



US010873615B1

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

(10) **Patent No.:** **US 10,873,615 B1**

(45) **Date of Patent:** ***Dec. 22, 2020**

(54) **SOURCE ASSIGNMENT BASED ON NETWORK PARTITIONING**

(71) Applicant: **Conviva Inc.**, Foster City, CA (US)

(72) Inventors: **Ion Stoica**, Piedmont, CA (US); **Hui Zhang**, Pittsburgh, PA (US); **Xi Liu**, Belmont, CA (US); **Florin Dobrian**, Norfolk, VA (US); **Henry Milner**, Berkeley, CA (US)

(73) Assignee: **Conviva Inc.**, Foster City, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 134 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/971,023**

(22) Filed: **Dec. 16, 2015**

Related U.S. Application Data

(63) Continuation of application No. 13/752,252, filed on Jan. 28, 2013, now Pat. No. 9,246,965.

(60) Provisional application No. 61/697,225, filed on Sep. 5, 2012.

(51) **Int. Cl.**
H04L 29/06 (2006.01)
H04L 29/08 (2006.01)

(52) **U.S. Cl.**
CPC **H04L 65/602** (2013.01); **H04L 65/4084** (2013.01); **H04L 65/80** (2013.01); **H04L 67/16** (2013.01); **H04L 67/42** (2013.01)

(58) **Field of Classification Search**
CPC **H04L 65/60**; **H04L 65/80**; **H04L 65/4092**; **H04L 65/602**; **H04L 67/16**; **H04L 67/42**; **H04L 65/4084**

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,926,204 A	7/1999	Mayer
5,930,473 A	7/1999	Teng
5,987,621 A	11/1999	Duso
6,006,264 A	12/1999	Colby
6,026,077 A	2/2000	Iwata
6,032,857 A	3/2000	Kitagawa
6,188,993 B1	2/2001	Eng
6,208,977 B1	3/2001	Hernandez
6,223,206 B1	4/2001	Dan
6,250,557 B1	6/2001	Forstlund

(Continued)

OTHER PUBLICATIONS

"Relay Nodes in Wireless Sensor Networks: A Survey" Ataul Bad, University of Windsor, Nov. 2005. http://richard.myweb.cs.uwindsor.ca/cs510/survey_bari.pdf.

(Continued)

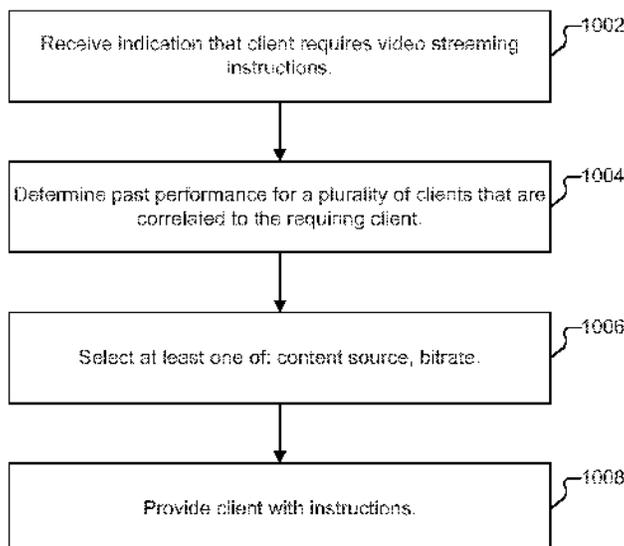
Primary Examiner — Aaron N Strange

(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Directing a content player to a content source is disclosed. A first manifest request from a client requesting content is received at a server. The manifest request includes information generated as a result of the client contacting a content management system and in response, receiving from the content management system information associated with the client and a direction to the server. Based at least in part on the received information, at least one of a bitrate and a content source is determined. Instructions are sent to the client based at least in part on the determination. The client is configured to obtain the requested content according to the instructions.

17 Claims, 11 Drawing Sheets



US 10,873,615 B1

Page 2

(56)

References Cited

U.S. PATENT DOCUMENTS

6,279,039	B1	8/2001	Bhal	8,038,535	B2	10/2011	Jensen
6,324,565	B1	11/2001	Holl, III	8,046,765	B2	10/2011	Cherkašova
6,377,996	B1	4/2002	Lumelsky	8,098,160	B2	1/2012	Howarth
6,405,251	B1	6/2002	Bullard	8,108,403	B2	1/2012	Gopalraj
6,434,159	B1	8/2002	Woodward	8,135,855	B2	3/2012	Sitaraman
6,438,592	B1	8/2002	Killian	8,156,204	B2	4/2012	Leblanc
6,469,991	B1	10/2002	Chuah	8,182,326	B2	5/2012	Specr, II
6,470,389	B1	10/2002	Chung	8,230,105	B2	7/2012	Melnyk
6,487,540	B1	11/2002	Smith	8,234,350	B1 *	7/2012	Gu H04N 21:234309 709/203
6,502,747	B1	1/2003	Stoutenburg	8,259,597	B1	9/2012	Oak
6,542,933	B1 *	4/2003	Durst, Jr. G06F 16/9554 709/229	8,370,887	B2	2/2013	Virdi
6,711,622	B1	3/2004	Fuller	8,374,929	B1	2/2013	Lappas
6,735,630	B1	5/2004	Gelvin	8,387,094	B1	2/2013	Ilo
6,754,833	B1	6/2004	Black	8,402,494	B1	3/2013	Ilu
6,769,028	B1 *	7/2004	Sass H04L 29:12047 348/L7.071	8,417,797	B2	4/2013	Thoen
6,836,691	B1	12/2004	Stirton	8,484,319	B2	7/2013	Wein
6,892,236	B1	5/2005	Conrad	8,489,683	B2	7/2013	Leblanc
6,892,307	B1	5/2005	Wood	8,489,923	B1	7/2013	Lakshminarayanan
6,906,743	B1	6/2005	Maurer	8,555,355	B2	10/2013	Ralhbun
6,920,580	B1	7/2005	Cramer	8,589,473	B2	11/2013	Bruss
6,941,270	B1	9/2005	Hannula	8,639,553	B1	1/2014	Knauth
6,950,855	B2	9/2005	Sampathkumar	8,647,203	B2	2/2014	Albrecht
7,006,666	B2	2/2006	Montgomery	8,677,428	B2	3/2014	Lewis
7,010,598	B2	3/2006	Sitaraman	8,683,066	B2	3/2014	Hurst
7,020,082	B2	3/2006	Bhagavath	8,719,876	B2	5/2014	Kotecha
7,024,452	B1 *	4/2006	O'Connell, Jr. .. G06F 17:30902 707/F17.12	8,751,605	B1	6/2014	Zhang
7,024,468	B1	4/2006	Meyer	8,751,679	B2	6/2014	McIlugh
7,039,694	B2	5/2006	Kampe	8,843,597	B2	9/2014	Leblanc
7,092,696	B1	8/2006	Hosain	8,874,725	B1	10/2014	Ganjam
7,139,834	B1	11/2006	Albanesc	8,874,964	B1	10/2014	Lakshminarayanan
7,159,234	B1	1/2007	Murphy	8,898,338	B1	11/2014	McGowan
7,222,190	B2	5/2007	Klinker	8,909,736	B1	12/2014	Bosch
7,233,926	B2	6/2007	Durand	8,924,996	B2	12/2014	Shafice
7,277,896	B2	10/2007	Matsubara	8,930,991	B2	1/2015	Philpott
7,313,087	B2	12/2007	Patil	8,943,170	B2	1/2015	Li
7,318,107	B1	1/2008	Menon	8,954,491	B1	2/2015	Medved
7,353,023	B1	4/2008	Link, II	9,098,844	B2	8/2015	Davis
7,356,341	B2	4/2008	Nanda	9,100,288	B1	8/2015	Ganjam
7,367,044	B2	4/2008	Fowler	9,100,826	B2	8/2015	Weiss
7,373,415	B1	5/2008	Deshan	9,264,780	B1	2/2016	Stoica
7,389,537	B1	6/2008	Callon	9,356,821	B1	5/2016	Jagannathan
7,487,509	B2	2/2009	Hugly	9,456,015	B2	9/2016	Chen
7,490,136	B2	2/2009	Suzuki	9,549,043	B1	1/2017	Stoica
7,509,372	B2	3/2009	Dutta	9,613,042	B1 *	4/2017	Joseph G06F 16:16
7,519,703	B1	4/2009	Stuart	9,819,566	B1 *	11/2017	Ganjam H04L 43:08
7,574,488	B2	8/2009	Matsubara	9,898,781	B1	2/2018	Silverman
7,593,333	B2	9/2009	Ti	10,021,672	B2	7/2018	Cole
7,599,698	B2	10/2009	Cheng	10,530,852	B2	1/2020	Newton
7,617,525	B1	11/2009	Mocck	2002/0002708	A1	1/2002	Aryc
7,620,848	B1	11/2009	Tanner	2002/0016831	A1	2/2002	Peled
7,627,872	B2	12/2009	Hebeler	2002/0029190	A1	3/2002	Gutierrez-Sheris
7,668,761	B2	2/2010	Jenkins	2002/0052841	A1	5/2002	Guthrie
7,668,914	B2	2/2010	Parker	2002/0082730	A1	6/2002	Capps
7,689,485	B2	3/2010	Kanekar	2002/0095400	A1	7/2002	Johnson
7,689,508	B2	3/2010	Davis	2002/0126135	A1	9/2002	Ball
7,698,460	B2	4/2010	Zhang	2002/0133601	A1	9/2002	Kennamer
7,702,581	B2	4/2010	Hogl	2002/0141343	A1	10/2002	Bays
7,721,292	B2	5/2010	Frasier	2002/0143798	A1	10/2002	Lisiecki
7,725,576	B2	5/2010	Sitaraman	2002/0175934	A1	11/2002	Hand
7,792,889	B1	9/2010	Lee	2002/0183972	A1	12/2002	Enck
7,813,326	B1	10/2010	Kelm	2002/0184357	A1	12/2002	Traversat
7,844,491	B1	11/2010	Haitisuka	2002/0198984	A1	12/2002	Goldstein
7,853,529	B1	12/2010	Walker	2003/0046383	A1	3/2003	Lee
7,873,727	B2	1/2011	Pal	2003/0046396	A1	3/2003	Richter
7,904,580	B2	3/2011	Mandera	2003/0046704	A1	3/2003	Laksono
7,921,215	B2	4/2011	Dessart	2003/0046708	A1	3/2003	Jutzi
7,921,222	B2	4/2011	Courtemanche	2003/0050966	A1	3/2003	Dutta
7,930,347	B2	4/2011	Maxwell	2003/0051051	A1	3/2003	O'Neal
7,941,823	B2	5/2011	Hasck	2003/0061305	A1	3/2003	Copley
7,969,987	B1	6/2011	Hansen	2003/0061356	A1	3/2003	Jason, Jr.
7,970,402	B2	6/2011	Wu	2003/0065739	A1	4/2003	Shmier
8,028,159	B2	9/2011	Ti	2003/0065763	A1	4/2003	Swildens
				2003/0074142	A1	4/2003	Stecg
				2003/0084003	A1	5/2003	Pinkas
				2003/0105850	A1	6/2003	Lean
				2003/0135593	A1	7/2003	Lee
				2003/0140108	A1	7/2003	Sampathkumar
				2003/0140180	A1	7/2003	Brown

