US012393640B2

(12) **United States Patent**
Shen et al.

(10) **Patent No.: US 12,393,640 B2**
(45) **Date of Patent: *Aug. 19, 2025**

(54) **COOPERATIVE WEB BROWSING USING MULTIPLE DEVICES**

(71) Applicant: **Microsoft Technology Licensing, LLC**, Redmond, WA (US)

(72) Inventors: **Huifeng Shen**, Beijing (CN); **Shipeng Li**, Palo Alto, CA (US); **Yan Lu**, Beijing (CN); **Zhaotai Pan**, Anhui (CN); **Jianfeng Wang**, Beijing (CN)

(73) Assignee: **Microsoft Technology Licensing, LLC**, Redmond, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **18/339,678**

(22) Filed: **Jun. 22, 2023**

(65) **Prior Publication Data**

US 2023/0334112 A1 Oct. 19, 2023

Related U.S. Application Data

(63) Continuation of application No. 17/189,882, filed on Mar. 2, 2021, now Pat. No. 11,727,079, which is a (Continued)

(51) **Int. Cl.**
G06F 16/957 (2019.01)

(52) **U.S. Cl.**
CPC ***G06F 16/9577*** (2019.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,443,863 A 4/1984 Busch
5,255,361 A 10/1993 Callaway et al.
(Continued)

FOREIGN PATENT DOCUMENTS

CN 1189049 A 7/1998
CN 1694058 A 11/2005
(Continued)

OTHER PUBLICATIONS

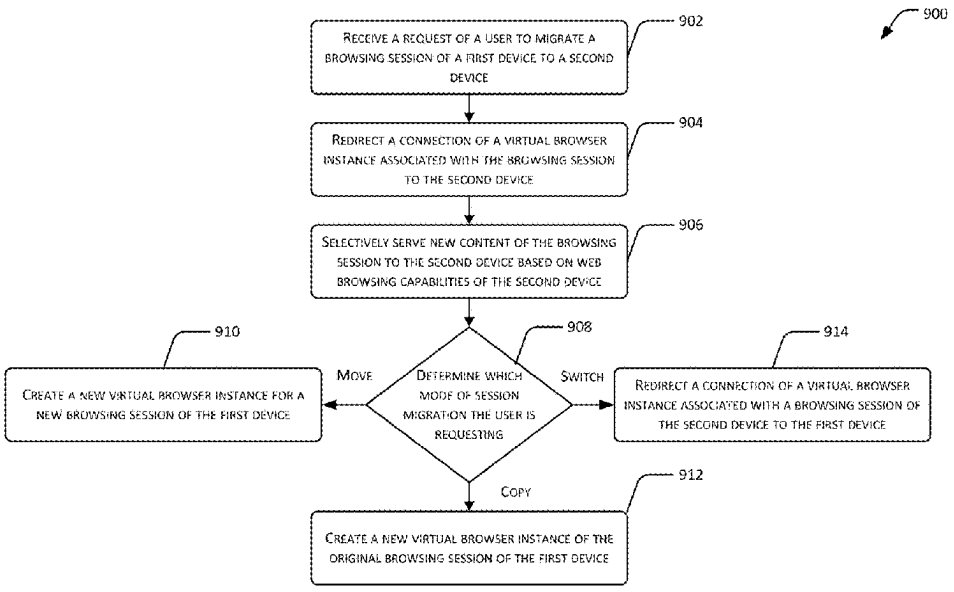
Mate, Sujeet & Chandra, Umesh & Curcio, Igor. "Movable-multimedia: Session mobility in ubiquitous computing ecosystem" 2006, Proceedings of the 5th international conference on mobile and ubiquitous multimedia, MUM 2006, pp. 1-5 (Year: 2006).*
(Continued)

Primary Examiner — Ariel Mercado-Vargas

(57) **ABSTRACT**

A proxy-based thin-client web browsing framework enables cooperative web browsing of multiple devices. The multiple devices may include devices that are not intended for web browsing and have limited or no web browsers and/or user input capabilities. The proxy-based thin client web browsing framework employs a virtual browser at a proxy server to perform all browser-engine logics, and retrieve, render and encode web pages on behalf of the multiple devices. The multiple devices therefore only need to have limited decoding and display capabilities to perform web browsing. The proxy-based thin client web browsing framework further includes a touch controller as a remote controller for a device that has no or limited user texting or manipulating capabilities.

20 Claims, 9 Drawing Sheets



Related U.S. Application Data

continuation of application No. 15/619,248, filed on Jun. 9, 2017, now Pat. No. 10,970,355, which is a continuation of application No. 13/585,185, filed on Aug. 14, 2012, now Pat. No. 9,721,036.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,913,727	A	6/1999	Ahdoot	
6,014,133	A	1/2000	Yamakado	
7,171,444	B2	1/2007	Deshpande	
7,295,208	B2	11/2007	White	
7,376,278	B2	5/2008	Iwamura	
7,430,681	B1	9/2008	Hobbs	
7,487,454	B2	2/2009	Czerwinski	
7,702,646	B2	4/2010	Rabines	
8,200,796	B1	6/2012	Margulis	
8,918,528	B2	12/2014	Gao et al.	
9,201,627	B2 *	12/2015	Tam	G06F 3/1454
2002/0097229	A1	7/2002	Rose	
2003/0229900	A1 *	12/2003	Reisman	H04N 21/422 348/E7.071
2004/0010622	A1	1/2004	O'Neill	
2004/0249980	A1	12/2004	Hutler et al.	
2005/0052427	A1	3/2005	Wu	
2005/0057524	A1	3/2005	Hill et al.	
2005/0060759	A1	3/2005	Rowe et al.	
2005/0212750	A1	9/2005	Marvit et al.	
2006/0010392	A1	1/2006	Noel et al.	
2006/0010400	A1	1/2006	Dehlin et al.	
2006/0025218	A1	2/2006	Hotta	
2006/0026536	A1	2/2006	Hotelling	
2006/0056708	A1	3/2006	Shen et al.	
2006/0161868	A1	7/2006	Van Dok et al.	
2006/0242607	A1	10/2006	Hudson	
2007/0024597	A1	2/2007	Matsuoka	
2007/0150842	A1	6/2007	Chaudhri et al.	
2007/0180492	A1	8/2007	Hassan et al.	
2007/0189603	A1	8/2007	Kasperkiewicz	
2007/0201562	A1	8/2007	Ganesh et al.	
2007/0201751	A1	8/2007	Wu	
2007/0242056	A1	10/2007	Engelhardt	
2007/0257097	A1	11/2007	Nurmela et al.	
2007/0268824	A1	11/2007	Kodaka	
2007/0273666	A1	11/2007	Shin et al.	
2007/0283292	A1	12/2007	Bucher et al.	
2008/0015058	A1	1/2008	Noble et al.	
2008/0048997	A1	2/2008	Gillespie et al.	
2008/0074431	A1	3/2008	Bakalash	
2008/0084400	A1	4/2008	Rosenberg	
2008/0094370	A1	4/2008	Ording et al.	
2008/0189613	A1	8/2008	Jong et al.	
2008/0215671	A1	9/2008	Lu	
2009/0007017	A1	1/2009	Anzures	
2009/0146951	A1	6/2009	Welland	
2009/0227369	A1	9/2009	Higbie et al.	
2010/0017722	A1	1/2010	Cohen	
2010/0064261	A1	3/2010	Andrews et al.	
2010/0074321	A1	3/2010	Beaudreau	
2010/0088061	A1	4/2010	Horodezky	
2010/0111410	A1	5/2010	Lu	
2010/0138780	A1	6/2010	Marano et al.	
2010/0169310	A1	7/2010	Latzina	
2010/0306772	A1	12/2010	Arnold et al.	
2010/0329358	A1	12/2010	Zhang et al.	
2011/0004827	A1	1/2011	Doerr	
2011/0072349	A1	3/2011	Catanese	
2011/0126197	A1	5/2011	Larsen et al.	
2011/0163972	A1	7/2011	Anzures et al.	
2011/0199389	A1	8/2011	Lu et al.	
2011/0209064	A1	8/2011	Jorgensen et al.	
2011/0219419	A1 *	9/2011	Reisman	G06F 16/954 725/112
2012/0054616	A1 *	3/2012	Mittal	G06F 16/957 715/722
2012/0064971	A1	3/2012	Devine	
2012/0124509	A1	5/2012	Matsuda	
2012/0188279	A1	7/2012	Demaine	
2012/0210241	A1 *	8/2012	Wong	H04N 21/6125 715/741
2013/0125014	A1 *	5/2013	Sharif-Ahmadi	H04N 21/44213 715/748
2013/0204857	A1	8/2013	Kartoun	

FOREIGN PATENT DOCUMENTS

JP	H11212726	A	8/1999
JP	H11345072	A	12/1999
JP	2001125727	A	5/2001
JP	2003001688	A	1/2003
JP	2006093901	A	4/2006
JP	2006099468	A	4/2006
JP	2006139615	A	6/2006
JP	2008508600	A	3/2008
JP	2008140182	A	6/2008
KR	20060008735	A	1/2006
KR	20060085850	A	7/2006
KR	100797788	B1	1/2008
WO	2004001560	A1	12/2003
WO	2006094308	A2	9/2006
WO	2007061057	A1	5/2007

OTHER PUBLICATIONS

"Exclusive: Windows Mobile 7 to Focus on Touch and Motion Gestures", Retrieved from: http://microsoft.blognewschannel.com/archives/2008/01/06/exclusive-windows-mobile-7-to-focus-on-touch-and-motion-gestures, Jul. 1, 2008, 38 Pages.

"FingerWorks iGesture Pad (igesturenumpad) Touch Pad: Product Features", Retrieved from: http://www.dealtime.com/xPF-Cat-Fing-IGESTURENUMPAD, Jul. 1, 2008, 2 Pages.

"Remote Desktop Protocol", retrieved at http://msdn.microsoft.com/en-us/library/aa383015.aspx, Oct. 23, 2008, 3 Pages.

"TCP/IP: The Best Protocol for Remote Computing", accessed from http://www.rysavy.com/Articles/remote.htm, Oct. 31, 2008, 13 Pages.

"Touch Sensor Features Gesture Recognition Technology", Retrieved at http://findarticles.com/p/articles/mi_m0PIL/is_2008_April_28/ai_n25359461, Jul. 1, 2008, 6 Pages.

"Windows Imaging Component Codec Guidelines for RAW Image Formats", Retrieved From: http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/RAWcodec_guide.doc, Nov. 6, 2006, 15 Pages.

Baratto, et al., "THINC: A virtual display architecture for thin-client computing", ACM SIGOPS Operating Systems Review, vol. 39, Issue No. 5, Oct. 20, 2005, pp. 277-290.

Baratto, et at, "MobiDesk: Mobile Virtual Desktop Computing", Proceedings of the 10th annual international conference on Mobile computing and networking, Sep. 26, 2004, 18 pages.

Communication pursuant to Rules 70(2) and 70a(2) EPC Received for European Application No. 09829652.8, mailed on Oct. 12, 2012, 01 pages.

Communication Pursuant to Rules 70(2) and 70a(2) Received in European Patent Application No. 09813532.0, mailed on Jan. 23, 2012, 1 page.

Communication under Rule 71(3) Received in European Patent Application No. 09829652.8 mailed on Jun. 8, 2016, 07 pages.

Decision to grant a European patent pursuant to Article 97(1) received in European Application No. 09829652.8, mailed on Oct. 27, 2016, 1 pages.

Dipert, B., "Instigating a platform tug of war: Graphics vendors hunger for CPU suppliers' turf", retrieved from http://www.edn.com/index.asp?layout=article&articleid=CA6262535, Oct. 13, 2005, 11 pages.

Extended Search Report received for European Application No. 09813532.0, Jan. 4, 2012, 5 pages.

