix at p. 249 n.41 (citing RFC 793 at § 2.7); RFC 793 at Fig. 6 (discussing the use of TCP and the performance of "Active OPEN" and "Passive OPEN').

Regarding claim 7:

The method according to claim 1, wherein the steps are sequentially executed.

MorphMix discloses the sequential execution of claim 1's steps. In the tunnel setup example discussed above, node (*c*) must send its identifier to node (*b*) in order to establish a virtual link. Once nodes (*b*) and (*c*) are "neighbours," node (*b*) can request that node (*c*) be added to an anonymous tunnel. As part of such requests, node (*c*) will periodically or continuously determine whether its resource utilization can support a tunnel. In the event node (*c*) deems itself incapable of supporting a tunnel for one request, but thereafter capable of supporting a tunnel for a second request, there will be a shift from first state, to second state, and back to first state. Following such a shift, node (*c*) may receive a request from the first server, and thereafter perform the tasks, in sequential order, associated with fetching first content over the Internet.

Regarding claim 8:

The method according to claim 1, wherein the first identifier comprises an IP address that is in IPv4 or IPv6 form.

MorphMix discloses that the client device (node (*c*)) sends a first identifier (node (*c*)'s IP address) to the first server (node (b)) over the Internet. See pp. 98-99, 131-33, 267-68. MorphMix discloses that:

> this thesis [i.e., MorphMix] focuses on IP version 4 (IPv4) [and] MorphMix heavily depends on the IPv4 addressing structure and on the way IPv4 addresses are assigned to ISPs or institutions in general. This has directly led to our choice of using the 16-bit prefix of IP addresses as the basis for the three core components of MorphMix.
>
> MorphMix at p. 144.

Thus, node (c)'s IP address is at least in IPv4 form in the majority of the MorphMix disclosure. MorphMix also discloses a first identifier in IPv6 form. See pp. 144-47.

Regarding claims 9-10:

The method according to claim 1, wherein the client device is further storing, operating, or using, a client operating system.

The method according to claim 9, wherein the client operating system consists or, comprises of, or is based on, one out of Microsoft Windows 7, Microsoft Windows XP, Microsoft Windows 8, Microsoft Windows 8.1, Linux, and Google Chrome OS.

MorphMix discloses that "[a]nybody who has access to a computer that is connected to the Internet should be able to join and use MorphMix after having installed the MorphMix software." See p. 96. All that is required is an IP address and connectivity to the Internet. See at p. 96. [A]ny state-of-the-art personal computer," including those with modest bandwidth, "can run a MorphMix node." See at p. 142.

A POSA would understand that ordinary computers, such as those disclosed for use in MorphMix, would use an operating system. Further, when discussing the background of mix-networks (of which MorphMix is an example), MorphMix acknowledges a way to "not reveal information about the user's operating system." See p. 38. MorphMix acknowledges that an attack "could manipulate the

MorphMix software, [or] the operating system." See p. 152. Finally, MorphMix discloses that tests of its system were carried out on a typical computer that was "running Linux as operating system." See p. 129.

Thus, at the very least, MorphMix discloses the client device using a client operating system, including, specifically, a Linux operating system.

Regarding claim 13:

The method according to claim 1, further comprising executing an application, and wherein the application comprises a web browser.

MorphMix discloses a web browser application that initiates an HTTP request for web content from a web server. See pp. 96, 100 ("Three client applications C_1-C_3 (e.g., web browser windows) on the initiator's computer..."), 103 (""[W]e focus on web browsing... ."), 107, 152 ('the application (for instance a web browser) that is used to access the Internet anonymously"), 205. This web browser is the originating source of the request for web content discussed above.

Regarding claim 17:

The method according to claim 1, for use with a set threshold value, and wherein the criterion is satisfied when the resource utilization is above or below the threshold.

As discussed above, MorphMix discloses two examples of "determining" that takes place. The first example involves determining whether the utilization of tunneling resources is sufficient or insufficient to support a tunnel; the second example involves determining whether the utilization of communication resources is sufficient or insufficient to warrant issuance of CREDIT messages. Both examples involve a set threshold value, wherein satisfaction of the criterion is triggered in relation to that threshold. In the example of tunneling, the threshold value relates to the "spare resources" present, or not present, on a potential node. See p. 114, p. 148, p. 232. In their own tests, the authors of MorphMix reported CPU "computational overhead of about 6%," which was "easily" within the threshold of their own test Computer See pp. 140-41, p. 221.