(56)

References Cited

U.S. PATENT DOCUMENTS

2003-0145066	A1	7/2003	Okada	2007/0282994	A1	12/2007	Beers
2003-0169863	A1	9/2003	Hernandez	2007/0286351	A1	12/2007	Ethier
2003-0204613	A1	10/2003	Hudson	2007/0288638	A1	12/2007	Vuong
2004/0010544	A1	1/2004	Slater	2008/0016205	A1	1/2008	Svendesen
2004/0019675	A1	1/2004	Hebeler	2008/0037438	A1	2/2008	Iwiss
2004-0047354	A1	3/2004	Slater	2008/0046499	A1	2/2008	Cabrera
2004-0049793	A1	3/2004	Chou	2008/0046777	A1	2/2008	Chen
2004-0057420	A1	3/2004	Curcio	2008/0046974	A1	2/2008	Minodier
2004/0064556	A1	4/2004	Zhang	2008/0049616	A1	2/2008	Kamath
2004/0088347	A1	5/2004	Ycager	2008/0049786	A1	2/2008	Ram
2004/0093155	A1	5/2004	Simonds	2008/0052226	A1	2/2008	Agarwal
2004/0107387	A1	6/2004	Larsson	2008/0063195	A1	3/2008	Li
2004-0128682	A1	7/2004	Tiga	2008/0096562	A1	4/2008	Wu
2004-0133471	A1	7/2004	Pisaris-Henderson	2008/0104454	A1	5/2008	White
2004-0136327	A1	7/2004	Sitaraman	2008/0112315	A1	5/2008	Hu
2004-0158643	A1	8/2004	Suzuki	2008/0120707	A1	5/2008	Ramia
2004/0162901	A1	8/2004	Mangipudi	2008/0140520	A1	6/2008	Hyder
2004/0187159	A1	9/2004	Gaydos	2008/0140688	A1	6/2008	Clayton
2004-0193716	A1	9/2004	McConnell	2008/0141131	A1	6/2008	Corny
2004-0233918	A1	11/2004	Larsson	2008/0144513	A1	6/2008	Small
2004-0236846	A1	11/2004	Alvarez	2008/0151821	A1	6/2008	Cho
2004-0267691	A1	12/2004	Vasudeva	2008/0155586	A1	6/2008	Yang
2005/0010915	A1	1/2005	Chen	2008/0195461	A1	8/2008	Ti
2005/0021715	A1	1/2005	Dugatkin	2008/0209499	A1	8/2008	Ramesh
2005/0060158	A1	3/2005	Lindo	2008/0215718	A1	9/2008	Stolorz
2005-0076104	A1	4/2005	Tiskov	2008/0215756	A1	9/2008	Lee
2005-0086300	A1	4/2005	Yeager	2008/0235746	A1	9/2008	Peters
2005-0086469	A1	4/2005	Dunagan	2008/0247326	A1	10/2008	Cormier
2005/0089043	A1	4/2005	Seckin	2008/0263180	A1	10/2008	Hurst
2005/0114262	A1	5/2005	Howard	2008/0305801	A1	12/2008	Burgess
2005/0120131	A1	6/2005	Allen	2008/0313040	A1	12/2008	Rose
2005/0169179	A1	8/2005	Antal	2009/0010155	A1	1/2009	Liu
2005-0177785	A1	8/2005	Shrader	2009/0019503	A1	1/2009	Vorbau
2005-0183120	A1	8/2005	Jain	2009/0043906	A1	2/2009	Hurst
2005-0187873	A1	8/2005	Labrou	2009/0059812	A1	3/2009	Chinnaswamy
2005/0243735	A1	11/2005	Kashima	2009/0082020	A1	3/2009	Ch'ng
2005/0251835	A1	11/2005	Scott	2009/0083631	A1	3/2009	Sidi
2005/0278259	A1*	12/2005	Gunaseclan	2009/0106392	A1	4/2009	Zuckerman
				2009/0117893	A1	5/2009	Weigand
				2009/0119722	A1	5/2009	Versteeg
				2009/0125625	A1	5/2009	Shim
				2009/0150487	A1	6/2009	Wollich
				2009/0164656	A1	6/2009	Guan
				2009/0170440	A1	7/2009	Fyuboglu
				2009/0172200	A1	7/2009	Morrison
				2009/0187956	A1	7/2009	Sommer
				2009/0192929	A1	7/2009	Hoeflinger
				2009/0234940	A1	9/2009	Pal
				2009/0248872	A1	10/2009	Fuzzalti
				2009/0271101	A1	10/2009	Relyea
				2009/0271520	A1	10/2009	Siddiqui
				2009/0288012	A1	11/2009	Hertel
				2009/0327489	A1	12/2009	Swildens
				2009/0328124	A1	12/2009	Khouzam
				2010/0043014	A1	2/2010	Hebeler, Jr.
				2010/0080290	A1	4/2010	Mehrotra
				2010/0088373	A1	4/2010	Pinkham
				2010/0100635	A1	4/2010	Takemura
				2010/0114562	A1	5/2010	Hutchinson
				2010/0125675	A1	5/2010	Richardson
				2010/0131642	A1	5/2010	Chalikouras
				2010/0138273	A1	6/2010	Baleni
				2010/0161729	A1	6/2010	Leblanc
				2010/0169195	A1	7/2010	Trest
				2010/0235503	A1	9/2010	Sitaraman
				2010/0241701	A1	9/2010	Lester
				2010/0302002	A1	12/2010	Guo
				2010/0306368	A1*	12/2010	Gagliardi
							G06F 17/30864 709/224
				2011/0014972	A1	1/2011	Herrmann
				2011/0016225	A1	1/2011	Park
				2011/0047413	A1	2/2011	McGill
				2011/0058675	A1	3/2011	Brueck
				2011/0060649	A1	3/2011	Dunk
				2011/0082946	A1*	4/2011	Gopalakrishnan
							1104L 65/4084 709/231
				2011/0119724	A1	5/2011	Damola
				2011/0179435	A1	7/2011	Cordray

(56)

References Cited

U.S. PATENT DOCUMENTS

2011-0196943 A1 8/2011 Bornstein
 2011-0202593 A1 8/2011 Vaderna
 2011-0218957 A1 9/2011 Coon
 2011-0235103 A1 9/2011 Ito
 2011-0252138 A1 10/2011 Ahuja
 2011-0258049 A1 10/2011 Ramer
 2011-0296048 A1 12/2011 Knox
 2011-0314130 A1 12/2011 Strasman
 2012/0007866 A1 1/2012 Tahan
 2012/0047542 A1 2/2012 Lewis
 2012/0093098 A1 4/2012 Charbit
 2012/0110167 A1 5/2012 Joch
 2012-0124179 A1 5/2012 Cappio
 2012-0166430 A1 6/2012 Bakalov
 2012-0173753 A1* 7/2012 Moorthy H04N 21:23406
 709:231
 2012-0178426 A1* 7/2012 Filipov H04T 12/281
 455/414.1
 2012-0198492 A1 8/2012 Dhruv
 2012-0204068 A1 8/2012 Ye
 2012-0204153 A1 8/2012 Peterson
 2012/0209717 A1 8/2012 Henry
 2012/0219271 A1 8/2012 Vunic
 2012/0221421 A1 8/2012 Hammad
 2012-0226734 A1 9/2012 Poesse
 2012-0231885 A1 9/2012 Speer, II
 2012-0240176 A1* 9/2012 Ma H04N 21:23439
 725:114
 2012-0259698 A1 10/2012 Yurow
 2012-0265596 A1 10/2012 Mazed
 2012-0278496 A1 11/2012 Hsu
 2012-0311126 A1 12/2012 Jadallah
 2013/0024364 A1 1/2013 Shrivastava
 2013/0024371 A1 1/2013 Hariramani
 2013-0046861 A1 2/2013 Biderman
 2013-0067052 A1 3/2013 Reynolds
 2013-0067109 A1 3/2013 Dong
 2013/0091249 A1 4/2013 McIlugh
 2013/0094445 A1 4/2013 De Foy
 2013/0124724 A1 5/2013 Madgwick
 2013/0132605 A1 5/2013 Kocks
 2013-0142129 A1* 6/2013 Rinne H04W 24:00
 370:329
 2013-0144785 A1 6/2013 Karpenko
 2013-0151673 A1 6/2013 Gouache
 2013-0151687 A1 6/2013 Mooneyham
 2013/0159531 A1 6/2013 Katyal
 2013/0166332 A1 6/2013 Hammad
 2013-0198328 A1* 8/2013 Green H04L 67/02
 709:217
 2013-0198361 A1 8/2013 Matsuzaki
 2013-0218765 A1 8/2013 Hammad
 2013-0304642 A1 11/2013 Campos

2013/0305299 A1 11/2013 Bergstrom
 2013/0326024 A1 12/2013 Chen
 2013/0336221 A1 12/2013 Damola
 2014/0019352 A1 1/2014 Shrivastava
 2014/0108671 A1 4/2014 Watson
 2014/0129303 A1 5/2014 Aiglstorfer
 2014/0143089 A1 5/2014 Campos
 2014/0149557 A1 5/2014 Lohmar
 2014/0150046 A1* 5/2014 Epstein H04N 21/23439
 725/126
 2014/0198641 A1 7/2014 Perkuhn
 2014/0245359 A1 8/2014 De Foy
 2014/0324690 A1 10/2014 Allen
 2014/0330980 A1 11/2014 Richardson
 2014/0337175 A1 11/2014 Katzin
 2014/0348049 A1 11/2014 Kim
 2015/0026239 A1 1/2015 Hofmann
 2015/0067185 A1 3/2015 Tamblin
 2015/0088607 A1 3/2015 Georgoff
 2015/0112774 A1 4/2015 Georgoff
 2015/0134431 A1 5/2015 Georgoff
 2015/0201042 A1 7/2015 Shah
 2015/0220989 A1 8/2015 Hayes
 2015/0302456 A1 10/2015 Rego
 2015/0334153 A1 11/2015 Koster
 2015/0379554 A1 12/2015 Copeland
 2016/0055583 A1 2/2016 Liberty
 2016/0234293 A1 8/2016 Berger
 2016/0292731 A1 10/2016 Khwaja
 2016/0328272 A1 11/2016 Ahmed
 2017/0161717 A1 6/2017 Xing
 2017/0289188 A1 10/2017 Shaikh
 2017/0339222 A1 11/2017 Newton

OTHER PUBLICATIONS

"Simple Opportunistic Routing Protocol for Wireless Mesh Networks"—
 Rozner et al. University of Texas at Austin. Aug. 2011 <http://ericzoner.com/papers/wmesh06.pdf>.
<http://www.cs.berkeley.edu/~kubitron/articles/techreview.pdf> "The Internet Reborn" Wade Roush, Technology Review, Oct. 2003.
 Theotokis et al., A Survey of Peer-to-Peer Content Distribution Technologies, ACM Computing Surveys, vol. 36, No. 4, Dec. 2004, pp. 335-371.
 Wade Roush, Technology Review, Published by MIT, TR10: Peering into Video's Future, Mar. 12, 2007. http://technologyreview.com/printer_friendly_article.aspx?id=18284.
 "Firefly-Inspired Heartbeat Synchronization in Overlay Networks" Binci et al., University of Bologna, Italy, Feb. 2007 <http://www.cs.unibo.it/babaoglu/courses/cas06-07/papers/pdf/fireflies.pdf>.
 Coppens et al., Design and Performance of a Self-Organizing Adaptive Content Distribution Network, 2006.
 Flo et al., "Virtual URIs for Browsing & Searching Large Information Spaces", WebNet Journal, pp. 38-43, p. 66, Jan.-Mar. 1999.