Extended Search Report received for European Application No. 09829652.8, Sep. 25, 2012, 9 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Final Office Action mailed on Apr. 14, 2014, in U.S. Appl. No. 12/206,747, 30 pages.
Final Office Action mailed on Jun. 7, 2013, in U.S. Appl. No. 12/206,747, 30 pages.
Final Office Action mailed on Oct. 11, 2011, in U.S. Appl. No. 12/206,747, 24 pages.
Final Office Action mailed on Sep. 1, 2015, in U.S. Appl. No. 14/063,106, 11 pages.
First Office Action Received for Chinese Application No. 200980135963.X, mailed on Aug. 31, 2012, 9 pages (English Translation Provided).
First Office Action Received for Chinese Application No. 200980143747.X, mailed on Nov. 22, 2012, 15 pages. (English Translation Provided).
International Search Report and Written Opinion received for PCT Application No. PCT/US09/56357, Mar. 15, 2010, 10 pages.
International Search Report and Written Opinion received for PCT Application No. PCT/US2009/062863, Jun. 15, 2010, 7 pages.
Jiang, et al., "A Novel Remote Screen Synchronization Mechanism for Ubiquitous Environments", IEEE First International Symposium on Pervasive Computing and Applications, Aug. 2006, 6 pages.
Kaplinsky, Konstantin, V., "VNC Tight Encoder—Data Compression for VNC", Modern Techniques and Technology, 2001, pp. 155-157.
Laan, et al., "GPU-Accelerated Dirac video codec", accessed from http://www.cs.rug.n1/~wladimir/sc-cudat, Oct. 24, 2008, 2 Pages.
Mather, Alexis., "GPU-Accelerated Video Encoding", Retrieved from http://ati.amd.com/products/firepro/Siggraph_200B_video_encode_final.pdf, 2008, 38 pages
Montagnat, et al., "Efficient Visualization of 3D Medical Scenes for Remote Interactive Applications", Image and Signal Processing and Analysis (ISPA), Sep. 20, 2003, 6 pages.
Non-Final Office Action mailed on Apr. 19, 2013, in U.S. Appl. No. 13/094,725, 7 pages.
Non-Final Office Action mailed on Jan. 9, 2012, in U.S. Appl. No. 12/408,611, 06 Pages.
Non-Final Office Action mailed on Jun. 10, 2011, in U.S. Appl. No. 12/206,747, 21 pages.
Non-Final Office Action mailed on May 5, 2015, in US Application No. 14/063, 106, 11 pages.
Non-Final Office Action mailed on Nov. 18, 2013, in U.S. Appl. No. 12/206,747, 33 pages.
Non-Final Office Action mailed on Nov. 21, 2012, in U.S. Appl. No. 12/206,747, 30 pages.
Notice of Allowance mailed on Apr. 27, 2012, in U.S. Appl. No. 12/408,611, 07 Pages.
Notice of Allowance mailed on Dec. 30, 2015, in U.S. Appl. No. 14/063,106, 8 pages.
Notice of Allowance mailed on Jul. 19, 2013, in U.S. Appl. No. 13/094,725, 9 pages.
Notice of Allowance mailed on Oct. 7, 2015, in U.S. Appl. No. 13/663,720, 05 pages.
Notice of Allowance Received for Chinese Application No. 200980143747.X, mailed on May 6, 2014, 04 pages. (English Translation Provided).
Notice of Rejection Received for Japanese Application No. 2011-526938, mailed on Nov. 21, 2013, 4 pages. (English Translation Provided).
Office Action Received for Korean Application No. 10-2011-7005120, mailed on Aug. 14, 2015, 5 pages. (English Translation Provided).
Richardson, et al., "Virtual Network Computing", IEEE Internet Computing, vol. 2, No. 1, Feb. 1998, pp. 32-38.
Scheifler, et al., "The X Window System", ACM Transactions on Graphics (TOG), vol. 5, No. 2, Apr. 1986, pp. 79-109.
Second Office Action Received for Chinese Application No. 200980143747.X, mailed on Jun. 20, 2013, 07 pages. (English Translation Provided).
Third Office Action Received for Chinese Application No. 200980143747.X, mailed on Nov. 8, 2013, 11 pages. (English Translation Provided).
Thrane et al., "A comparison of acceleration structures for GPU assisted ray tracing", Retrieved From: http://larsole.com/files/GPU_BVHthesis.pdf, Aug. 1, 2005, 108 Pages.
Yang, et al., "A Fast and Efficient Codec for Multimedia Applications in Wireless Thin-client Computing", International Symposium on a World of Wireless, Mobile and Multimedia Networks. IEEE, Jun. 2007, 12 Pages.
Yangi, "The performance of remote display mechanisms for thin-client computing,", USENIX Annual Technical Conference, General Track, Jun. 2002, 15 Pages.
Zhang, et al., "Implementing remote display on commodity operating systems.", Columbia University, Thesis, Jan. 2006, 52 Pages.
Zong, et al., "A QT Interval Detection Algorithm Based on ECG Curve Length Transform", Computers in Cardiology. IEEE, 2006, pp. 377-380.
First Examination Report Received for Indian Application No. 01096/CHENP/2011, mailed on Aug. 20, 2018, 06 Pages.

* cited by examiner

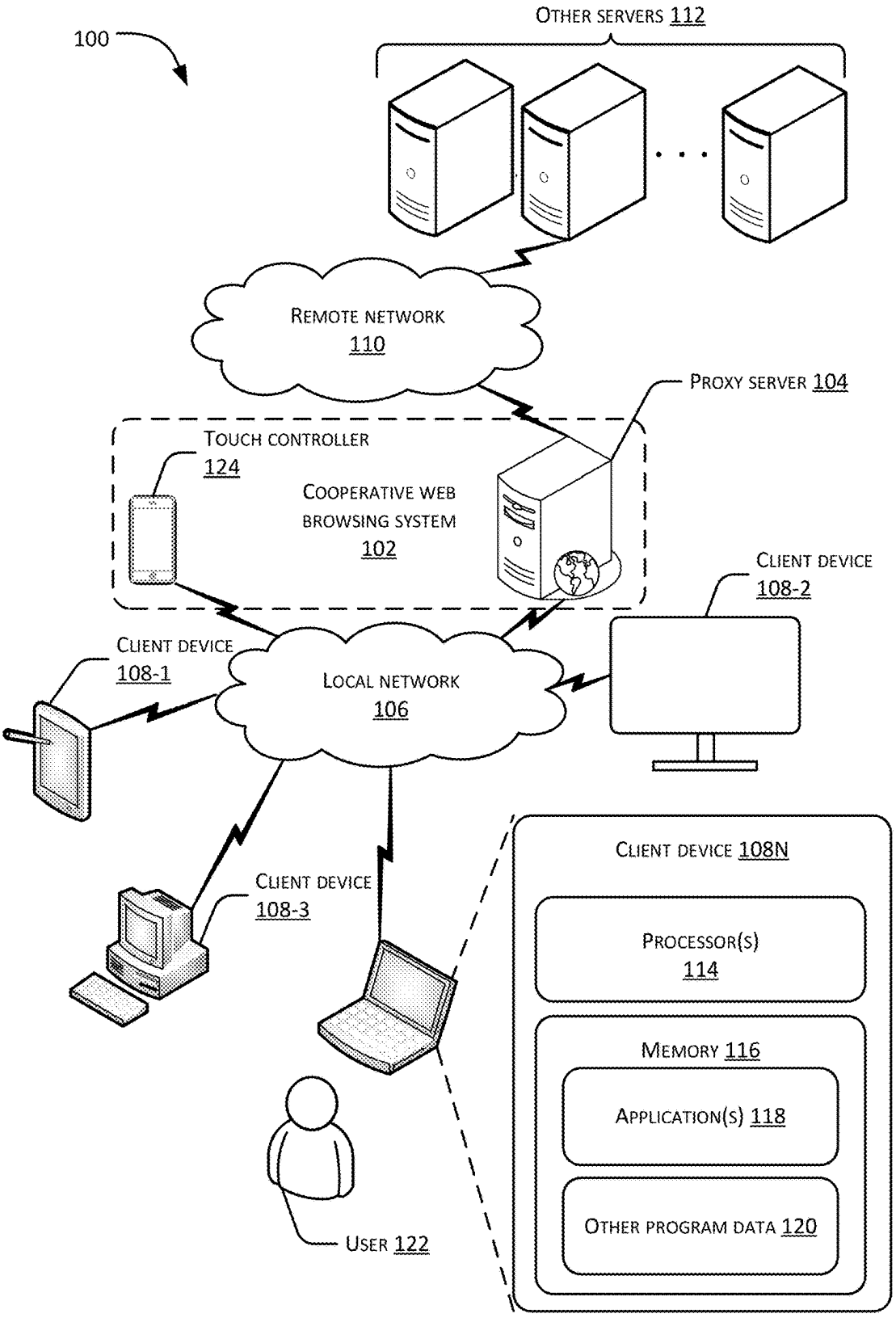

100
OTHER SERVERS 112
REMOTE NETWORK 110
PROXY SERVER 104
TOUCH CONTROLLER 124
COOPERATIVE WEB BROWSING SYSTEM 102
CLIENT DEVICE 108-2
CLIENT DEVICE 108-1
LOCAL NETWORK 106
CLIENT DEVICE 108-3
CLIENT DEVICE 108N
PROCESSOR(S) 114
MEMORY 116
APPLICATION(S) 118
OTHER PROGRAM DATA 120
USER 122

FIG. 1

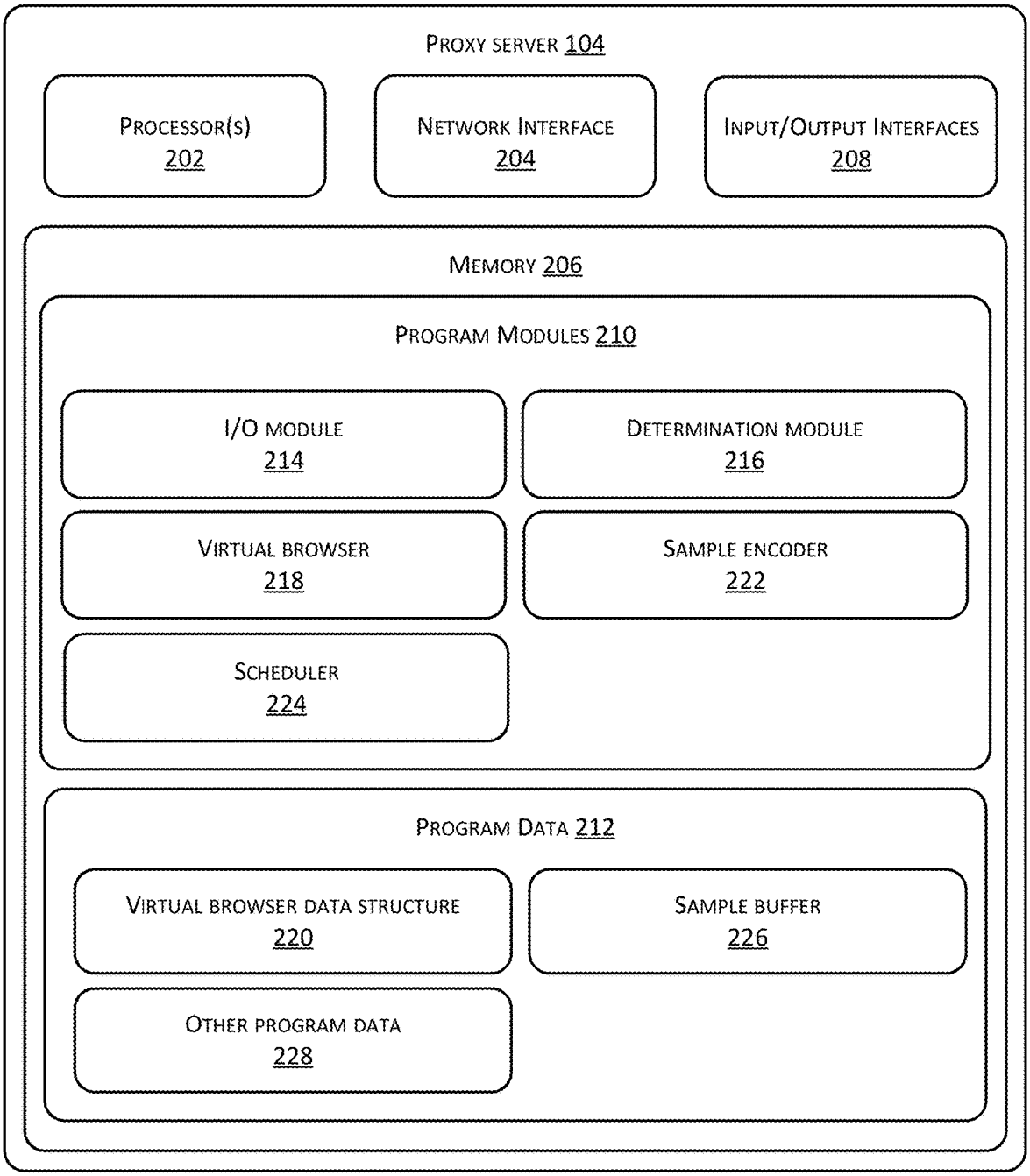

PROXY SERVER 104
PROCESSOR(S) 202
NETWORK INTERFACE 204
INPUT/OUTPUT INTERFACES 208
MEMORY 206
PROGRAM MODULES 210
I/O MODULE 214
DETERMINATION MODULE 216
VIRTUAL BROWSER 218
SAMPLE ENCODER 222
SCHEDULER 224
PROGRAM DATA 212
VIRTUAL BROWSER DATA STRUCTURE 220
SAMPLE BUFFER 226
OTHER PROGRAM DATA 228