In the example of flow control, the threshold value relates to a credit-based scheme—specifically, a threshold of 30 forwarded credits—tracked by each node. See pp. 272-73. Whether resource utilization is above or below each threshold determines whether each respective criterion has been satisfied.

Regarding claim 18-20 and 22-23:

18. The method according to claim 1, wherein the resource comprises, or consists of, a hardware component in the client device.

19. The method according to claim 18, wherein the hardware component comprises, or consists of, a processor or Central Processing Unit (CPU) operation in the client device.

20. The method according to claim 19, wherein the resource utilization is based on, or comprises, the processor or CPU time of executing one or more threads or processes, wherein the resource utilization is based on, or comprises, the processor or CPU idling time, or wherein the resource utilization is based on, or comprises, the processor or CPU executing a system idle process.

22. The method according to claim 1, wherein the resource comprises, or consists of, input or output capability.

23. The method according to claim 22, wherein the resource comprises, or consists of, communication bandwidth of communication with another device over the Internet.

As discussed above, regarding tunneling resources, a potential node must have "spare resources" capable of supporting a tunnel. MorphMix p. 114, p. 148, p. 232, p. 269 (a node can choose to not accept a tunnel request). While MorphMix discloses that the "bandwidth or computing power requirements" of nodes "should be modest," there is still a requirement for sufficient "spare resources" to support a tunnel. Examples of such requirements include at least "the computational overhead imposed by public-key cryptography operations." MorphMix pp. 140-41. In one test, "a computer equipped with a 1GHz AMD Athlon CPU" had "a computational overhead of about 6%" associated with such cryptographic operations. MorphMix p. 221. Thus, in this example the resource comprises or consists of a hardware

component in the client device: specifically, the CPU of node (c), which itself comprises or consists of CPU operation and the execution thereof, including, for example, execution times, idling times, and idle processes.

Regarding flow control, each node determines whether its own utilization of communication resources is sufficient or insufficient to warrant issuance of CREDIT messages to prior nodes in the communication path, thereby allowing for the passing of additional data along the path. MorphMix at pp. 272-73. Only when a node forwards enough cells forward along the communication path (30 cells) is it able to send a CREDIT message to the prior node, thereby allowing more cells to be delivered. MorphMix at pp. 272-73.

Flow control messages directly contribute to "protocol overhead," which MorphMix acknowledges can be "quite a burden for ISDN nodes and even for slower ADSL or Cable connections." MorphMix p. 286. This acknowledges two facts: first, that flow control is a way to ensure each node's bandwidth (including client device node (c)'s bandwidth) is capable of handling the communications being sent through the anonymous tunnel; second, that the flow control communication resources comprises or consists of a hardware component in the client device: specifically, the input/output hardware of node (c) that enables Internet communications, which itself comprises or consists of communication bandwidth with other nodes over the Internet.

Regarding claim 25-26:

25. The method according to claim 1, further comprising periodically sending, by the client device, a message that comprises a status of the client device, or is in response to the status of the client device.

26. The method according to claim 25, wherein the status is associated with being in the first or second state, and wherein the message is sent over the Internet to the first server.

As discussed above, tunnel setup involves, in relevant part, node (c) receiving a tunnel set-up message and "check[ing] if it is indeed willing to accept an anonymous tunnel from 5." MorphMix at pp. 116-18 (step 6). "If yes, node c responds OK and participates in key encryption exchanges to build the

tunnel. MorphMix at pp. 116-18 (steps 6-10). As part of that key encryption exchange—in response to the status of node (c) choosing to move forward with tunnel setup—client device node (c) sends its half of the encryption key-exchange to first server node (b). See pp. 116-18 (step 9). This processes periodically repeats itself as new tunnels are created ("every two minutes on average) in support of the MorphMix system. See p.114, p. 129, p. 201, p. 279.