* cited by examiner

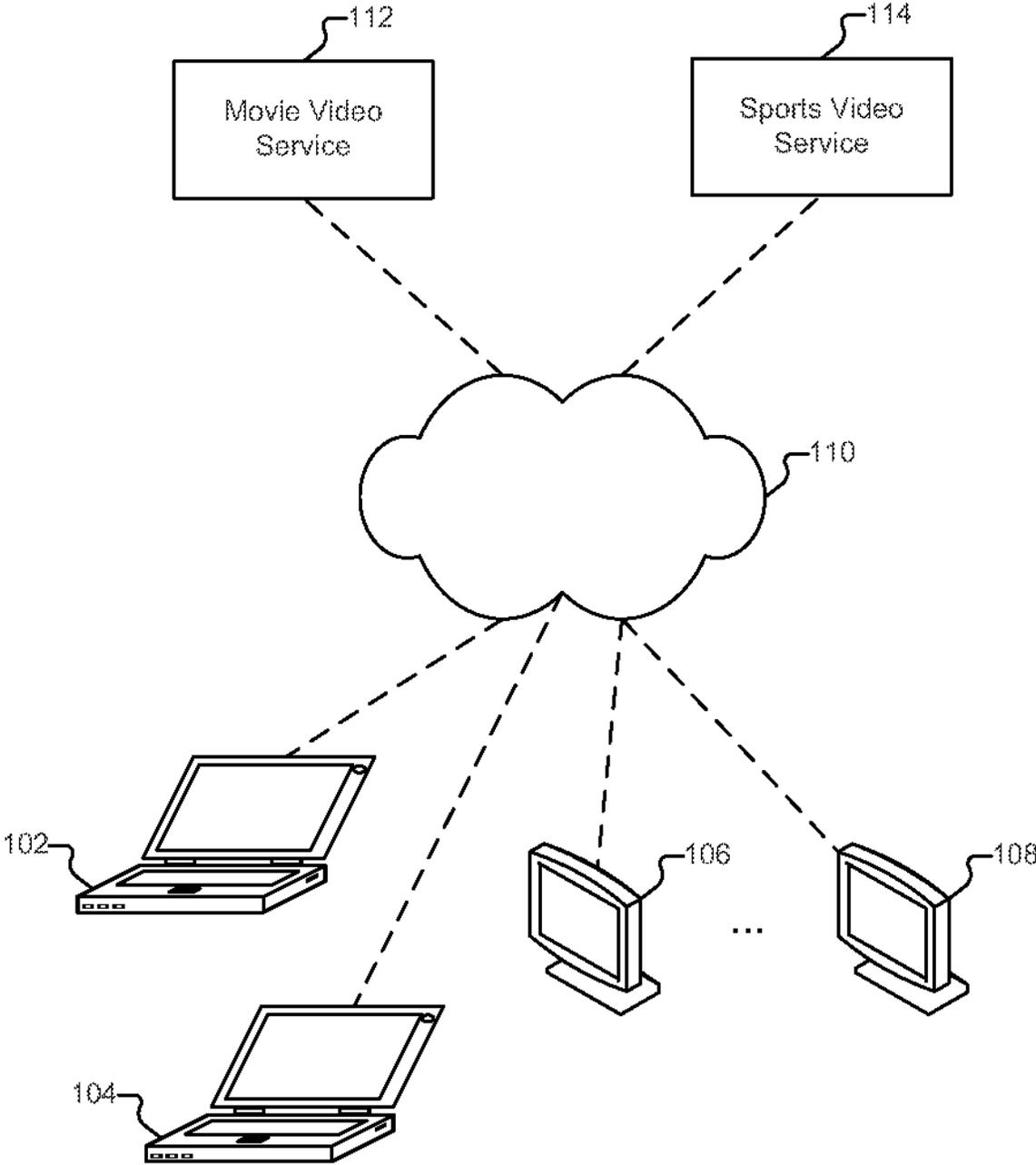


FIG. 1

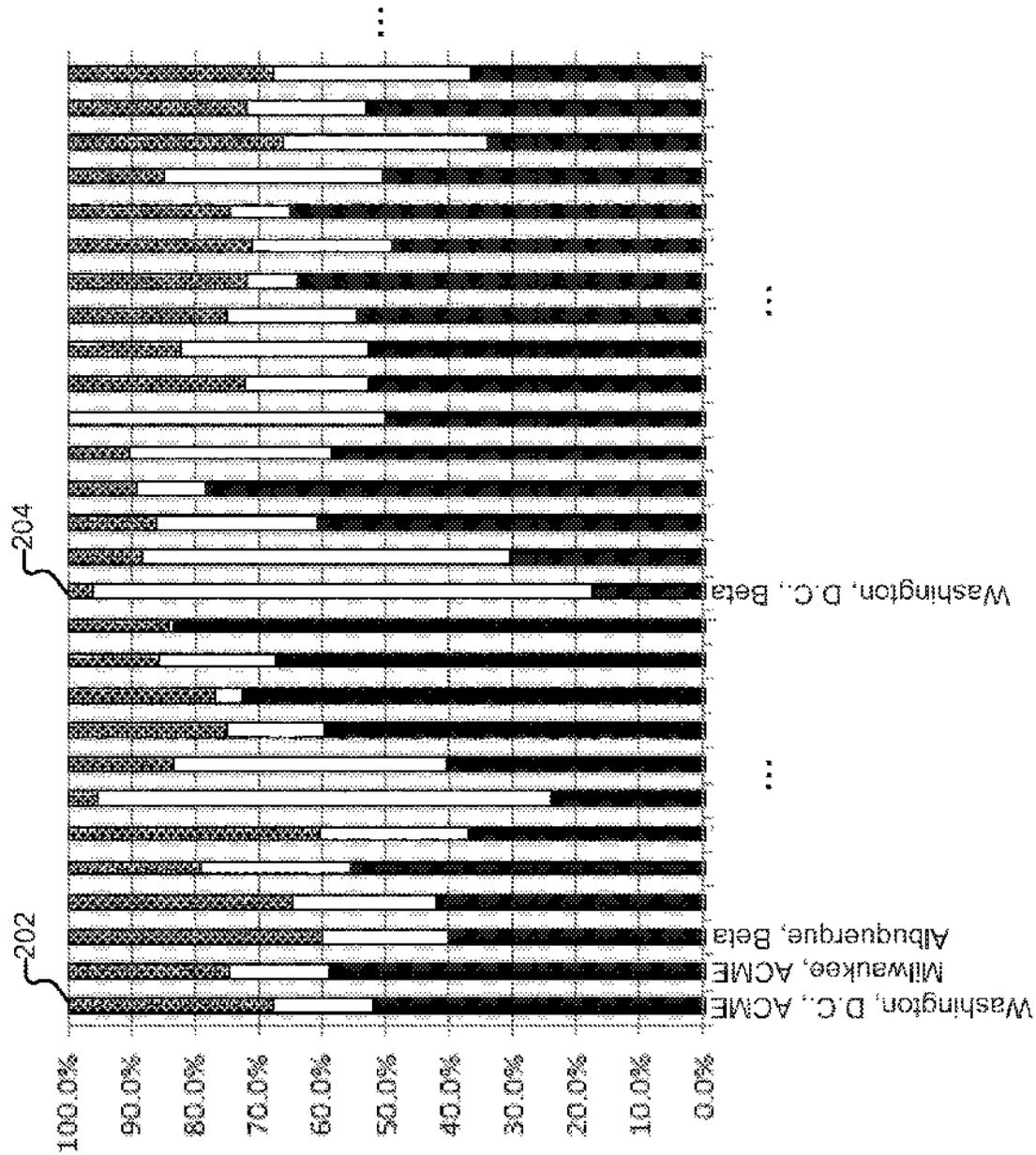


FIG. 2

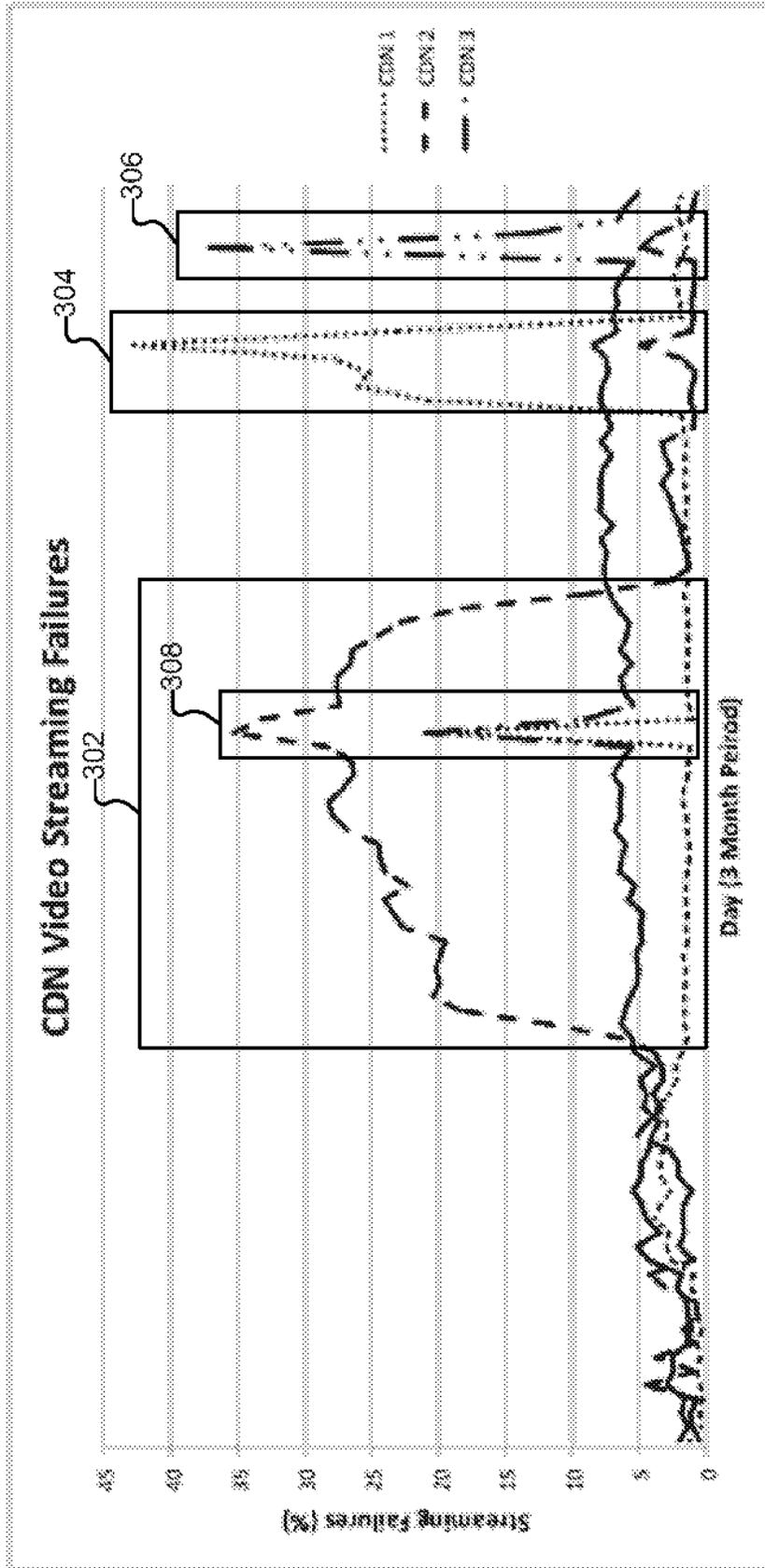


FIG. 3

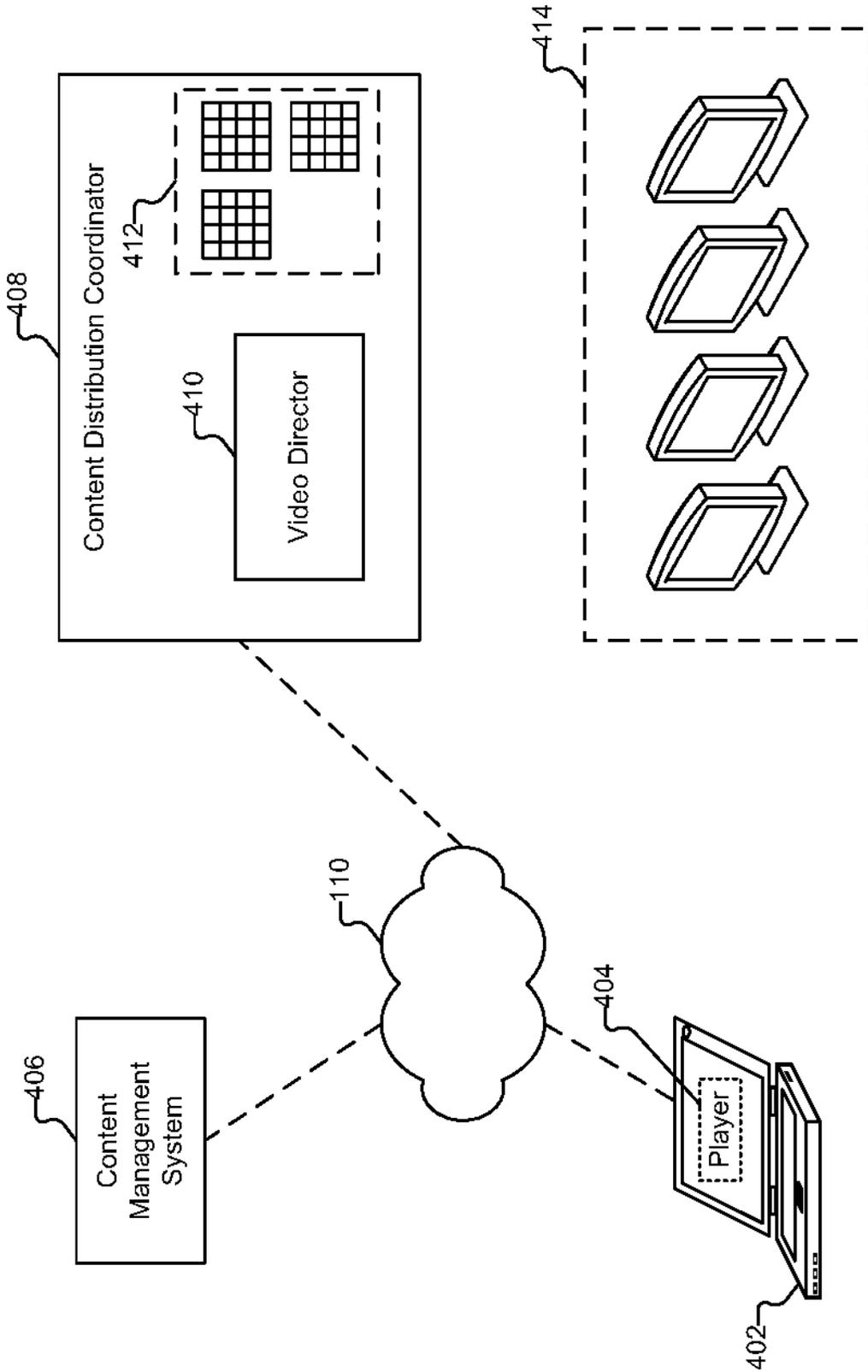


FIG. 4

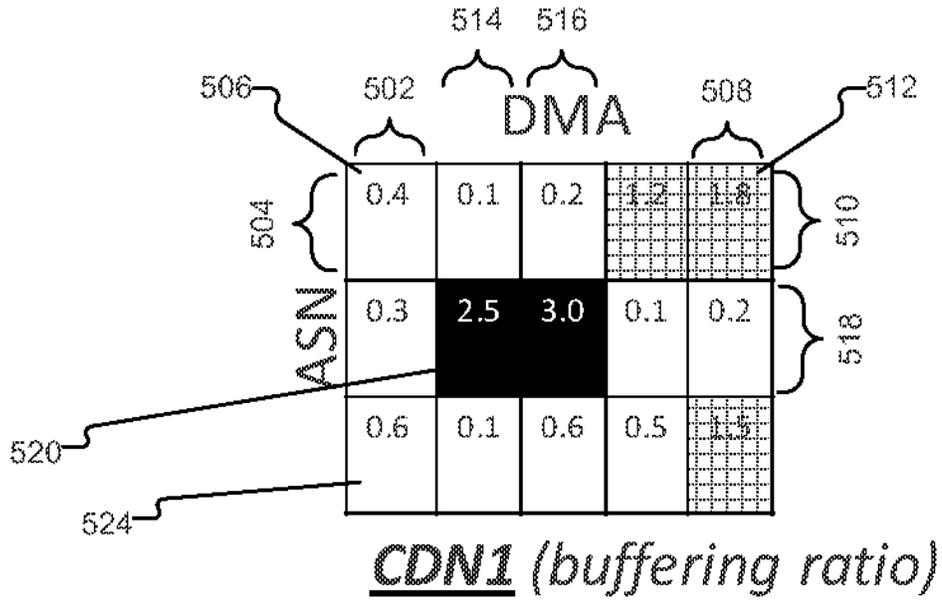


Figure 5A