FIG. 2

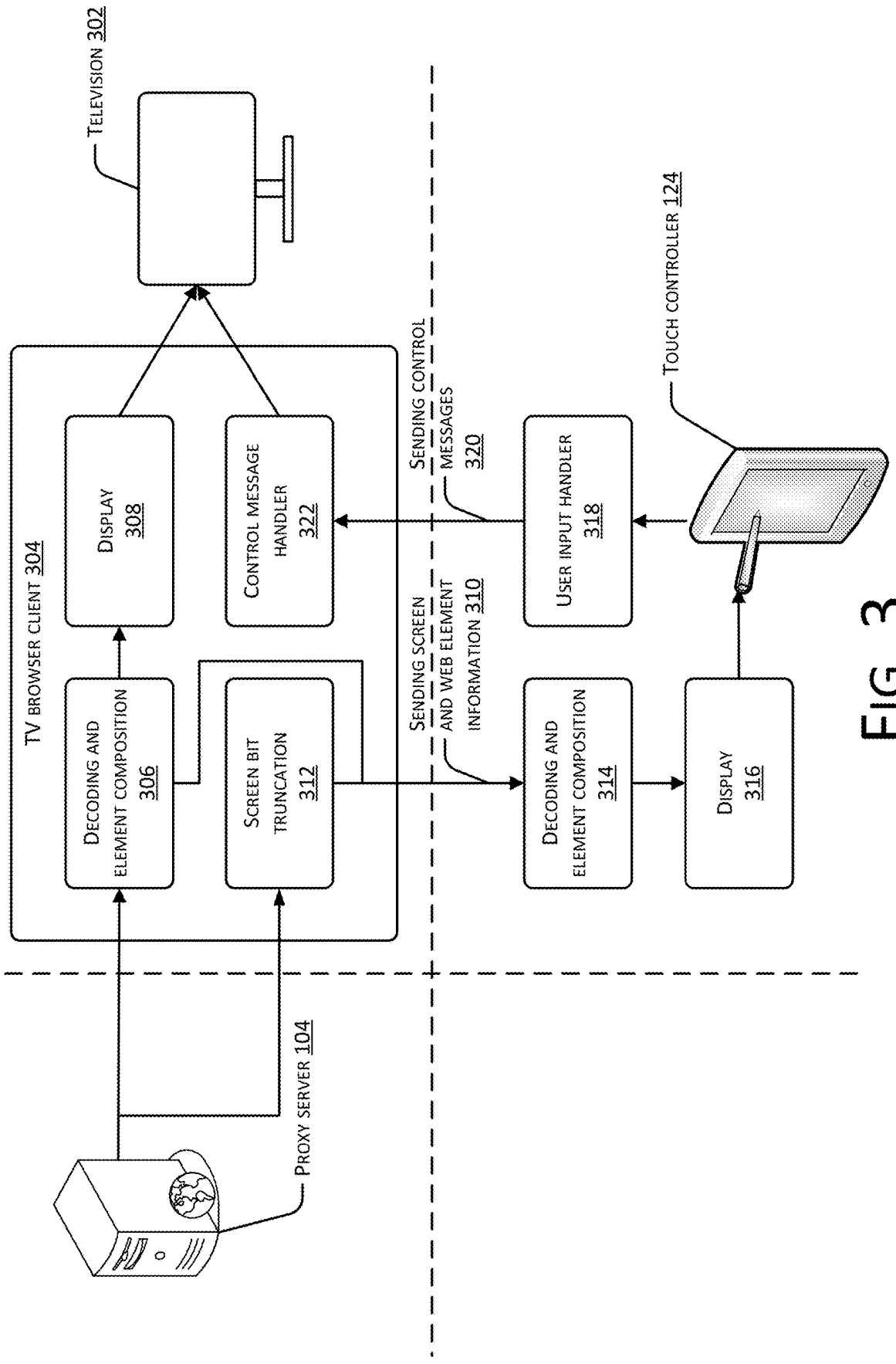

TELEVISION 302
TV BROWSER CLIENT 304
DISPLAY 308
CONTROL MESSAGE HANDLER 322
DECODING AND ELEMENT COMPOSITION 306
SCREEN BIT TRUNCATION 312
SENDING CONTROL MESSAGES 320
SENDING SCREEN AND WEB ELEMENT INFORMATION 310
USER INPUT HANDLER 318
TOUCH CONTROLLER 124
DECODING AND ELEMENT COMPOSITION 314
DISPLAY 316
PROXY SERVER 104

FIG. 3

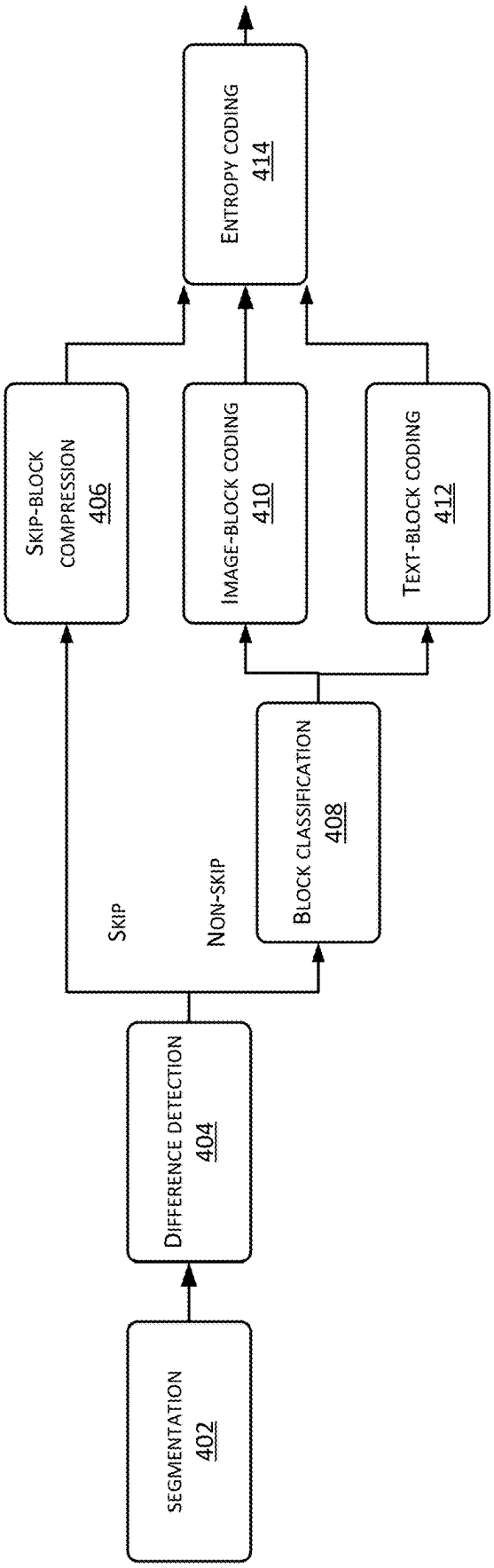

SEGMENTATION
402
DIFFERENCE DETECTION
404
SKIP
NON-SKIP
BLOCK CLASSIFICATION
408
SKIP-BLOCK COMPRESSION
406
IMAGE-BLOCK CODING
410
TEXT-BLOCK CODING
412
ENTROPY CODING
414