As also discussed above, flow control involves, in relevant part, "flow control messages" that allow any node (including node (c)) to avoid congestion. MorphMix pp. 272-73. Once a node in the communication path forwards a sufficient amount of cells (30 cells) onward, "it sends a CREDIT message back" to the previous node, thereby allowing that previous node to send more cells along the path. See pp. 272-73. Client device node (c) will periodically send CREDIT messages back to first server node (b) in response to its determination that sufficient communication resources are present, largely dependent on the input/output hardware and related Internet connection of each node (affecting bandwidth, and thus, flow rates within the communication path). See pp. 208-213. Faster input/output connections will result in CREDIT messages being sent more periodically; slower connections will result in the opposite. See pp. 208-213. Both tunnel setup, and flow control, utilize messages sent from node to node, indicating messages being sent over the Internet. See p. 96 (goal of MorphMix is anonymous Internet access), 96 (Internet access required to join and use MorphMix), pp. 98-100 (nodes communicate via IP addresses, ports, and TCP connections).

Regarding claim 29:

The method according to claim 1, wherein the web server uses HyperText Transfer Protocol (HTTP) and responds to HTTP requests via the Internet, and wherein the sending of the first content identifier to the web server comprises a HTTP request, or wherein the communication with the web server is based on, or uses, HTTP persistent connection, and wherein the first content includes, consists of, or comprises, a part or whole of files, text, numbers, audio, voice, multimedia, video, images, music, or computer program, or wherein the first content includes, consists of, or comprises, a part of, or a whole of, a web-site page.

As discussed above, MorphMix discloses a system that can be used to fetch content from a web server over the Internet. See p. 96 (goal of MorphMix is anonymous Internet access), p. 103 (MorphMix's focus 1s "web browsing" and it supports HTTP and HTTPS), p. 200 (performance of MorphMix is analyzed using web browsing as the example application), p. 204 (the final node in a tunnel connects to "the web server"). MorphMix discloses that "in the case of web browsing, embedded objects in a web page are automatically requested by the browser, which results in quickly sending back data to the web server once the web browser has received the index file of a web page." See p. 107. In our ongoing example, server (s) can (1) be a web server; (2) receive requests for webpages made over HTTP; and (3) serve webpages in response to requests for the same. See p. 107, p. 204, p. 207, p. 103, p. 207, p. 107, p. 168, p. 205, p. 215.

Thus, server (s) uses HTTP, receives HTTP requests with first content identifiers (URLs), and serves content that consists of webpages. See p. 236 (regarding files). MorphMix also discloses communication between the client device node (c) and the web server is based on an HTTP persistent connection. See p. 206 ("With HTTP 1.1, the entire web page is usually downloaded over a single TCP connection"), p. 213 ('with HTTP 1.1 where the index file and all embedded objects of a page are fetched through the same anonymous connection"). MorphMix also discloses that different types of web content are fetched, including webpages, files, embedded objects, and others. See p. 205, pp. 211-213.

Conclusion

13. The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving Patent No. 10,469,614 throughout the course of this reexamination proceeding. The Patent Owner is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

14. Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that reexamination proceedings "will be conducted with special

dispatch" (37 CFR 1.550(a)). Extension of time in ex parte reexamination proceedings are provided for in 37 CFR 1.550(c).

15. All correspondence relating to this ex parte reexamination proceeding should be directed:

By EFS: registered users may submit via the electronic filing system EFS-Web, at https://efs.uspto.gov/efile/myportal/efs-registered

By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By FAX to: (571) 273-9900
Central Reexamination Unit

By hand: Customer Service Window
Attn: Central Reexamination Unit
Randolph Building, Lobby Level
401 Dulany Street
Alexandria, VA 22314

For EFS-Web transmissions, 37 CFR 1.8(a)(1)(i) (C) and (ii) states that correspondence (except for a request for reexamination and a corrected or replacement request for reexamination) will be considered timely filed if (a) it is transmitted via the Office's electronic filing system in accordance with 37 CFR 1.6(a)(4), and (b) includes a certificate of transmission for each piece of correspondence stating the data of transmission, which is prior to the expiration of the set period of time in the Office action.

Any inquiry by the patent owner concerning this communication or earlier communications from the Legal Advisor or Examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

__/Ovidio Escalante/_____
Ovidio Escalante
Reexamination Specialist
Central Reexamination Unit - Art Unit 3992
(571) 272-7537

Conferees:

/MAJID A BANANKHAH/
Reexamination Specialist, Art Unit 3992
/M.F/
Supervisory Patent Examiner, Art Unit 3992