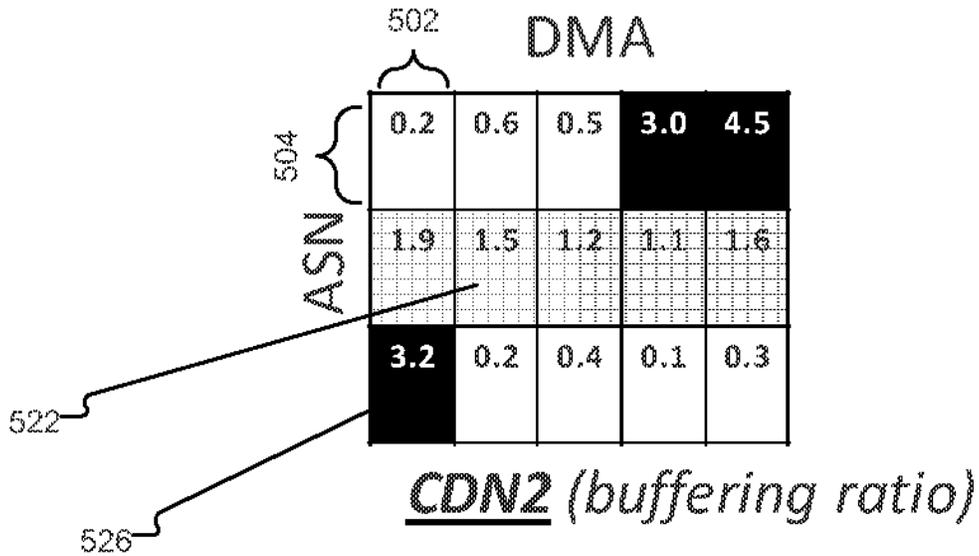


Figure 5B

602

DMA

ASN

0.2 (CDN 2)	0.1 (CDN 1)	0.2 (CDN 1)	1.2 (CDN 1)	1.8 (CDN 1)
0.3 (CDN 1)	1.5 (CDN 2)	1.2 (CDN 2)	0.1 (CDN 1)	0.2 (CDN 1)
0.6 (CDN 1)	0.1 (CDN 1)	0.4 (CDN 2)	0.1 (CDN 2)	0.3 (CDN 2)

Best CDN (buffering ratio)