FIG. 4

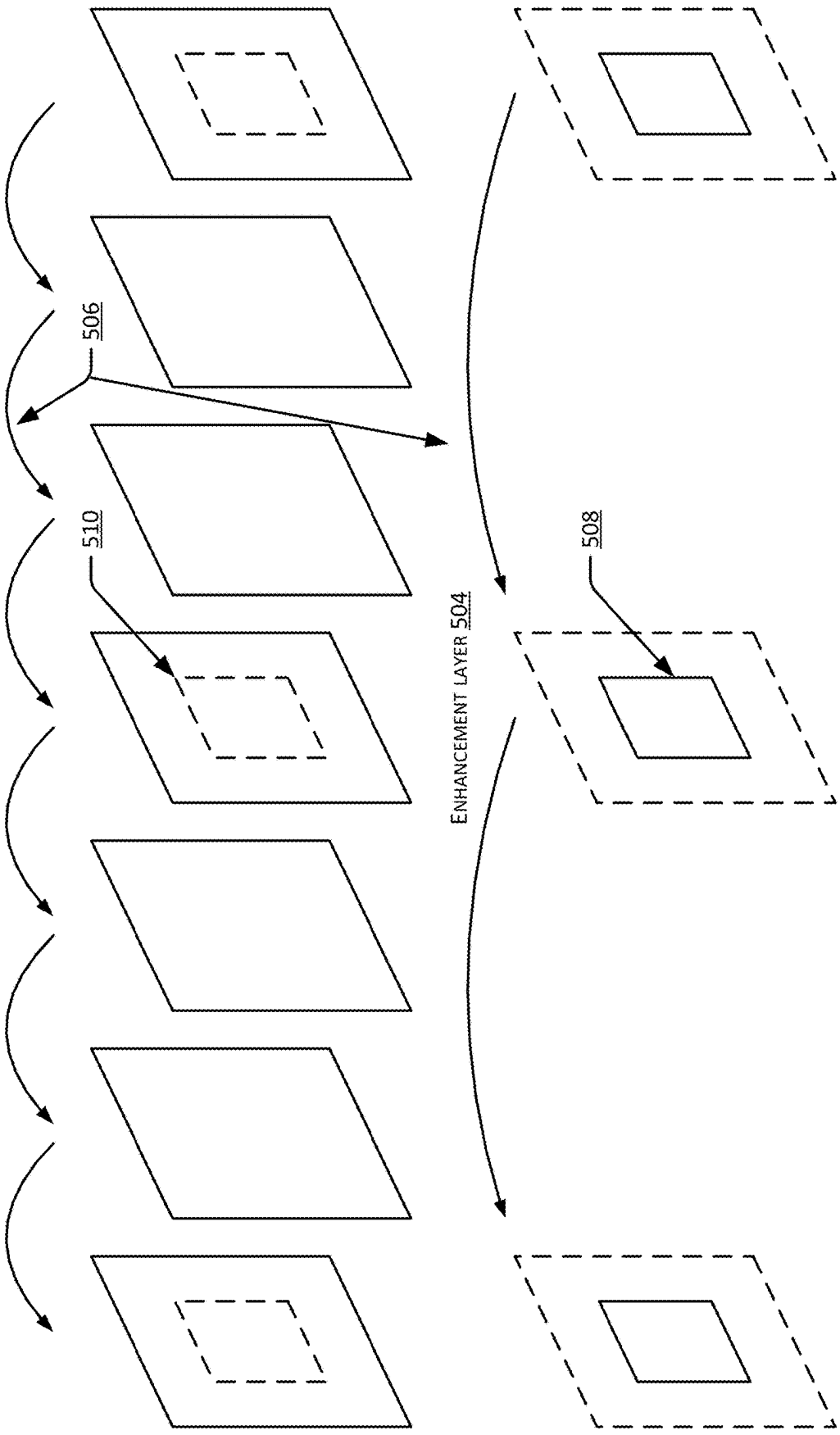


FIG. 5

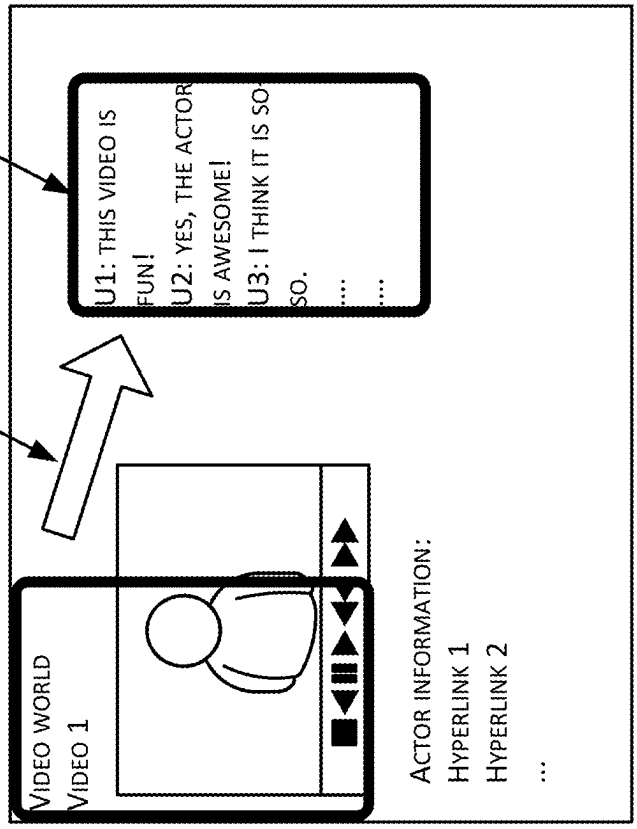

124
602
VIDEO WORLD
VIDEO 1
U1: THIS VIDEO IS FUN!
U2: YES, THE ACTOR IS AWESOME!
U3: I THINK IT IS SO-SO.
....
....
ACTOR INFORMATION:
HYPERLINK 1
HYPERLINK 2
...

FIG. 6B

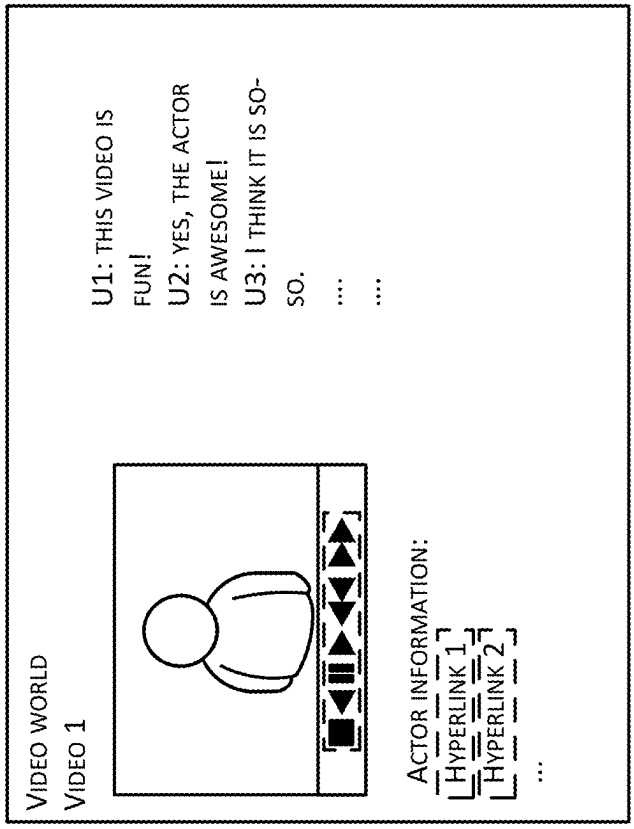

VIDEO WORLD
VIDEO 1
U1: THIS VIDEO IS FUN!
U2: YES, THE ACTOR IS AWESOME!
U3: I THINK IT IS SO-SO.
....
....
ACTOR INFORMATION:
HYPERLINK 1
HYPERLINK 2
...

FIG. 6A

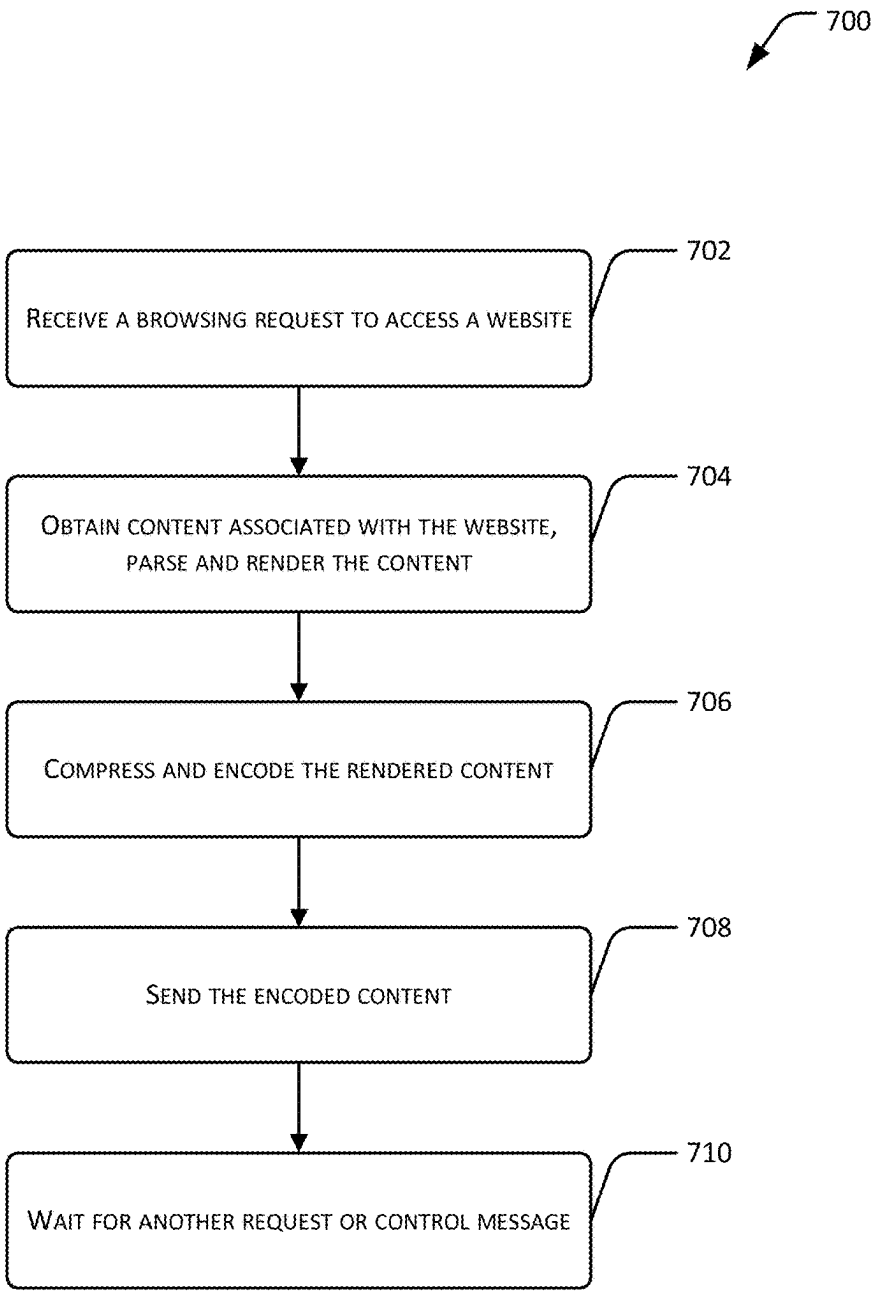

700
RECEIVE A BROWSING REQUEST TO ACCESS A WEBSITE
702
OBTAIN CONTENT ASSOCIATED WITH THE WEBSITE, PARSE AND RENDER THE CONTENT
704
COMPRESS AND ENCODE THE RENDERED CONTENT
706
SEND THE ENCODED CONTENT
708
WAIT FOR ANOTHER REQUEST OR CONTROL MESSAGE
710

FIG. 7

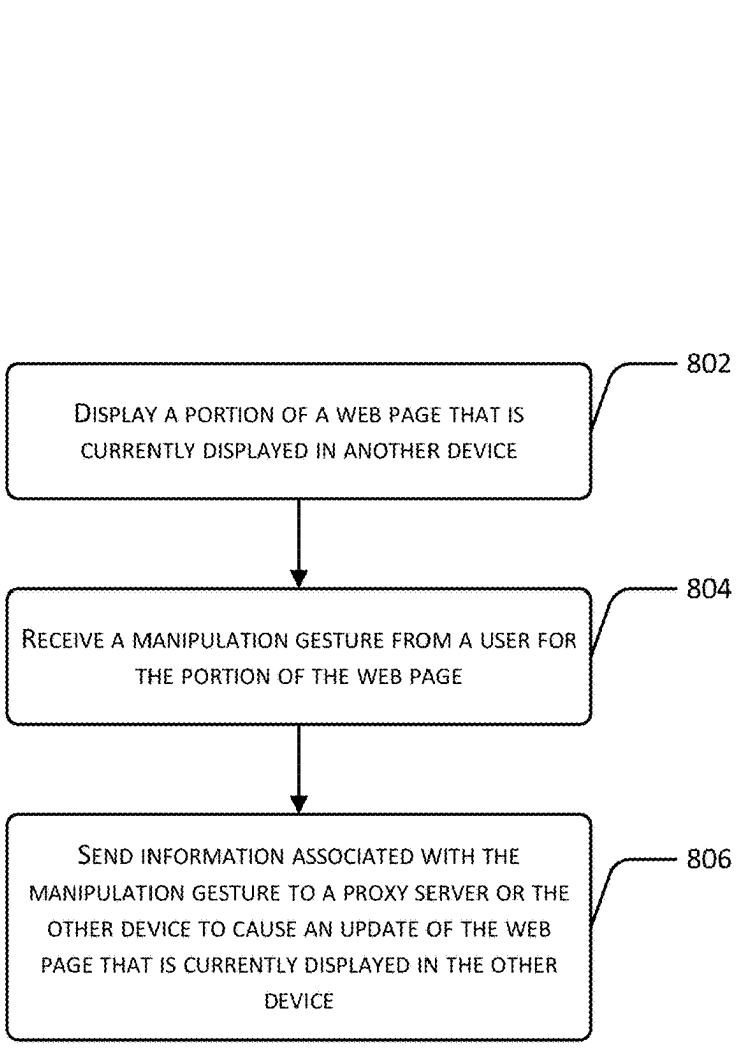

800
DISPLAY A PORTION OF A WEB PAGE THAT IS CURRENTLY DISPLAYED IN ANOTHER DEVICE
802
RECEIVE A MANIPULATION GESTURE FROM A USER FOR THE PORTION OF THE WEB PAGE
804
SEND INFORMATION ASSOCIATED WITH THE MANIPULATION GESTURE TO A PROXY SERVER OR THE OTHER DEVICE TO CAUSE AN UPDATE OF THE WEB PAGE THAT IS CURRENTLY DISPLAYED IN THE OTHER DEVICE
806

FIG. 8

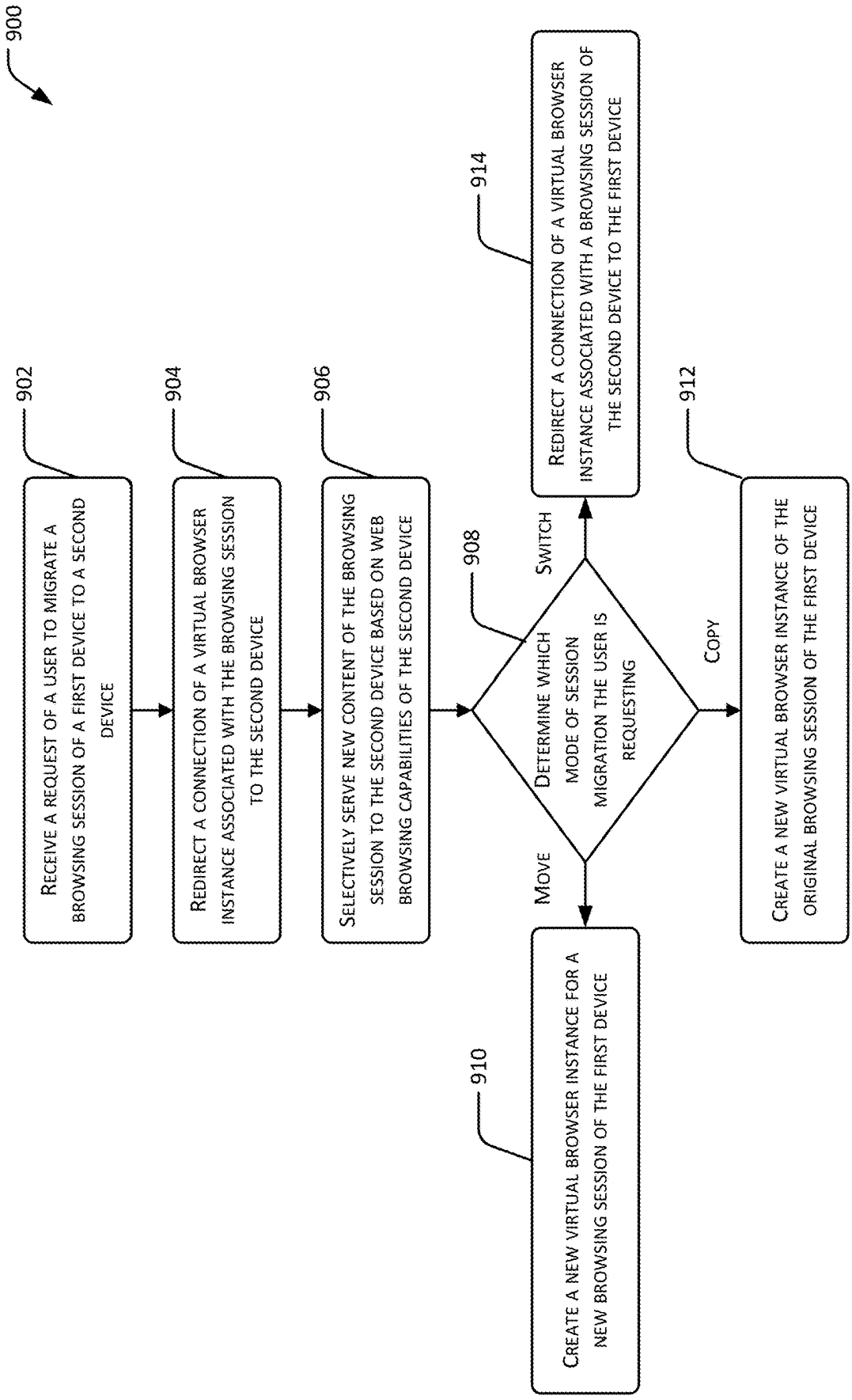

900
902
RECEIVE A REQUEST OF A USER TO MIGRATE A BROWSING SESSION OF A FIRST DEVICE TO A SECOND DEVICE
904
REDIRECT A CONNECTION OF A VIRTUAL BROWSER INSTANCE ASSOCIATED WITH THE BROWSING SESSION TO THE SECOND DEVICE
906
SELECTIVELY SERVE NEW CONTENT OF THE BROWSING SESSION TO THE SECOND DEVICE BASED ON WEB BROWSING CAPABILITIES OF THE SECOND DEVICE
908
DETERMINE WHICH MODE OF SESSION MIGRATION THE USER IS REQUESTING
SWITCH
MOVE
COPY
910
CREATE A NEW VIRTUAL BROWSER INSTANCE FOR A NEW BROWSING SESSION OF THE FIRST DEVICE
912
CREATE A NEW VIRTUAL BROWSER INSTANCE OF THE ORIGINAL BROWSING SESSION OF THE FIRST DEVICE
914
REDIRECT A CONNECTION OF A VIRTUAL BROWSER INSTANCE ASSOCIATED WITH A BROWSING SESSION OF THE SECOND DEVICE TO THE FIRST DEVICE

FIG. 9

COOPERATIVE WEB BROWSING USING MULTIPLE DEVICES

RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 17/189,882 filed Mar. 2, 2021, now Issued U.S. Pat. No. 11,727,079, which is a continuation of U.S. patent application Ser. No. 15/619,248, filed on Jun. 9, 2017, now Issued U.S. Pat. No. 10,970,355, which is a continuation of U.S. Patent Application No. 13/585,185, filed on Aug. 14, 2012, and now issued as U.S. Pat. No. 9,721,036, the contents of which are incorporated by reference in their entirety herein.

BACKGROUND

Internet browsing or surfing has become popular for home entertainment. Traditionally, web browsing can only be done on a personal computer (PC). With the advance of mobile technology, surfing the Internet using mobile devices, such as mobile phones and tablets, is feasible and becomes prevalent. However, since the mobile devices usually have very limited display sizes and computing power, etc., mobile devices are unsuitable for Internet surfing for a long period of time. Personal computers and televisions, though equipped with better displays and audio speakers, are not portable as flexibly as the mobile devices. Furthermore, the televisions, even web-based smart televisions, normally do not provide a functionality of text inputting that is frequently used in web browsing. In other words, web browsing using a single device unavoidably suffers from limited capabilities provided by that single device, forcing a user to sacrifice one browsing quality factor (e.g., mobility, etc.) in exchange for another browsing quality factor (e.g., video display quality, etc.).

SUMMARY

This summary introduces simplified concepts of cooperative web browsing of multiple devices, which are further described below in the Detailed Description. This summary is not intended to identify essential features of the claimed subject matter, nor is it intended for use in limiting the scope of the claimed subject matter.

This application describes example embodiments of cooperative web browsing of multiple client devices. In one embodiment, a proxy server may receive a browsing request from a first client device of a plurality of client devices of a user to access a website. In one embodiment, the proxy server may be a server installed in a cloud computing architecture. In some embodiments, the proxy server may be installed in a computing device that is local or in proximity to the plurality of client devices of the user. In one embodiment, the proxy server may be independent of the website, i.e., not affiliated with or part of the website, for example. In response to receiving the browsing request, the proxy server may obtain content associated with the website, for example, a web page, for the user. The proxy server may parse and render the content on behalf of the plurality of client devices of the user. In one embodiment, upon parsing and rendering the content associated with the website, the proxy server may encode and selectively provide portions of the rendered content to the plurality of client devices for consumption by the user.

In some embodiments, the proxy server may receive a request to transfer a browsing session from a first client device to a second client device. The request may include, but is not limited to, a move request, a copy request or a switch request. In response to receiving the request, the proxy server may redirect a connection of an instance of a virtual browser associated with the browsing session of the first client device at the proxy server from the first client device to the second client device. The proxy server may then selectively serve new content of the browsing session to the second device based on one or more capabilities of the second client device.

BRIEF DESCRIPTION OF THE DRAWINGS

The detailed description is set forth with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items.

FIG. **1** illustrates an example environment usable to implement an example cooperative web browsing system.

FIG. **2** illustrates an example proxy server included in the example cooperative web browsing system of FIG. **1**.

FIG. **3** illustrates an example architecture usable to implement an example touch controller included in the example cooperative web browsing system of FIG. **1**.

FIG. **4** illustrates an example single-layer encoding algorithm.

FIG. **5** illustrates an example scalable encoding framework.

FIGS. **6**A and **6**B illustrate an example scenario showing how content displayed in the example touch controller of FIG. **3** changes in response to receiving a user input.

FIG. **7** illustrates an example method of proxy-based cooperative web browsing using multiple devices.

FIG. **8** illustrates an example method of using the example touch controller in cooperative web browsing.

FIG. **9** illustrates an example method of session migration.

DETAILED DESCRIPTION

Overview

As noted above, web browsing using any single type of devices unavoidably suffers from limited capabilities of respective type of devices that a user normally needs to sacrifice one browsing quality factor (e.g., mobility, etc.) in exchange for another browsing quality factor (e.g., video display quality, etc.).

This disclosure describes a cooperative web browsing system, which facilitates cooperative web browsing using multiple devices of a user. In one embodiments, the multiple devices of the user may include devices having different or same web browsing capabilities. By way of example and not limitation, the devices of different or same web browsing capabilities may include, but are not limited to, devices that include or are associated with web browsers and/or physical or soft keyboards (e.g., computers, tablets, smartphones, etc.). Additionally, the devices of different or same web browsing capabilities may include, for example, devices that include or are associated with limited or no web browsers and/or limited or no physical or soft keyboards (e.g., televisions, etc.).

In one embodiment, the cooperative web browsing system may include a proxy server. The proxy server provides a centralized thin-client browser framework that enables interactions among the plurality of devices and facilitates coop-

erative web browsing of the plurality of devices regardless of whether one or more of the plurality of devices having limited or no capabilities of web browsing and/or user input functionalities (such as text inputting, moving and/or zooming of a web page, etc.) for web browsing. In one embodiment, the proxy server may maintain a virtual web browser. The virtual web browser may execute all browser-engine logics and maintain state information of all active browsing sessions associated with the plurality of devices of the user. For example, the virtual web browser may be configured to parse and render web pages on behalf of the plurality of devices of the user. After parsing and rendering the web pages, the proxy server may encode the rendered images of the web pages and stream the rendered images to the plurality of devices. The plurality of devices of the user may therefore need only capabilities of decoding the encoded images and displaying decoded images of the web pages to the user.

In some embodiments, the cooperative web browsing system may further include a touch controller as a remote controller for one or more of the plurality of devices of the user. In one embodiment, the touch controller may include a mobile device (such as a smartphone, a tablet, etc.) installed with a touch controller client. The cooperative web browsing system may employ the touch controller as a remote controller for other devices that have limited user input capabilities, for example, devices having limited or no capabilities of allowing the user to input text in a textbox in a web page, moving or zooming a certain part of the web page displayed on respective displays or screens, etc. In one embodiment, the proxy server may serve interactive portions (such as selectable and/or clickable controls, links, etc.) of a web page to the touch controller. Additionally or alternatively, the proxy server may serve a portion less than all of the web page that is currently displayed in another device. The proxy may allow a user to manipulate content of the web page to be displayed in the other device (e.g., the television, etc.) by manipulating the interactive portions (or data included in the portion less than all) of the web page displayed on a display of the touch controller.

In one embodiment, the proxy server may further enable seamless migration of a browsing session from one device to another device. The proxy server may automatically migrate a browsing session from a first device to a second device in response to detecting that the second device is present in a neighborhood of the first device and determining that the second device has a better capability of serving content that is currently served by the first device. Additionally or alternatively, the proxy server may migrate a browsing session from a first device to a second device in response to receiving a migration request from the user through the first device or the second device. In one embodiment, the migration request may include a move request, a copy request and/or a switch request, etc. The proxy server may perform this migration of the browsing session by redirecting a connection of an instance of the virtual web browser at the proxy server from the first device to the second device and serving content of the web page rendered by that instance of the virtual web browser to the second device through the redirected connection. Since the content of the web page has been rendered by the virtual web browser and states of browsing sessions are maintained in the virtual web browser, the second device can simply and quickly display the rendered content to the user without obtaining and knowing information of past web browsing in the first device.