Figure 6

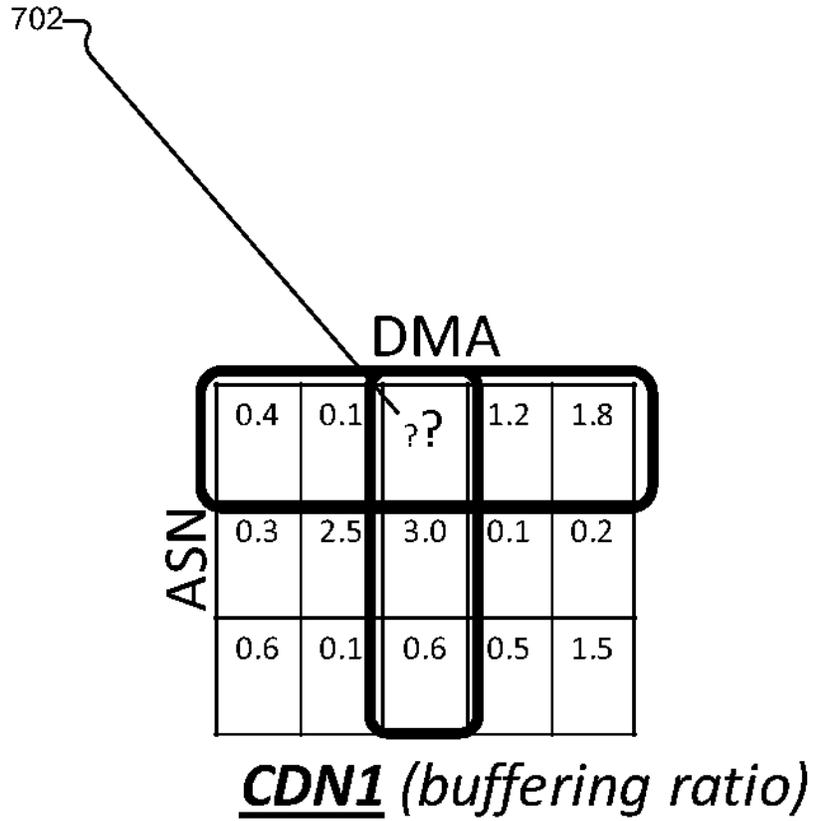


Figure 7

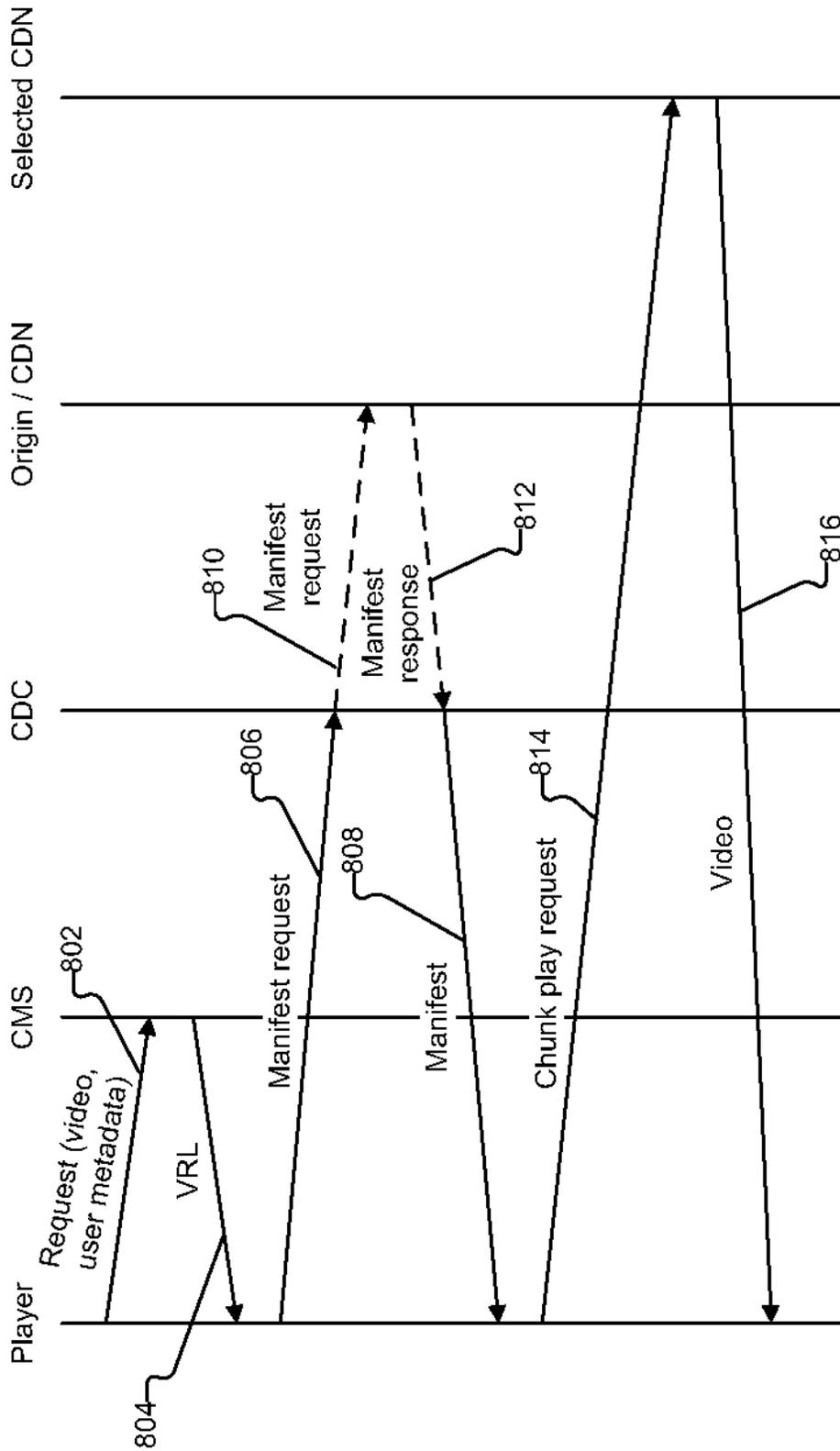


FIG. 8

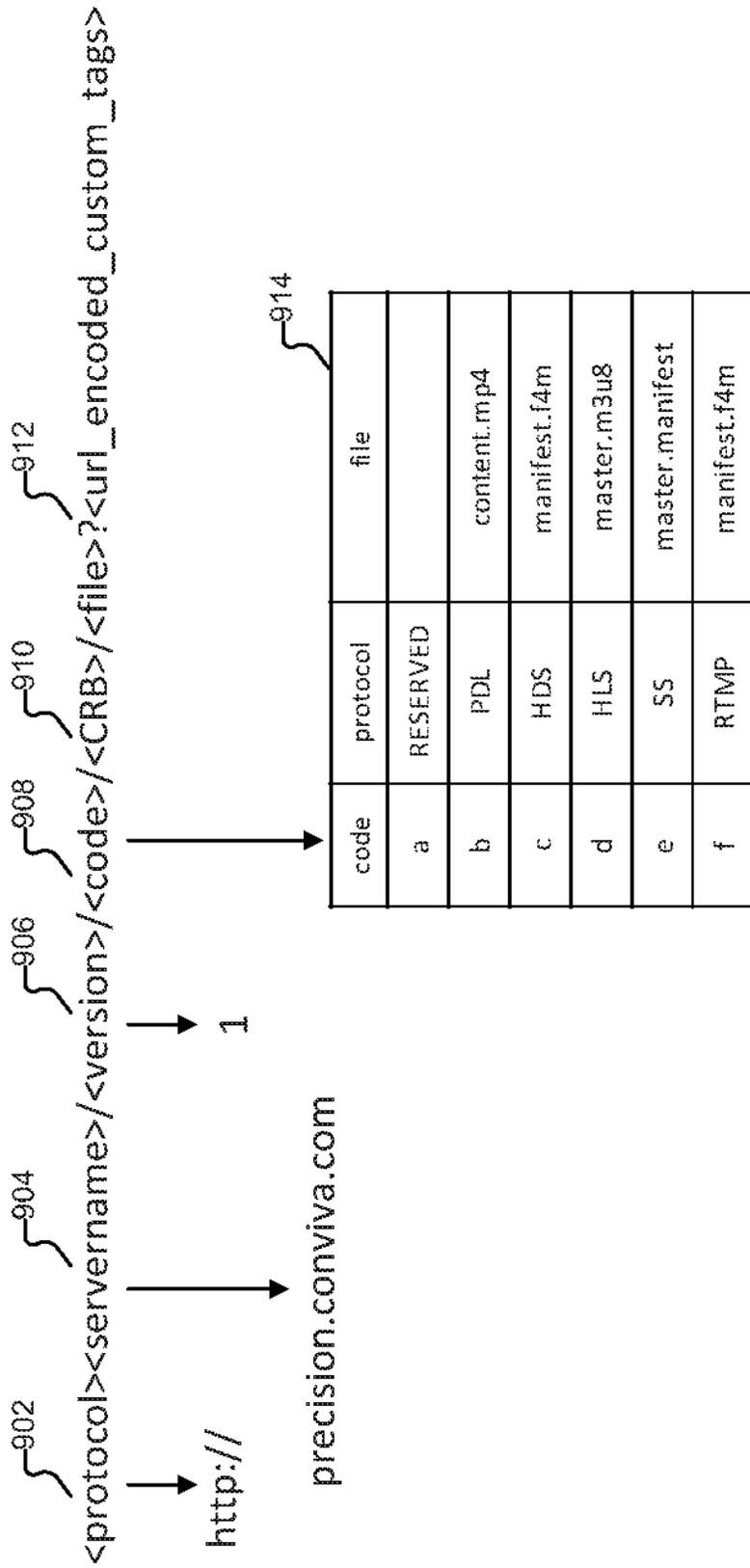


FIG. 9

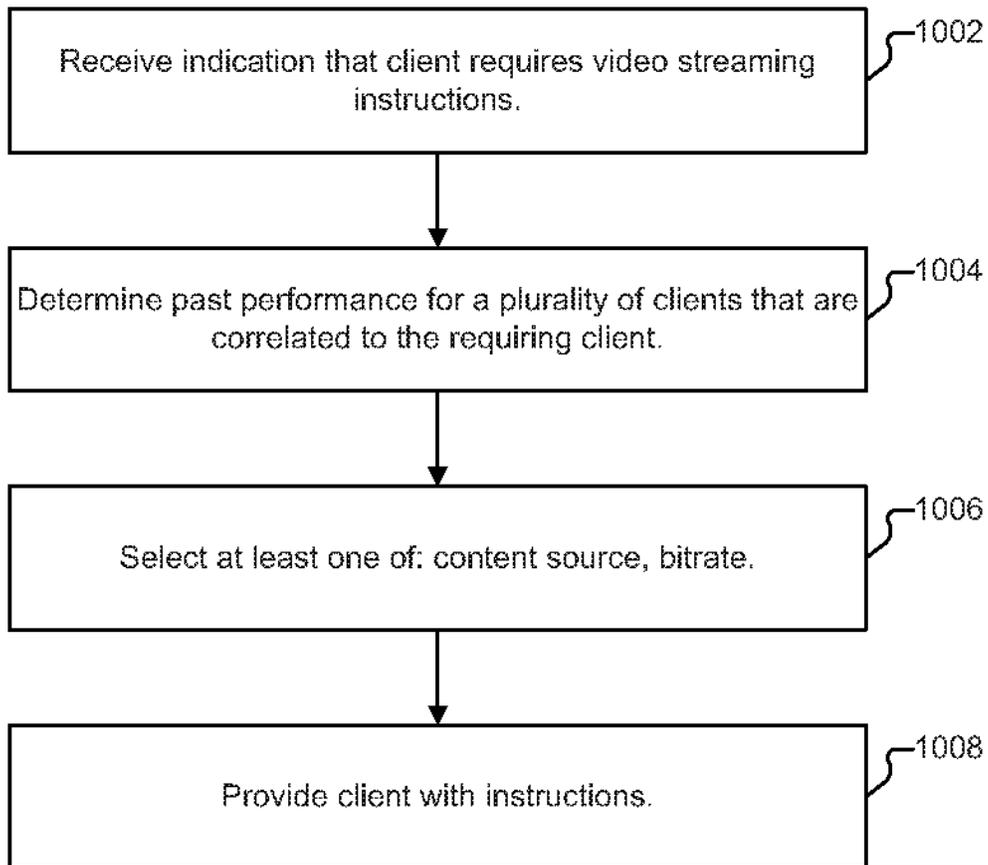


FIG. 10

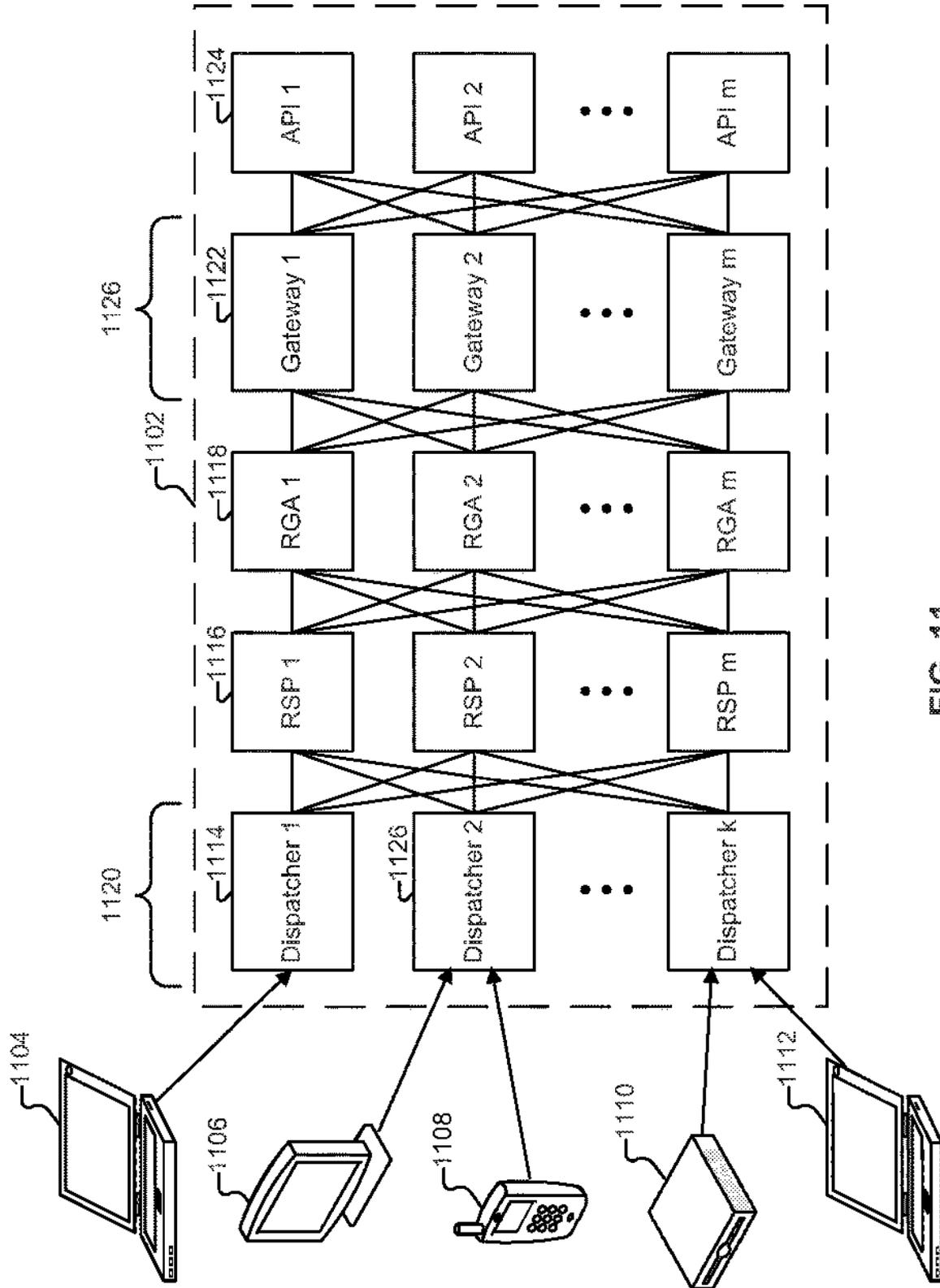


FIG. 11

SOURCE ASSIGNMENT BASED ON NETWORK PARTITIONING

CROSS REFERENCE TO OTHER APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 13/752,252, entitled SOURCE ASSIGNMENT BASED ON NETWORK PARTITIONING filed Jan. 28, 2013 which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Patent Application No. 61/697,225 entitled SOURCE ASSIGNMENT BASED ON NETWORK PARTITIONING filed Sep. 5, 2012 which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Users are increasingly using networks such as the Internet to access content, such as video files and live streaming/video on demand content, via client machines. Such content is often large, time sensitive, or both. As demand for such content increases, there are ongoing challenges in distributing that content efficiently and with high quality. As one example, when a client requests content (e.g., a particular television episode), the content may be available in multiple bitrates from multiple sources. Unfortunately, the client may not receive the content from the most appropriate source, or in the most appropriate bitrate.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. 1 illustrates an environment in which content is distributed.

FIG. 2 depicts an example of the performance associated with three different content delivery networks (CDNs).

FIG. 3 depicts examples of both uncorrelated and correlated quality issues.

FIG. 4 illustrates an environment in which source assignment based on partitioning is performed.

FIGS. 5A and 5B depict examples of partitioning pertaining to two CDNs.

FIG. 6 illustrates a table depicting a selection of the best CDN for a given DMA/ASN pair.

FIG. 7 depicts an example of a partition that includes limited information.

FIG. 8 depicts a call flow used, in some embodiments, to direct a video player to a video source.

FIG. 9 illustrates an example of a Virtual Resource Locator.

FIG. 10 illustrates an embodiment of a process for directing a video player to a video source.

FIG. 11 is an illustration of an environment in which status information is received and processed.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form

that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term ‘processor’ refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. 1 illustrates an environment in which content is distributed. In the example shown, client devices are used to access content, such as audiovisual content (e.g., movies, songs, television shows, sporting events, games, images, etc.) that is owned by content owners. Content is stored (or captured) at origin servers, then distributed via other servers, caches, content delivery networks (CDNs), proxies, etc. (collectively, “content sources”). Examples of CDNs include Akamai, Limelight Networks, and Level 3 Communications. Content sources employ a variety of technologies and include HTTP, Adobe Flash Media, and Microsoft Internet Information Service servers. Examples of clients include personal computers, laptops, cellular phones/personal digital assistants, and other types of information appliances such as set-top boxes, game consoles, broadband routers, file servers, video servers, and digital video recorders, as applicable.

In the example shown in FIG. 1, client device 102 is operated by a user hereinafter referred to as Alice. Alice lives in the Washington, D.C. area. Her Internet Service Provider (ISP) is hereinafter referred to as “ACME Internet.” Suppose Alice wishes to watch a movie, streamed to client device 102 via one or more networks (represented as a single network cloud 110), from movie video service 112. Client device 102 includes a web browser application. Alice uses the web browser application to navigate to a portal owned by movie video service 112, such as “http://movievideoservice12.com/welcome.” Her request for a particular movie is directed to a particular CDN and ultimately to a particular CDN node. Her client device begins streaming the content from the CDN node, which is in turn rendered in her browser (e.g., via a Flash or Silverlight player).

The CDN (and/or CDN node) assigned to Alice’s client device can be selected in a variety of ways. As one example, a CDN can be selected based on current estimated load, or for financial reasons (e.g., where traffic to one CDN is preferred over another). Further, a given CDN node can be assigned (e.g., by the CDN) based on its distance to Alice (i.e., with the CDN node that is the fewest hops away from

her client device being selected). As another example, a CDN node may be assigned based on a load balancing scheme, irrespective of or in addition to its distance from Alice. As will be described in more detail below, the CDN node closest to Alice (or that would be assigned based, e.g., on a round robin load balancing scheme) may or may not offer her the best viewing experience relative to other CDN nodes having been assigned (including those belonging to other CDNs). Using the techniques described herein, a decision of which content source should be used, and/or what bitrate should be used, can be made. As will be described in more detail below, such decisions can be made both at the time Alice initially requests content, and also as an ongoing decision. Also as will be described in more detail below, such decisions can be made for Alice's client based on historical performance information collected from other client devices that share some commonality with her client.

Client device 104 is operated by a user hereinafter referred to as Bob. Bob also lives in Washington, D.C., and also likes to stream movies from service 112. Bob's ISP is "Beta Telecommunications." Many additional client devices (represented by client devices 106 and 108) connect to service 112, from a variety of geographical locations (e.g., Milwaukee and Albuquerque), via a variety of ISPs. Client devices also connect to other services, such as sports video service 114.

Quality Issue Examples

FIG. 2 depicts an example of a performance metric associated with three different content delivery networks (CDNs). In particular, FIG. 2 depicts the percentage of time a given CDN provides the best performance for a particular combination of a Designated Market Area (DMA) and an Autonomous System Number (ASN). For a given combination, performance with respect to each of three CDNs is shown as a stack, where CDN 1 is depicted as the black portion of the stack, CDN 2 is depicted as the white portion of the stack, and CDN 3 is represented as the crosshatched portion of the stack. A variety of measures of performance can be used—whether individually or in combination. Examples include, but are not limited to: (1) the number/percentage of client connection failures, (2) how quickly a video begins playing on a client (e.g., number of seconds until playback starts), (3) the amount of time clients spend buffering (e.g. as a buffering ratio), and (4) whether the maximum bandwidth of the client is used. The example shown depicts an aggregate assessment of quality (e.g., taking into account all four measures by weighting each equally). Other aggregation techniques can also be used. For example, buffering ratio can be weighted more heavily than other factors. As another example, not all factors need be aggregated—some can be omitted.

Using bar 202 as an example: approximately 51% of the time, an arbitrary client located in Washington, D.C. and serviced by ACME will have the best video streaming experience when connected to CDN 1. CDN 2 will provide the best experience to the client approximately 16% of the time, and CDN 3 will provide the best experience to the client approximately 33% of the time. Using bar 204 as another example: approximately 17% of the time, an arbitrary client also located in Washington, D.C., but serviced by Beta will have the best video streaming experience when connected to CDN 1. CDN 2 will provide the best experience to the client approximately 76% of the time, and CDN 3 will provide the best experience to the client approximately 5% of the time. Thus, even though Alice and Bob are both located in Washington, D.C., their respective clients will likely experience different performance when connect-

ing to the same CDN (e.g., CDN 1), because they have different ISPs. Specifically, Alice's client is most similar to clients represented by bar 202, and Bob's client is most similar to clients represented by bar 204.

FIG. 3 depicts examples of both uncorrelated and correlated quality issues. In particular, FIG. 3 depicts, over a three month period, the percentage of connection failures experienced by clients attempting to connect to each of the three CDNs. During period 302, CDN 2 had the worst performance (i.e., had the most connection failures). During period 304, CDN 1 had the worst performance, and during period 306, CDN 3 had the worst performance. Further, during period 308, all three CDNs experienced a failure spike (i.e., a correlated failure). One reason for such a spike is a period of high load (e.g., a holiday weekend or national sporting event when many people watch videos at the same time).

Determining Assignments

FIG. 4 illustrates an environment in which source assignment based on partitioning is performed. Suppose an operator of a client device, such as client device 402, wishes to watch a video using video player application 404. In some embodiments, the player application is a dedicated application. In other embodiments, the player application is provided by a web browser application. In the example shown, client device 402 connects to Content Management System (CMS) 406 (e.g., via one or more networks 110) and requests the content. As will be described in more detail below, information about the client device (e.g., its IP address, User Agent string, what type of device it is, what operating system it is running, firmware version, browser type/version, player version, display size, and/or any other applicable attributes) is used to determine a set of other client devices that are correlated with client 402. Historical performance of devices in that set is evaluated to determine a content source and/or bitrate for client 402. Instructions for contacting the content source and obtaining the content are then provided to client 402. In some embodiments, content management system 406 contacts content distribution coordinator 408 to determine the set of instructions to provide to device 402. In other embodiments (e.g., described in conjunction with FIG. 8), content management system 406 redirects client 402's request (e.g., via HTTP redirect).

In some embodiments, content management system 406 and content distribution coordinator 408 are operated by the same entity. For example, an online movie streaming service could operate both content management system 406 and content distribution coordinator 408. Content distribution coordinator can also be operated by other entities. For example, content distribution coordinator 408 can be operated by an operator of a CDN. Content distribution coordinator 408 can operate independently of content management system 406 and/or independently of a CDN. As one example, the owner of content distribution coordinator 408 could provide services to multiple content management systems (e.g., one associated with movie service 112 and one associated with sports video service 114) and/or multiple CDNs.

As will be described in more detail below, quality experienced by clients such as clients 414 is measured and stored by content distribution coordinator 408. The information obtained from/about clients 414 is partitioned (412) and can be used, e.g., by a video direction engine 410 to predict a performance quality likely to be experienced by a client device if it is assigned a given content source.