In some embodiments, the cooperative web browsing system may allow the user to provide a voice input or command through the touch controller, for example. The user may perform the aforementioned operations using a voice input or command in cooperation with or in alternative to a touch input (e.g., through a finger, a digital pen, a stylus, etc.) on the touch controller.

The described system provides a centralized thin-client framework for a plurality of client devices of a user, and moves parsing and rendering of a web page from a client side to a proxy side, thereby alleviating computing burden of the client devices.

While in the examples described herein, the cooperative web browsing system receives a browsing request from one of a plurality of devices, obtains content of a web page, parses and renders the content of the web page, encodes the rendered content, and streams the encoded content to the plurality of devices of the user, in other embodiments, these functions may be performed by multiple separate systems or services in a cloud computing architecture and/or multiple devices. For example, in one embodiment, a request service may receive a request including a browsing request and/or a migration request, while a separate service may obtain content of the web page in response to receiving the browsing request, and yet another service may perform migration of a browsing session from one device to another. A parsing and rendering service may parse and render the content of the web page, while an encoding service may encode the rendered content, and stream the encoded content to the plurality of client devices of the user.

The application describes multiple and varied implementations and embodiments. The following section describes an example environment that is suitable for practicing various implementations. Next, the application describes example systems, devices, and processes for implementing a cooperative web browsing system.

Exemplary Environment

FIG. **1** illustrates an exemplary environment **100** usable to implement a cooperative web browsing system **102** including a proxy server **104**. In some embodiments, the environment **100** may include a local network **106** and a plurality of client devices **108-1**, **108-2**, **108-3**, . . . , **108**-N (collectively referred to as client device **108**). The plurality of client devices **108** may communicate data with the proxy server **104** via the local network **106**. In one embodiment, the plurality of client devices **108** may further communicate data with each other via the local network **106** and/or the proxy server **104**. In some embodiments, the environment **100** may further include a remote network **110** and other servers **112**. In one embodiment, the plurality of client devices **108** may connect to the remote network **110** and the other servers **112** via the proxy server **104** when engaged in cooperative web browsing. Additionally or alternatively, in some embodiments, one or more of the plurality of client devices **108** may directly communicate data with the other server **112** via the remote network **110**.

Although the proxy server **104** is described to be separate from the plurality of client devices **108**, in some embodiments, functions of the proxy server **104** may be included and distributed among one or more client devices **108**. For example, one of the client devices **108** may include part of the functions of the proxy server **104** while other functions of the proxy server **104** may be included in one or more other client devices **108**. Furthermore, in some embodiments, the proxy server **104** may be included in a third-party server,

such as one or more of the other servers **112**. In one embodiment, the proxy server **104** may be part of a cloud computing architecture.

The client device **108**-N, representative of the client devices **108**, may be implemented as any of a variety of conventional consumer or computing devices including, for example, a television, a setup box, a mainframe computer, a server, a notebook or portable computer, a handheld device, a netbook, an Internet appliance, a tablet or slate computer, a mobile device (e.g., a mobile phone, a personal digital assistant, a smart phone, etc.), etc. or a combination thereof.

The local network **106** may be a wireless or a wired network, or a combination thereof. The local network **106** may be a collection of individual networks interconnected with each other and functioning as a single large network (e.g., the Internet or an intranet). Examples of such individual networks include, but are not limited to, Local Area Networks (LANs), WiFi network, etc. Further, the individual networks may be wireless or wired networks, or a combination thereof.

The remote network **110** may be a wireless or a wired network, or a combination thereof. The remote network **110** may be a collection of individual networks interconnected with each other and functioning as a single large network (e.g., the Internet or an intranet). Examples of such individual networks include, but are not limited to, telephone networks, cable networks, Local Area Networks (LANs), Wide Area Networks (WANs), and Metropolitan Area Networks (MANS). Further, the individual networks may be wireless or wired networks, or a combination thereof.

In one embodiment, some or all of the client devices **108** may include one or more processors **114** coupled to memory **116**. The memory **116** includes one or more applications or services **118** (e.g., web browser, etc.) and other program data **120**. The memory **116** may be coupled to, associated with, and/or accessible to other devices, such as network servers, routers, and/or the client devices **108**. Additionally or alternatively, in some embodiments, some of the client devices **108** (such as television, for example) may include no processor and/or memory as of a computing device.

A user **122** of the plurality of client devices **108** may want to cooperatively browse or access one or more websites or web pages using the plurality of client devices **108**. The user **122** may send a request of browsing a web page using a first device **108** (e.g., a mobile device **108-1**) to the proxy server **104**.

In one embodiment, the cooperative web browsing system **102** may further include a touch controller **124**. The touch controller **124** may be included in one or more of the plurality of client devices **108** that include a touch screen or display and has been installed with a touch controller client for the cooperative web browsing system **102**. The user **122** may use the touch controller **124** to browse the website or the web page that is displayed in another client device **108** (such as television, for example) that includes no web browser and/or has limited user input capability (such as physical or soft keyboard, mouse, etc.) for normal web browsing operations (e.g., inputting a text, selecting a link, actuating a control, moving up or down, zooming in or out, etc.). Upon receiving the browsing request, the proxy server **104** may obtain content of the requested web page, parse and render the web page, and selectively send portions of the web page to the first device **108** and one or more other devices **108** of the user for cooperative web browsing.

In some embodiments, the cooperative web browsing system **102** may allow the user to provide a voice input or command through the touch controller **124**, for example. The user may perform the aforementioned operations using a voice input or command in cooperation with or in alternative to a touch input (e.g., through a finger, a digital pen, a stylus, etc.) on the touch controller **124**.

FIG. **2** illustrates the example proxy server **104** in more detail. In one embodiment, the proxy server **104** includes, but is not limited to, one or more processors **202**, a network interface **204**, memory **206**, and an input/output interface **208**. The processor(s) **202** is configured to execute instructions received from the network interface **204**, received from the input/output interface **208**, and/or stored in the memory **206**.

The memory **206** may include computer-readable media in the form of volatile memory, such as Random Access Memory (RAM) and/or non-volatile memory, such as read only memory (ROM) or flash RAM. The memory **206** is an example of computer-readable media. Computer-readable media includes at least two types of computer-readable media, namely computer storage media and communications media.

Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, phase change memory (PRAM), static random-access memory (SRAM), dynamic random-access memory (DRAM), other types of random-access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disk read-only memory (CD-ROM), digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other non-transmission medium that can be used to store information for access by a computing device.

In contrast, communication media may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave, or other transmission mechanism. As defined herein, computer storage media does not include communication media.

The memory **206** may include program modules **210** and program data **212**. The proxy server **104** provides a centralized thin-client framework for supporting cooperative web browsing of the plurality of devices **106** of the user **122**. In one embodiment, the proxy server **104** may include an I/O module **214**. The I/O module **214** is configured to receive requests, instructions and/or control messages from one or more client devices **108** of the user **122**. By way of example and not limitation, the I/O module **214** may receive a connection request from a client device **108** (e.g., the client device **108-1**) of the user **122** for connecting to the proxy server **104**.

In response to receiving the connection request from the client device **108**, a determination module **216** of the proxy server **104** may determine whether the client device **108** is allowed or authorized to connect and/or access the cooperative web browsing system **102** and/or the proxy server **104**. By way of example and not limitation, the determination module **216** may determine whether a client device **108** is allowed to connect and/or access the cooperative web browsing system **102** and/or the proxy server **104** based on, for example, a device identifier and/or an IP (Internet address) of the client device **108**, etc.

Additionally or alternatively, the determination module **216** may determine whether the user **122** is allowed or authorized to use or access the cooperative web browsing

system **102** and/or the proxy server **104** using the client device **108**. For example, the determination module **216** may prompt the user **122** to provide a password when the user **122** initially tries to access a website or web page through the cooperative web browsing system **102** or the proxy server **104**. Additionally or alternatively, the determination module **216** may identify or determine which one or more of the plurality of client devices **108** that the user **122** may be allowed to use during cooperative web browsing.

Additionally or alternatively, in some embodiments, the determination module **216** may further obtain information of the client device **108**. In one embodiment, the determination module **216** may obtain the information of the client device **108** based on the device identifier of the client device **108**. Additionally or alternatively, the determination module **216** may obtain the information of the client device **108** manually from the user **122**. In some embodiments, the determination module **216** may obtain information of the client device **108** related to web browsing capabilities, such as display resolution or quality, audio capability (and/or quality), computing (or processing) power, text inputting capabilities (e.g., whether including a physical or soft keyboard, etc.), whether including a touch screen, etc. The proxy server **104** may use this information of the client device **108** to determine what type of content (e.g., which portions of a web page) may be provided to that client device **108**.

In response to successfully authenticating the client device **108** and/or the user **122**, the proxy server **104** may create an instance of a virtual browser **218** for the client device **108**. Additionally, in some embodiments, the proxy server **104** may further notify other client devices **108** of the user **122** which have been authenticated and currently connected to the cooperative web browsing system **102** that one new client device **108** has joined to the cooperative web browsing system **102** and is available for use and/or migration thereto or therefrom.

In one embodiment, the virtual browser **218** may retrieve, parse and render web pages on behalf of the client device **108** (and/or one or more other client devices **108**). In one embodiment, the virtual browser **218** may render the web pages into bitmaps (or display bitmaps) as synthesized by a computing device. In this disclosure, "screen" is used hereinafter to refer to the display bitmap synthesized by the computing device. In one embodiment, the proxy server **104** may store the instance of the virtual browser **218** in a virtual browser data structure **220** corresponding to the user **122**. In some embodiments, the virtual browser data structure **220** may include, for example, an array, a list, etc. In one embodiment, a separate virtual browser data structure **220** is used for each user.

In some embodiments, the virtual browser **218** may further extract audio, and web-element side information from the web pages. In one embodiment, web elements may include, but are not limited to, clickable and/or selectable elements (such as hyperlinks, controls, text, etc.) on a web page. The web-element side information may include, but is not limited to, types, positions and visibility properties of the web elements.

In one embodiment, the proxy server **104** may further include a sample encoder **222**. The sample encoder **222** may receive data of screen, audio and web elements that has been extracted by the virtual browser **218**. Upon receiving the extracted data, the sample encoder **222** may encode different types of data using different encoding mechanisms. By way of example and not limitation, the sample encoder **222** may employ a compression algorithm to compress a web screen as synthesized by the virtual browser **218**. The sample encoder **222** may further adaptively identify text blocks and image blocks from screen images and use different encoding algorithms for text and image contents. Details of example compression and encoding algorithms that may be used by the sample encoder **222** will be described in further detail hereinafter.

Upon encoding the extracted data from the virtual browser **218**, the proxy server **104** may further include a scheduler **224** that adaptively push the encoded data or samples (e.g., screen samples, audio samples and web element samples) to a sample buffer **226** based on a control message received from the client device **108**. For example, the client device **108** may send a control message indicating a selection of a web element such as a link on the web page. In response to receiving the control message, the scheduler **224** may push encoded samples related to the selected web element to the sample buffer **226**.

In one embodiment, the I/O module **214** of the proxy server **104** may retrieve the encoded samples from the sample buffer **226** and sends the encoded samples to one or more of the plurality of client devices **108**.

In one embodiment, in response to receiving the encoded samples from the proxy server **104** or the I/O module **214**, the client device **108** may decode the encoded samples, and composite the web screen, audio and/or web elements spatially and temporally together for presentation to the user **122**. In some embodiments, the client device **108** may send one or more control messages to the proxy server **104**. The one or more control messages may include, but are not limited to, user input messages, current visible client-region information, network conditions, etc.

In some embodiments, the proxy server **104** may further include other program data **228**. The other program data **228** may store device identifiers of the plurality of client devices **108** that are authorized to use and access the cooperative web browsing system **102**, passwords of one or more authorized users, log data of web browsing sessions, etc.