FIGS. 5A and 5B depict examples of partitioning pertaining to two CDNs. In particular, FIG. 5A shows, for clients located in one of five DMAs, and located within one of three

ASNs, the buffering ratio experienced with respect to CDN 1. For example, clients in DMA 502 and in ASN 504 experience a buffering ratio of 0.4 (506). Clients in DMA 508 and in ASN 510 have a worse (higher) buffering ratio—1.8 (512). Clients in either of DMA 516 or 518, and also in ASN 514 have the worst buffering ratios—2.5 and 3.0, respectively.

FIG. 5B shows, for the same ASN/DMA partitioning as FIG. 5A, the buffering ratio experienced with respect to CDN 2. For some clients, CDN 2 will provide better performance than CDN 1. As one example, compare the values in regions 520 and 522. For some clients, CDN 2 will provide worse performance than CDN 1. As one example, compare the values in regions 524 and 526.

When a client, such as client device 402, requests content, information such as is shown in FIGS. 5A and 5B can be used (e.g., by content distribution coordinator 408) to determine whether to assign the client CDN 1 or CDN 2 as a content source. Other decisions can also be made using similar information, such as determining which CDN node to assign and/or determining which bitrate to assign. Further, once a video is being played on a client, information such as is shown in FIGS. 5A and 5B can be used to assess the predicted impact switching the client from one content source to another (or adjusting the bitrate) would have.

The values depicted in FIGS. 5A and 5B pertain to a single quality metric—buffering ratio. Data for other metrics can also be used in partitioning (e.g., pertaining to failed connection attempts). Further, the data can be combined in a variety of ways. For example, a bitrate dimension can be added (i.e., further partitioning the clients based on their bitrates). Other attributes can also be used to partition the data, such as client type (e.g., desktop computer vs. tablet computer), browser (e.g., Internet Explorer vs. Firefox), and display size (e.g., 17" vs. 10").

FIG. 6 illustrates a table depicting a selection of the best CDN for a given DMA/ASN pair. In particular, each cell in FIG. 6 corresponds to whichever of CDN 1 or CDN 2 provides a better performance quality for the DMA/ASN pair represented by the cell. For example, compare clients in DMA 502/ASN 504 pair with respect to CDN 1 and CDN 2. Such clients have a predicted buffering ratio of 0.4 if connected to CDN 1, and a predicted buffering ratio of 0.2 if connected to CDN 2. Accordingly, CDN 2 is a better choice for performance reasons (as indicated in cell 602).

In the examples shown in FIGS. 5A and 5B, the values included in a given cell (e.g., the value of 0.4 appearing in cell 506) represent the buffering ratio experienced by the average client included in the given partition (i.e., the score is determined as an average of the buffering ratios of all clients in the set). Other techniques can also be used to assess the quality scores. For example, a distribution of buffering ratios can be determined, and only those clients having scores in the second/third quartile used as predictors, or of specific percentiles (e.g., 80th percentile).

In some cases, a given partition may have data on few or no clients. Such a scenario is depicted in FIG. 7. Specifically, information pertaining to clients in cell 702 is limited. In some embodiments, when a given partition has insufficient information to make a reliable prediction, a larger partition is used. For example, the quality experienced by all clients in the same ASN can be aggregated, and/or the quality experienced by all clients in the same DMA can be aggregated, and used to predict which CDN would be best for a client having the attributes of cell 702.

FIG. 8 depicts a call flow used, in some embodiments, to direct a video player to a video source. The video distribu-

tion depicted in FIG. 8 makes use of HTTP chunking delivery techniques. In HTTP chunking, content is divided into chunks (e.g., with each chunk being 2-10 seconds of data) and then distributed as files via CDNs. Clients download chunks and stitch them together, such that a viewer of the content sees a continuous video stream. A CMS provides metadata associated with a video stream, including the information required to deliver the content, e.g., the URL of a manifest file. In particular, the manifest file contains the URL names of the chunks encoding the content, and can also include the URL/names of other manifest files. Each chunk is encoded at a particular bitrate, and it is stored on a particular CDN or (origin) server. Manifest files can be themselves distributed through CDNs.

Referring now to FIG. 8, an appropriate CDN/bitrate can be initially selected for a client such as client 402, and the selection(s) can also be revised as needed. The call flow is as follows: first, the client sends a request to the CMS (802). In addition to the video being requested, additional information about the user/client is included in the request, or is otherwise obtained by the CMS about the client. The additional user/client information may be limited (e.g., just an IP address, or an IP address and a User Agent string) but can also include a variety of attributes (e.g., client type and display size). As one example, the CMS may be publicly accessible, and not require a user to have an account to watch videos. Further, that CMS may make use of standard video playback capabilities provided by a browser application. In such a scenario, limited information about the client may be provided at 802. As another example, access to the CMS may require a user account, and/or may require a customized player application that can obtain more detailed information about the user/client and provide it at 802. The CMS replies with metadata, including a URL to a manifest file (804).

To control the selection of the CDN and bitrate, an indirection mechanism is used. Specifically, instead of including a traditional URL to the manifest file, the CMS returns a URL that points the client to the CDC. This URL is also referred to herein as a "Virtual Resource Locator" or VRL. The VRL (described in more detail in conjunction with FIG. 9) includes the content name, and encodes other information (e.g., about the device). When the CDC is accessed via a VRL (806), the CDC dynamically generates and sends to the client (808) a manifest file based on the following information: (1) information embedded in the URL; (2) playback quality experienced by other similar clients watching at the same time; (3) a Global IP address; (4) bitrates at which the content is encoded, and the CDNs or origin servers where the content is stored; and (5) Global policies such as the fraction of clients allowed on each CDN, or the maximum arrival rate of new clients at a given CDN. Information associated with (4) can be communicated offline by the content provider, embedded in the VRL, or obtained from existing manifest files of the content (specified in VRL) that are already stored at various CDNs (see dotted lines 810 and 812).

In the example of FIG. 8, "similar" clients include those clients sharing a subset of attributes with the client making the request. Examples of these attributes include: DMA, ASN, content, content provider, device type, etc. The higher the overlap between the values of the attributes of two clients, the higher the "similarity" of the clients. The quality experienced by similar clients is used by the CDC to accurately predict the quality that the requesting client will experience when streaming at a given bitrate and from a given CDN.

Once the client receives the manifest file (808), the client uses (814) the information in this file to download (816) and start playing the video chunks. In various embodiments, content source/bitrate decision-making is performed on an ongoing, periodic basis, and a given client can be provided with updated instructions midstream. As one example, after ten consecutive chunks are received by a client, the client can be provided with a new VRI 804, or a manifest that includes a VRI (i.e., causing CDC to re-evaluate the client's content source/bitrate) and send revised instructions if warranted. As another example, in some embodiments, clients run dedicated video player applications and/or applications that otherwise natively support a feature that allows the client to periodically evaluate its performance and request revised instructions, e.g., when quality issues are experienced midstream.

FIG. 9 illustrates an example of a Virtual Resource Locator. As explained above, a VRL is provided by a content provider's CMS to a client requesting content, and is used by the client to invoke the services of the CDC. As shown in FIG. 9, in some embodiments, a VRL includes six regions. The first (902) indicates the protocol being used to perform the request. One example of a protocol is HTTP. The second (904) indicates the location of the CDC (e.g., a DNS name of the CDC). The third (906) indicates the version of VRL being used. One example is version "1." The fourth (908) is a code that indicates a protocol type associated with the data to be retrieved. A chart of example codes is included in FIG. 9 (914). The fifth (910) is a request block that specifies required parameters, such as the asset (content) name, customer ID, bitrates at which the content is stored, and CDNs at which the content is stored. The final region includes the name of what is being requested, and a set of optional parameters encoded as key-value pairs. Examples of such parameters include content metadata (e.g., episode, show, season), content type (e.g., advertisement, short video on demand clip, or live stream).

The following is a list of example optional parameters that can be embedded in portion 912.

tok ("securityToken"): A token generated by the CDC that is passed out-of-band to the video CMS and is validated at request time.

an ("assetName"): a human-readable string describing the video asset. Maximum length: 127 bytes.

at ("assetType"): a string identifying the type of the content. May be one of the following values: ON DEMAND, LIVE EVENT, LINEAR LIVE, EVENT REPLAY, OTHER.

dvt ("deviceType"): a string indicating the form-factor type of the device. May be one of the following: PC, MOBILE, CONSOLE, SETTOP, SMART TV, OTHER.

dvb ("deviceBrand"): a string that names the brand of the device. Maximum length is 64 bytes. For example, "Roku," "Microsoft."

dvm ("deviceModel"): a string that defines the model of the device. Maximum length is 64 bytes. For example, "Xbox 360."

dvv ("deviceVersion"): a string representing the firmware version number of the device. Maximum length is 64 bytes.

dvc ("deviceConnection"): a string describing the last-mile access connection the video device has to the Internet. May have one of the following values: T1, CABLE, DSL, WIFI, 4G, 3G, OTHER.

dvid ("deviceId"): a string defining a unique device identifier. Must be less than 127 bytes. For example, UDIDs for iOS devices.

drm ("drmType"): The type of digital rights management technology used. May be one of the following: MICROSOFT_PLAY_READY, APPLE_FAIR_PLAY, ADOBE_ACCESS, MARLIN, ADOBE_PHDS, ADOBE_PHLS, OTHER, NONE.

os: a string naming the operating system in which the video is playing. Maximum length is 64 bytes.

osv ("osVersion"): a string indicating the full version number of the operating system indicated under "os," if any. Maximum length is 64 bytes.

vid ("viewerId"): a string identifying the video sessions of a single viewer, without collecting any personally identifiable information. Typically a hashed/obfuscated version of a Subscriber ID. Maximum length is 127 bytes.

<other customer meta-data> For example:

"serviceType": "SkyGo"

"show": "PopIdol"

"episode": "S8:115"

FIG. 10 illustrates an embodiment of a process for directing a video player to a video source. In some embodiments, process 1000 is performed by content distribution coordinator 408. As explained above, content distribution coordinator 408 can be operated independently, but can also be operated by an operator of a content management system (e.g. CMS 406) or other entity, such as an operator of a CDN. Accordingly, in some embodiments scenarios, other components may perform process 1000, instead of or in addition to CDC 408. Further, multiple components may cooperate to perform process 1000.

The process begins at 1002 when an indication that a client requires video streaming instructions is received. In some embodiments, portion 1002 of the process is performed when CDC 408 is contacted by client 102 as a result of client 102 receiving a VRL from CMS 406 and being directed to CDC 408. In other embodiments, portion 1002 of the process is performed when a CMS receives a request for a resource, and contacts CDC 408 on behalf of the requesting client.

At 1004, past performance for a plurality of clients that are correlated to the requiring client is determined. As one example, portion 1004 of the process is performed when CDC 408 receives attributes about the requesting client, determines (e.g., from partitions 412) which other clients are most similar to the requesting client, and evaluates those clients performance history. Suppose a determination is made that the requesting client is in a particular DMA and ASN. The performance history of other clients in the same DMA and ASN is examined at 1004.

In some embodiments, a video director engine included in CDC 408 is configured to perform the actions of (a) determining which set of clients are most similar to the requesting client, and (b) evaluating their performance history. The clients most similar to the requesting client need not be requesting (or have requested) the same content as is requested by the client. For example, other clients in the same DMA and ASN may be (have been) watching other movies from movie video service 112. As a further example, correlated clients may be (have been) watching video provided by other services, such as sports video service 114. CDC 408 can be configured to collect statistical information from a wide variety of clients, including clients associated with multiple services (e.g. services 112 and 114), and can use the collected information to aid in decision-making.