FIG. **3** illustrates an example architecture of the example touch controller **124** for web browsing based on a thin-client browser framework. In this example, the touch controller **124** is illustrated to be used as a remote controller for a television **302**. However, the touch controller **124** is not limited thereto and can be used as a remote controller for other types of devices. In this example, the television **302** may include a limited TV browser client **304**. For example, the television **302** may be associated with the limited TV browser client **304** provided by other devices such as set-top box, game console or other companion device which usually has general computing capabilities. The TV browser client **304** may receive compressed (and/or encoded) screen, audio and web elements from the proxy server **104**, and decode **306** the compressed screen, audio and web elements for presentation **308** in the television **302** to the user **122**.

Additionally, the TV browser client **304** may further send **310** screen data and the web elements to the touch controller **124** through the local network **106** to enable user-friendly interaction, for example. In some embodiments, the TV browser client **304** may truncate **312** the screen data (or bits) and send. In one embodiment, the touch controller **124** may decode **314** and present **316** the web screen to the user **122**. The touch controller **124** may further receive and process **318** various user input operations of the user **122** including, for example, clicking, scrolling, text inputting, zooming, panning, etc., on a display of the touch controller **124**. In one embodiment, the touch controller **124** may further employ the web-element side information to enable quick response to an input of the user **122**.

By way of example and not limitation, when the user **122** clicks or selects a web element, the touch controller **124** may responds correspondingly based on a type of the web element. For example, if the user **122** clicks or selects a web element such as a hyperlink in the touch controller **124**, the touch controller **124** may highlight the clicked web element and indicate to the user **122** that associated user input has been accepted, thereby avoiding duplicated clicks or selections. For another example, if the user **122** clicks a text input box displayed in the touch controller **124**, the touch controller **124** may display an input box and a soft keyboard to the user **122** for user input with or without causing a corresponding display of the input box and the soft keyboard in the television, for example.

Additionally or alternatively, in some embodiments, the touch controller **124** may further allow the user **122** to perform various input gestures on the touch controller **124** to alter the portion of the web page displayed on the display of the touch controller **124** without altering the content of the web page displayed in the other client device (i.e., the television in this example). For example, the touch controller **124** may allow the user **122** to move to have another portion of the web page to be displayed in the touch controller **124**, zoom in or out of the portion of the web page currently displayed, etc., without causing a change to the content of the web page displayed in the other client device.

In response to receiving the user input operations, the touch controller **124** may send **320** information of these user input operations as control messages to the TV browser client **304**. In response to receiving the information of these user input operations from the touch controller **124**, the TV browser client **304** may update **322** content of the web page to be displayed in the television according to the received control messages. Additionally or alternatively, in some embodiments, the touch controller **124** may send information of these user input operations as control messages to the proxy server **104** which may send updated samples to the TV browser client **304** for presentation to the user **122**.

In one embodiment, since spatial and temporal resolutions or criteria may be different for the television and the touch controller **124** (which usually has a smaller display size), a spatially downsized web page image or screen may be sent to the touch controller **124** in order to save power and avoid unnecessary bandwidth consumption. Furthermore, the web image or screen on the touch controller **124** may be updated at a lower frame rate than that of the TV browser client **304**. In order to achieve this effect, in one embodiment, the TV browser client **304** may transcode web page images or screens received from the proxy server **104** to images or screens with downsized spatial size and send the downsized images or screens at a lower frame rate that is suitable or acceptable for presentation in the touch controller **124**. Additionally or alternatively, in some embodiments, the proxy server **104** or the sample encoder **222** may employ a scalable screen encoding algorithm to support multiple decodable spatial regions and temporal frame rates in a single stream. With this scalable stream, the TV browser client **304** may only need to truncate screen bits of the web page images and screens received from the proxy server **104**, select and send to the touch controller **124** base-layer bits that provide the downsized spatial region and frame rate without transcoding.

FIG. **4** illustrates an example single layer screen encoding algorithm that may be used by the sample encoder **222**. In one embodiment, the sample encoder **222** may first segment **402** each web page image into a number of un-overlapped blocks (e.g., un-overlapped 16×16 blocks). The sample encoder **222** may then perform difference detection **404** for each of these blocks between two successive web page images or frames. In one embodiment, the sample encoder **222** may identify one or more blocks that are not changed between two successive web page images or frames as skipped blocks. For example, the sample encoder **222** may identify that a block is changed between successive web page images or frames as a skipped block if a difference between respective blocks in the successive web page images or frames is less than a predetermined threshold. Otherwise, the sample encoder **222** may identify the block as a non-skipped block. In one embodiment, the sample encoder **222** may not perform further processing (except compression operation **406**, for example) for the skipped blocks.

Upon performing difference detection **404**, the sample encoder **222** may classify **408** the non-skipped blocks into image blocks and text blocks based on, for example, respective contents. In one embodiment, the sample encoder **222** may classify a non-skipped block having sharp edges and/or a limited number of colors as a text block, and classify the block as an image block otherwise. Additionally or alternatively, in some embodiments, the sample encoder **222** may employ the web-element side information (such as the type and the position of the web-elements) to determine whether a non-skipped block is a text block or an image block. For example, the sample encoder **222** may find that a non-skipped block is within an area corresponding to text information of the web page based on the web-element side information and determine that the non-skipped block is a text block.

Upon classifying the non-skipped blocks into image blocks and text blocks, the sample encoder **222** may encode the image blocks and the text blocks using different coding schemes. By way of example and not limitation, the sample encoder **222** may encode **410** the image blocks using, for example, a DCT (Discrete Cosine Transform) coding scheme, and encode **412** the text blocks using, for example, a pixel domain coding scheme. Upon encoding the image blocks and the text blocks, the sample encoder **222** may compress the image blocks and the text blocks using a compression algorithm such as entropy coding **414**, for example.

Additionally or alternatively, in one embodiment, the sample encoder **222** may employ a scalable screen encoding algorithm. FIG. **5** illustrates an example scalable screen compression framework for the scalable screen encoding algorithm. As shown in FIG. **5**, the scalable screen compression framework includes two layers—one is a base layer **502** and the other is an enhancement layer **504**. A solid line arrow **506** in FIG. **5** represents a temporal reference relationship between corresponding two screen frames. In one embodiment, a solid rectangle **508** in the base layer **502** corresponds to an encoded region visible to the touch controller **124**, and content outside this visible region of the touch controller **124** is not encoded in the base layer **502**. Temporally a web page image or screen may be updated with a relatively low frequency in the base layer **502** (e.g., one frame per second) as compared to that of the enhancement layer **504**. Furthermore, a current frame that is displayed in the touch controller **124** may take a frame with a temporally longer distance as reference. In one embodiment, the visible region of the touch controller **124** may vary when the user **122** scrolls or pans the content displayed in the touch controller **124**, for example. Additionally, the region of updated content in the base layer **502** may also change accordingly.

In some embodiments, the enhancement layer **504** may include content in a larger region than the base layer **502**, which may correspond to a visible region of a client device having a larger display such as the television **302** (and/or the TV browser client **304**). Since some sub-region contents have been updated in the base layer **502**, the sample encoder **222** may skip these sub-region content in the enhancement layer **504**, as denoted by the dashed rectangle **510** in the enhancement layer **504**, which corresponds to the solid rectangle **508** in the base layer **502**. In the temporal dimension, the enhancement layer **504** may update the content in a higher frame rate, e.g., 20 fps.

With this scalable encoding framework, the proxy server **104** or the TV browser client **304** may only send frames in the base layer **502** to the touch controller **124**, while an entire stream (including frames in the base layer **502** and the enhancement layer **504**) is sent to the TV browser client **304** for decoding and presentation in the television for the user **122**. Furthermore, in one embodiment, the sample encoder **222** may encode the sub-region content of a full-size screen of the other client device **108** (i.e., the television in this example) in the base layer **502** rather than a down-sampled version of the full-size screen of the other client device **108**. It is because clickable or selectable web elements may be too small and/or clustered together in the display of the touch controller **124** for the down-sampled version. In an alternative embodiment, however, the sample encoder **222** may encode a down-sampled version of the full-size screen of the other client device **108** in the base layer **502** if, for example, receiving a request from the user **122**.

FIGS. **6**A and **6**B illustrate an example scenario when content displayed in the touch controller **124** changes due to a user input. FIG. **6**A illustrates content and web elements rendered and extracted by the virtual browser **218** of the proxy server **104**. FIG. **6**B illustrates an update of content displayed in the touch controller **124** in response to receiving a panning or moving operation (as represented by an arrow **602**) from the user **122**.

In one embodiment, the proxy server **104** may receive a request from the user **122** for migrating a browsing session from one client device to another client device. A migration of a browsing session herein refers to a scenario that web browsing content of the browsing session is migrated from one client device to another client device. For example, the user **122** may be using a first client device (e.g., the client device **108-1**) for playing an online video and want to move the browsing session associated with the online video from the first client device **108-1** to a second client device (such as the client device **108-2**) while content of the online video is displayed continuously despite the client devices **108-1** and **108-2** are different.

In one embodiment, during the migration, the proxy server **104** may fully preserve the states of the browsing session of the first client device **108-1**, including information inputted by the user **122** using the first client device **108-1** (such as information inputted by the user **122** into a textbox, a form, etc., in the web page), browsing history, cookies, cache, a playback position of media content (such as video, audio, etc.), a running state of an application program (e.g., JavaScript™, Flash™, etc.) associated with the browsing session, etc. The proxy server **104** may store these states of the browsing session in an instance of the virtual browser **218** corresponding to the browsing session of the first client device **108-1**. In one embodiment, the proxy server **104** may employ information of these states to restore the browsing session in the second client device **108-2**. Furthermore, the proxy server **104** may employ information of these states to help reducing a migration delay of the browsing session from the first client device **108-1** to the second client device **108-2** such that the user **122** may feel the two client devices **108-1** and **108-2** to be bridged seamlessly.

In one embodiment, the proxy server **104** may achieve a session migration by changing a link relationship between an instance of the virtual browser **218** and the client device **108** (from which corresponding browsing session is migrated). For example, the proxy server **104** may receive from the first client device **108-1** a migration request including a migration destination (i.e., the second client device **108-2**). In response to receiving the migration request, the proxy server **104** may link a browsing session originally associated with the first client device **108-1** to the second client device **108-2**. For example, the proxy server **104** may redirect a connection of an instance of the virtual browser **218** associated with the browsing session from the first client device **108-1** to the second client device **108-2**. Thereafter, the instance of the virtual browser **218** (which was originally lined to the first client device **108-1**) may send the rendered and encoded content (e.g., rendered screen, audio and web elements) associated with the browsing session to the second client device **108-2**. Furthermore, user inputs or control messages from the second client device **108-2** may be redirected to the browsing session associated with this same instance of the virtual browser **218**. Since the proxy server **104** stores all browsing state information in the instance of the virtual browser **218**, the proxy server **104** (and the virtual browser **218**) may complete this session migration statefully and provide a temporally seamless experience to the user **122**.

Furthermore, the proxy server **104** may enable different modes of session migrations between client devices **108**, including, for example, a move mode, a copy mode and a switch mode. The proxy server **104** may perform differently depending on which mode of session migration the user **122** is requesting. By way of example and not limitation, if the user **122** requests a move mode of session migration from a first client device to a second client device, the proxy server **104** may redirect an instance of the virtual browser **218** of the first client device to the second client device. Additionally, in some embodiments, the proxy server **104** may further destroy an original instance of the virtual browser **218** of the second client device, if one exists. In an alternative embodiment, the proxy server **104** may still maintain an original instance of the virtual browser **218** of the second client device (if one exists).

Additionally, in some embodiments, if the user **122** requests to copy a browsing session of a first client device to a second client device, the proxy server **104** may redirect an instance of the virtual browser **218** of the first client device to the second client device. Additionally, the proxy server **104** may create or construct a new instance of the virtual browser **218** for the first client device. In one embodiment, the proxy server **104** may retrieve original browsing history and browsing state information of the first client device to initiate the new instance of the virtual browser **218**. For example, the user **122** may find an interesting video when browsing on the mobile phone and may want to migrate the video to the television. The proxy server **104** may therefore migrate an associated browsing session from the mobile phone to the television, and create a new browsing session with original browsing session information the mobile phone so that the user **122** may resume to use the mobile phone to read and post comments about the video on the same page as the television.

Additionally or alternatively, in some embodiments, if the user **122** requests to switch a first browsing session of a first client device with a second browsing session of a second client device, the proxy server **104** may redirect a first instance of the virtual browser **218** associated with the first browsing session of the first client device to the second client device, and redirect a second instance of the virtual browser **218** associated with the second browsing session of the second client device to the first client device. For example, the user **122** may watch a video on a television while reading twitter on a mobile phone. After a short while the user **122** may want to read comments in a web page associated with the video and publish his/her own idea without missing updates in twitter. In this case the user **122** may switch between two browsing sessions of the television and the mobile phone. From another perspective, this may amount to taking a display of the television as an extended display of the mobile phone.

Although the foregoing embodiments describe session migration upon receiving a request from the user **122**, in some embodiments, the proxy server **104** may perform session migration automatically. By the way of example and not limitation, the proxy server may migrate or transfer a browsing session of a first client device (e.g., the client device **108-1**) to a second client device (e.g., the client device **108-2**) automatically in response to detecting a presence of the second client device in a neighborhood of the first client device and determining that the second client device has a better capability of serving a type of content that is currently provided to the first client device by the proxy server **104** in comparison with the first client device. For example, the user **122** may watch an online video on the way home using a mobile phone. When the user **122** comes to a location (e.g., home) where the cooperative web browsing system **102** or the proxy server **104** is located, the user **122** may want to continue to watch the online video on another client device such as a television. Upon turning on the television, the proxy server **104** may detect a presence of the television in the neighborhood of the mobile phone in which the online video is played and further determine that a display capability (such as display resolution or quality) is better than that of the mobile phone. The proxy server **104** may infer that the user **122** wants to transfer a browsing session associated with the online video on the mobile phone to the television, and automatically perform the session migration on behalf of the user **122**.

Exemplary Methods

FIG. **7** is a flow chart depicting an example method **700** of cooperative web browsing of multiple devices. FIG. **8** is a flow chart depicting an example method **800** of using the example touch controller **124** in cooperative web browsing. FIG. **9** is a flow chart depicting an example method **900** of session migration. The methods of FIG. **7-9** may, but need not, be implemented in the environment of FIG. **1** and using the systems of FIGS. **2** and **3**. For ease of explanation, methods **700**, **800** and **900** are described with reference to FIGS. **1-3**. However, the methods **700**, **800** and **900** may alternatively be implemented in other environments and/or using other systems.

Methods **700**, **800** and **900** are described in the general context of computer-executable instructions. Generally, computer-executable instructions can include routines, programs, objects, components, data structures, procedures, modules, functions, and the like that perform particular functions or implement particular abstract data types. The methods can also be practiced in a distributed computing environment where functions are performed by remote processing devices that are linked through a communication network. In a distributed computing environment, computer-executable instructions may be located in local and/or remote computer storage media, including memory storage devices.

The exemplary methods are illustrated as a collection of blocks in a logical flow graph representing a sequence of operations that can be implemented in hardware, software, firmware, or a combination thereof. The order in which the methods are described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method, or alternate methods. Additionally, individual blocks may be omitted from the method without departing from the spirit and scope of the subject matter described herein. In the context of software, the blocks represent computer instructions that, when executed by one or more processors, perform the recited operations. In the context of hardware, some or all of the blocks may represent application specific integrated circuits (ASICs) or other physical components that perform the recited operations.

Referring back to FIG. **7**, at block **702**, the proxy server **104** may receive a browsing request (or a connection request) from a user (e.g., the user **122**) to access or browse a website through the cooperative web browsing system **102**.

At block **704**, upon receiving the request, the proxy server **104** may obtain content of a web page associated with the website, parse and render the content into a display bitmap. The proxy server **104** may further extract other content including, for example, audio and web elements from the web page.

At block **706**, the proxy server **104** may compress and/or encode the rendered content using a scalable screen encoding algorithm.

At block **708**, in response to compressing and/or encoding the rendered content, the proxy server **104** may send the encoded content to the client device **108** for presenting to the user **122**.

At block **710**, the proxy server **104** may wait for another request or control message from the client device **108** (from which the browsing request is sent) or another client device **108** of the user **122**.

Referring back to FIG. **8**, at block **802**, the touch controller **124** may display a portion less than all of a web page currently displayed at another client device **108** that is incapable of manipulating the web page, e.g., having no physical or soft keyboard, having no or limited web browser, etc.

At block **804**, the touch controller **124** may receive a manipulation gesture from a user to manipulate the portion of the web page that is displayed at the first device. By way of example and not limitation, the manipulation gesture may include, but is not limited to, selecting or clicking a hyperlink, actuating a control on the portion of the web page, etc.

At block **806**, in response to receiving the manipulation gesture from the user, the touch controller **124** may send information associated with the manipulation gesture to the proxy server **104** or the other client device **108** to cause updating of the web page that is currently displayed at the other client device **108**. In one embodiment, the proxy server **104** may serve different portions of the web page or different web pages to the touch controller **124** and the other client device **108** in response to this manipulation gesture.

Referring back to FIG. **9**, at block **902**, the proxy server **104** may receive a request to transfer a browsing session of a first client device to a second client device.

At block **904**, in response to receiving the request, the proxy server **104** may redirect a connection of an instance of a virtual browser associated with the browsing session of the first client device at the proxy server from the first client device to the second client device without determining which mode of session migration request the user **122** is requesting. This speeds up the session migration to the second client device and facilitates seamless migration of the browsing session from one client device to another client device.

At block **906**, the proxy server **104** may selectively serve new content of the browsing session to the second device based on one or more capabilities of the second device.

At block **908**, the proxy server **104** may determine which mode of migration request the user **122** is requesting. Example modes of session migration may include, for example, a move mode, a copy mode and a switch mode.

At block **910**, in response to determining that the user **122** requests to move the browsing session of the first client device to the second client device, the proxy server **104** may create a new virtual browser instance of a new browsing session for the first client device.

At block **912**, in response to determining that the user **122** requests to copy the browsing session of the first client device to the second client device, the proxy server **104** may create a new virtual browser instance of the original browsing session for the first client device by retrieving original browsing history and browsing state information of the first client device to initiate the new virtual browser instance.

At block **914**, in response to determining that the user **122** requests to switch the browsing session of the first client device with a browsing session of the second client device, the proxy server **104** may redirect a connection of an instance of a virtual browser associated with the browsing session of the second client device at the proxy server from the second client device to the first client device.

Any of the acts of any of the methods described herein may be implemented at least partially by a processor or other electronic device based on instructions stored on one or more computer-readable media. By way of example and not limitation, any of the acts of any of the methods described herein may be implemented under control of one or more processors configured with executable instructions that may be stored on one or more computer-readable media such as one or more computer storage media.

CONCLUSION

Although the invention has been described in language specific to structural features and/or methodological acts, it is to be understood that the invention is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the invention.

What is claimed is:

1. A proxy device comprising:

a processor; and

memory communicatively coupled to the processor and storing computer-executable instructions that, when executed, perform operations comprising:

receiving, from a first device, a browsing request to access web content, the browsing request being associated with access information for the first device;

providing the first device access to a virtual browser for accessing the web content, the virtual browser facilitating a browsing session associated with the first device;

receiving, from the first device, a transfer request to migrate the browsing session to a second device;

in response to the transfer request, redirecting a first instance of the browsing session to the second device, wherein the second device is able to access the web content via the virtual browser;

in response to redirecting the first instance of the browsing session to the second device, determining the transfer request corresponds to a migration request mode of a set of migration request modes, wherein the migration request mode is a move request; and

in response to the move request, creating a second instance of the browsing session for the first device, wherein the second instance is a new instance of the browsing session.

2. The proxy device of claim **1**, wherein the proxy device in implemented by a cooperative web browsing system that enables multiple user devices of a user to concurrently access web browsing content.

3. The proxy device of claim **2**, wherein the virtual browser maintains state information of active browsing sessions associated with the multiple user devices of the user.

4. The proxy device of claim **2**, wherein the virtual browser parses and renders the web browsing content on behalf of the multiple user devices of the user.

5. The proxy device of claim **1**, wherein the access information comprises at least one of:

a user password;

a device identifier for the first device; or

an Internet Protocol (IP) address for the first device.

6. The proxy device of claim **1**, wherein:

upon receiving the move request, the first device is detected to be within a threshold distance of the second device; and

redirecting the instance of the browsing session to the second device based on the first device being within the threshold distance of the second device.

7. The proxy device of claim **1**, wherein, upon receiving the move request, the first device is detected to be outside of a threshold distance of the second device.

8. The proxy device of claim **1**, wherein redirecting the instance of the browsing session to the second device comprises at least one of:

ceasing display of the web content on the first device; or

terminating the browsing session on the first device.

9. The proxy device of claim **1**, wherein redirecting the instance of the browsing session to the second device does not terminate the browsing session on the first device.

10. The proxy device of claim **1**, wherein redirecting the instance of the browsing session to the second device comprises terminating an active browsing session on the second device to present the instance of the browsing session from the first device.

11. The proxy device of claim **1**, wherein redirecting the instance of the browsing session to the second device comprises:

recording a state of an active browsing session on the second device;

deactivating the active browsing session;

presenting the instance of the browsing session from the first device on the second device; and

reactivating the active browsing session on the second device when the instance of the browsing session from the first device is terminated on the second device.

12. The proxy device of claim **1**, wherein receiving the move request comprises:

determining a capability of the second device exceeds a capability of the first device; and

generating the move request based on the determining.

13. A system comprising:

a processor; and

memory communicatively coupled to the processor and storing computer-executable instructions that, when executed, perform operations comprising:

receiving, by a proxy device, a browsing request from a first device, the browsing request being a request to access first web content and being associated with access information for the first device;

providing the first device access to a virtual browser for accessing the first web content, the virtual browser facilitating a first browsing session associated with the first device;

receiving, by the proxy device, a transfer request to migrate the first browsing session to a second device that is associated with a second browsing session;

in response to the transfer request, redirecting a first instance of the first browsing session to the second device, wherein the second device is able to access the first web content via the virtual browser;

in response to redirecting the first instance of the first browsing session to the second device, determining the transfer request corresponds to a migration request mode of a set of migration request modes, wherein the migration request mode is a switch request; and

in response to the switch request, redirecting a second instance of the second browsing session to the first device.

14. The system of claim **13**, wherein redirecting the second instance of the second browsing session to the first device enables the first device to access second web content via the virtual browser.

15. The system of claim **13**, wherein the proxy device provides a centralized thin-client browser framework that enables interactions among multiple devices of a user.

16. The system of claim **15**, wherein the proxy device facilitates cooperative web browsing of the multiple devices of the user regardless of whether one or more devices of the multiple devices has limited or no capabilities of web browsing.

17. The system of claim **13**, wherein the proxy device:

encodes the first web content; and

provides encoded first web content to at least the first device.

18. The system of claim **13**, wherein redirecting the first instance of the first browsing session to the second device comprises:

retrieving a browsing history and browsing state information of the first device; and

using the browsing history and browsing state information of the first device to initiate the first instance of the first browsing session on the second device.

19. The system of claim **18**, wherein redirecting the second instance of the second browsing session to the first device comprises:

retrieving a browsing history and browsing state information of the second device; and

using the browsing history and browsing state information of the second device to initiate the second instance of the second browsing session on the first device.

20. A method comprising:

receiving, by a proxy device, a browsing request from a first device, the browsing request being a request to access web content and being associated with access information for the first device;

providing the first device access to a virtual browser for accessing the web content, the virtual browser facilitating a first browsing session associated with the first device;

receiving, by the proxy device, a transfer request to migrate the first browsing session to a second device that is associated with a second browsing session;

in response to the transfer request, redirecting the first browsing session to the second device;

in response to redirecting the first browsing session to the second device, determining the transfer request corresponds to a migration request mode of a set of migration request modes,

wherein the migration request mode is a switch request; and

in response to the switch request, redirecting the second browsing session to the first device.

* * * * *