At 1006, at least one of: a content source, and a bitrate, is determined based at least in part on the past performance information examined at 1004. As one example, the correlated clients' performance with respect to different CDNs

can be examined (e.g., at 1004), and a decision that the requesting client should obtain content from a particular CDN can be made based on the examination (e.g., as described in conjunction with FIG. 6). As another example, the current average bitrate of the correlated clients can be selected for use with the requesting client. A bitrate selection performed in this way can have a particularly positive impact on the experience of a user viewing short video clips, as contrasted with picking a default bitrate and adjusting the bitrate up or down empirically (either while the video plays, or as part of an initialization). As yet another example of the processing performed at 1006, a particular CDN node can be determined for the client (e.g., where an owner of a CDN network is operating CDC 408, or is otherwise using CDC 408 to assist in CDN node selection).

In some embodiments, the decision made at 1006 is based wholly, or predominantly, on the performance information of correlated clients. As one example, in the case of a video service that offers short clips and does not require users to log in, very little, if any, information may be known about the specific requesting client. Further, as explained above, the requesting client may be running a standard browser application that does not provide details about the client/the client's performance to other entities such as CMS 406 or CDC 408. In other embodiments, the performance information of correlated clients is one of multiple factors in deciding a content source and/or bitrate for the client. For example, if CMS has existing information about the client (e.g., because it has stored historical performance information about the client), or if CDC 408 has information about the client (e.g., because the client is already streaming content and/or is reporting statistical information to either of CMS 406 or CDC 408), the decision made at 1006 can be based both on knowledge about the client, as well as knowledge about clients correlated with the client.

Finally, at 1008, instructions are provided to the client. As one example, instructions are provided to the client at 1008 when a CDC provides a manifest to a client (e.g., 808). As another example, where CMS 406 contacts a CDC to obtain instructions on behalf of a client, the CMS provides the obtained instructions to the client at 1008.

Additional Detail Regarding Embodiments of the Content Distribution Coordinator

FIG. 11 is an illustration of an environment in which status information is received and processed. In particular, clients included in the environment provide periodic status reports (also referred to as "heartbeats") containing telemetry data to a content distribution such as CDC 408. In various embodiments, the services provided by the content distribution coordinator are implemented across a scalable infrastructure, particularly in embodiments where telemetry data is received from many clients. In the example shown, the elements contained within dashed region 1102 collectively provide the functionality of content distribution coordinator 408. Each of the layers (e.g., dispatcher layer 1120) is horizontally scalable and their respective components can be implemented on standard commercially available server hardware (e.g., having a multi-core processor, 4G+ of RAM, and Gigabit network interface adaptors) running a typical server-class operating system (e.g., Linux).

Clients 1104-1112 each include a monitoring module that collects status information. As one example, performance information associated with video playback is collected. In other information about the client (e.g., information usable for partitioning as described above). Note that clients 1104-1112 can represent a subset of all clients included in an environment such as is shown in FIG. 1 or FIG. 4. That is,

a client such as client 102 need not include a monitoring module. Instead, in some embodiments, data from those clients with monitoring modules is used to make predictions about other clients (e.g., client 102).

When the monitoring module on a client is activated (e.g., because the client has started an instrumented video player), the client is mapped to a dispatcher server. As one example, when the monitoring module starts, it reads a configuration file that includes a list of dispatcher servers. The monitoring module selects a dispatcher server at random from the list.

A dispatcher server (1114) includes two conceptual modules. The first module implements a communication interface for receiving status information from clients. In some embodiments the module is implemented using an off-the-shelf web server, and allows clients to connect over the HTTP protocol (and also allows clients to securely communicate via SSL). Data received by the first module is passed to the second module. The second module normalizes the data (to a format suitable for further processing) and passes the normalized data to a real-time stream processing component (1116).

The real-time stream processing (RSP) layer includes an optimized software component that processes the telemetry data that it receives from the dispatcher in real-time. A dispatcher sends all heartbeats belonging to the same session to the same RSP component.

In some embodiments, the RSP component is implemented as a continuously running service that reads and processes the telemetry data received from dispatchers via the network over TCP. The telemetry data stream comprises individual records, each of which represents the telemetry data sent by the monitoring module. The RSP component reads network data one record at a time and parses the data into a local data representation. The data received by the RSP component can be stored as in-memory hash tables of records allowing fast execution, and very high throughputs. Since the RSP component does not require old information, it can periodically purge the in-memory hash tables and increase scalability accordingly. In other embodiments, optimized in-memory databases are used.

An example mapping function to map heartbeats having a session identifier "ID" to a particular RSP component "i" as follows:

$$i = \text{hash}(\text{ID}) \bmod m,$$

where hash() is a hash function and "m" is the total number of RSP components.

Once an RSP component parses the data records, it performs two main tasks. First, it performs data filtering. A filter is a logical expression and is installed at each RSP component instance. As one example, the following filter would identify viewers located in San Francisco, connected to ISP SP1, streaming from CDN A, one of two particular shows:

```
(city="San Francisco" AND ISP="SP1" AND
CDN "CDN A" AND ((show "NewsAt10") OR
(show "SundayMagazine")))
```

For each message of incoming telemetry data, the (key, value) pairs in the record are matched against the filter. If the filter is matched, the data is associated with the filter.

The second task performed is to compute snapshots and on-line statistics over the telemetry data matching each filter. One example of a snapshot is the number of players that are in a particular state (e.g., "playing"). The RSP component generates sequences of these snapshots (e.g., one every second). Examples of statistics computed by the RSP component include: the average number of bytes played over all

11

video streams matching a filter over a given time interval (e.g. 10 seconds) and the minimum frames per second experienced by a stream matching a filter over a time interval. Snapshots and statistics are updated continuously, from new telemetry data received from clients.

The RSP component provides its computed snapshots and statistics to a real-time global aggregation component (1118). The real-time global aggregation (RTGA) component aggregates the information provided by the RSP component for each filter.

As explained above, each RSP component (1116) receives (via a dispatcher) telemetry data from a subset of the monitoring modules and calculates snapshots and statistics for all filters. Each RGA component instance is in turn responsible for a subset of the filters. Based on the identifier of the filter, all RSP components send data for that filter to a single RGA component. The RGA component combines the data from all RSP components for the filters that it is responsible for, resulting in a global snapshot and statistics based on information from all monitoring modules. Examples of aggregation operations performed by an RGA component include: counting the total number of viewers that match a particular filter, determining the percentage of viewers matching a given filter that are in buffering state, the joint time distribution experienced by viewers joining a given stream, the current number of viewers in a given city, the rank of the most popular live events, and so on.

In some embodiments an RGA component's functionality is implemented as a continuously running service. It reads records sent by the RSPs asynchronously, thus achieving a high throughput. The RGA component stores the records it receives in in-memory hash tables allowing optimized access for real-time processing. Old information is periodically purged from the hash tables to improve efficiency.

As shown in FIG. 11, gateway 1122 provides a web service API 1124 for accessing data. Through the API, RGAs data is available on a per-filter basis. In addition, it also exposes APIs to edit and install new filters and aggregation functions. In some embodiments gateway 1122 is implemented using an off-the shelf web server (such as Apache) with customized code to handle the various web service API calls. The handlers return data for a given web API call in various data formats including XML, JSON, SOAP, and HTML, as applicable. The handlers serve as middleware for querying and interactively controlling the RSPs and RGAs.

Gateway 1122 also provides access controls and persistently stores the information regarding the API requests, data access and presentation policies, and filter descriptions. The information is maintained in a persistent database, such as MySQL or Oracle database.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A system, comprising:

one or more processors configured to:

receive, at a content distribution coordinator, a manifest request from a client requesting content, wherein the client is configured to access the content distribution coordinator via a virtual resource locator provided to the client by a content management system, wherein the content management system provided the virtual resource locator to the client in response to the client having made a request to the content management

12

system for content, and wherein the virtual resource locator provided to the client by the content management system encodes information including a plurality of content delivery networks at which the requested content is stored;

in response to the content distribution coordinator being accessed via the virtual resource locator provided to the client by the content management system, dynamically generate a manifest file based at least in part on: playback quality experienced by one or more clients determined to be correlated with the client, and the plurality of content delivery networks encoded in the virtual resource locator; and

send, to the client, the manifest file dynamically generated based at least in part on: the playback quality experienced by the one or more clients determined to be correlated with the client, and the plurality of content delivery networks encoded in the virtual resource locator, wherein the client is configured to obtain the requested content according to the dynamically generated manifest file; and

a memory coupled to the one or more processors and configured to provide the one or more processors with instructions.

2. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of a protocol type associated with the requested content.

3. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an identifier associated with the content management system.

4. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of one or more bitrates at which the content is stored.

5. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes a name of the content being requested by the client.

6. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes a type of the content being requested by the client.

7. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of a type of a device associated with the client.

8. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of an access connection associated with a device associated with the client.

9. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of a digital rights management technology used for the requested content.

10. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of an operating system of a device associated with the client.

11. The system recited in claim 10, wherein the information encoded in the virtual resource locator includes an indication of a version of the operating system.

12. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of a location of the content distribution coordinator.

13. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes a security token that was previously generated by the content distribution coordinator and passed to the content management system.

14. The system recited in claim 13, wherein the one or more processors are further configured to validate the security token.

13

15. The system recited in claim 1, wherein the information encoded in the virtual resource locator includes an indication of a firmware version of a device associated with the client.

16. A method, comprising:

receiving, at a content distribution coordinator, a manifest request from a client requesting content, wherein the client is configured to access the content distribution coordinator via a virtual resource locator provided to the client by a content management system, wherein the content management system provided the virtual resource locator to the client in response to the client having made a request to the content management system for content, and wherein the virtual resource locator provided to the client by the content management system encodes information including a plurality of content delivery networks at which the requested content is stored;

in response to the content distribution coordinator being accessed via the virtual resource locator provided to the client by the content management system, dynamically generating, using one or more processors, a manifest file based at least in part on: playback quality experienced by one or more clients determined to be correlated with the client, and the plurality of content delivery networks encoded in the virtual resource locator; and

sending, to the client, the manifest file dynamically generated based at least in part on: the playback quality experienced by the one or more clients determined to be correlated with the client, and the plurality of content delivery networks encoded in the virtual resource locator, wherein the client is configured to obtain the requested content according to the dynamically generated manifest file.

14

17. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

receiving, at a content distribution coordinator, a manifest request from a client requesting content, wherein the client is configured to access the content distribution coordinator via a virtual resource locator provided to the client by a content management system, wherein the content management system provided the virtual resource locator to the client in response to the client having made a request to the content management system for content, and wherein the virtual resource locator provided to the client by the content management system encodes information including a plurality of content delivery networks at which the requested content is stored;

in response to the content distribution coordinator being accessed via the virtual resource locator provided to the client by the content management system, dynamically generating a manifest file based at least in part on: playback quality experienced by one or more clients determined to be correlated with the client, and the plurality of content delivery networks encoded in the virtual resource locator; and

sending, to the client, the manifest file dynamically generated based at least in part on: the playback quality experienced by the one or more clients determined to be correlated with the client, and the plurality of content delivery networks encoded in the virtual resource locator, wherein the client is configured to obtain the requested content according to the dynamically generated manifest file.

* * * * *