

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google LLC,

Petitioner

v.

Sandpiper CDN, LLC,

Patent Owner

Case: IPR2025-00860

U.S. Patent No. 10,924,573

Issue Date: February 16, 2021

Title: Handling Long-Tail Content in a Content Delivery Network (CDN)

**DECLARATION OF TODD C. MOWRY, PH.D.
IN SUPPORT OF PETITION FOR INTER PARTES REVIEW OF
U.S. PATENT NO. 10,924,573**

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Qualifications and Background	2
III.	Materials Considered.....	6
IV.	Legal Standards	7
	A. Claim Construction.....	7
	B. Priority.....	8
	C. Anticipation Under 35 U.S.C. § 102	9
	D. Obviousness Under 35 U.S.C. § 103.....	9
	E. Secondary Considerations of Non-Obviousness.....	11
V.	The '573 patent.....	13
	A. Overview of the '573 patent.....	13
	B. Summary of the Prosecution History of the '573 patent.....	19
	C. Priority Date of the '573 patent.....	21
	D. Claim Construction of Terms in the '573 patent.....	26
VI.	Person of Ordinary Skill in the Art.....	26
VII.	Opinions on Unpatentability.....	28
	A. Ground 1 Anticipates Claims 1-20.....	28
	1. Overview of Newton-471 (EX-1002).....	28
	2. Claim 1	30
	3. Claim 2: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”.....	43

4.	Claim 3	44
5.	Claim 4: “The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.”	46
6.	Claim 5: “The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.”	47
7.	Claim 6: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”	47
8.	Claim 7: “The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.”	48
9.	Claim 8: “The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.”	49
10.	Claim 9: “The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.”	49
11.	Claim 10	50
12.	Claim 11	52
13.	Claim 12: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”	54
14.	Claim 13	55
15.	Claim 14: “The content delivery network as recited in claim 11 wherein the first server of the second plurality	

	of servers is a metro mid-tier server of the content delivery network.”	56
16.	Claim 15: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a peer server of the first tier of servers of the content delivery network.”	56
17.	Claim 16: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”	57
18.	Claim 17: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the first server of the second plurality of servers of the content delivery network.”	57
19.	Claim 18: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the content server of the first tier of servers of the content delivery network.”	58
20.	Claim 19.....	58
21.	Claim 20: “The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.”	60
B.	Ground 2 Renders Obvious Claims 1-20	62
1.	Overview of Dilley (EX-1003)	62
2.	Overview of Pai (EX-1004)	65
3.	Overview of Wang (EX-1005).....	67
4.	Analogous Art.....	68
5.	Combining Dilley and Pai.....	69
6.	Further Combining Dilley and Pai with Wang	77

7.	Claim 1	80
8.	Claim 2: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”	105
9.	Claim 3	109
10.	Claim 4: “The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.”	117
11.	Claim 5: “The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.”	120
12.	Claim 6: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”	123
13.	Claim 7: “The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.”	125
14.	Claim 8: “The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.”	127
15.	Claim 9: “The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.”	127
16.	Claim 10	129
17.	Claim 11	131
18.	Claim 12: “The content delivery network as recited in claim 11 wherein the popularity designation associated	

	with the requested resource indicates that the requested resource is not popular.”	136
19.	Claim 13	136
20.	Claim 14: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a metro mid-tier server of the content delivery network.”	138
21.	Claim 15: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a peer server of the first tier of servers of the content delivery network.”	138
22.	Claim 16: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”	138
23.	Claim 17: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the first server of the second plurality of servers of the content delivery network.”	138
24.	Claim 18: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the content server of the first tier of servers of the content delivery network.”	139
25.	Claim 19	139
26.	Claim 20: “The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.”	142
VIII.	Secondary Considerations	143
IX.	Conclusion	143

I. INTRODUCTION

1. I, Todd Mowry, submit this declaration to state my opinions on the matters described below.

2. I have been retained by Google LLC (“Petitioner”) as an independent expert in this proceeding before the United States Patent and Trademark Office. I am being compensated at the rate of \$800 per hour for time that I spend on this case. No part of my compensation depends on the outcome and I have no other interest in this proceeding.

3. I understand that this proceeding involves U.S. Patent No. 10,924,573 (“the ’573 patent,” EX-1001), and I have been asked to provide my opinions as to the patentability of the claims of the ’573 patent.

4. I have been asked to consider the validity of certain claims of the ’573 patent based on certain prior art references. I have also been asked to consider the state of the art and prior art available as of April 4, 2008, which is the earliest claimed priority date on the face of the ’573 patent, and as of February 25, 2019, which I understand is the earliest priority date the claims of the ’573 patent are entitled to rely on.

5. Based on the combination of prior art references discussed in this declaration, it is my opinion that claims 1-20 of the ’573 patent are unpatentable for the reasons provided below.

II. QUALIFICATIONS AND BACKGROUND

6. Below is a summary of my education, work experience, and other qualifications. My qualifications are stated more fully in my curriculum vitae, which I understand is provided in EX-1014.

7. I am a Professor in the Department of Computer Science at Carnegie Mellon University. I also have a courtesy appointment in the Department of Electrical and Computer Engineering. I have served on the faculty of Carnegie Mellon University for 28 years starting in 1997 through the present (2025).

8. I also served on the faculty of the University of Toronto for four years between 1993 and 1997, in the Department of Electrical and Computer Engineering and a courtesy appointment in the Department of Computer Science. Prior to that appointment, I served as a Graduate Research Assistant in the Department of Electrical Engineering at Stanford University for four years between 1989 and 1993.

9. As a faculty member, I have taught and continue to teach courses and directed research in computer systems and software, operating systems, distributed and network systems, parallel processing, and compiler optimization.

10. I received a B.S. degree in Electrical Engineering with Highest Distinction from the University of Virginia in May 1988. I received an M.S. in Electrical Engineering from Stanford University in June 1989, and a Ph.D. in Electrical Engineering from Stanford University in March 1994.

11. I have worked in the computer industry in various capacities. I was a part-time Computer Architect and then Computer Architecture Consultant at Silicon Graphics, Inc., in Mountain View, California (formerly MIPS Computer Systems in Sunnyvale, California) from 1989 to 1993 and 1993 to 1996, respectively. I was a Visiting Scientist at IBM in Toronto from 1996 to 2004. During that same time period (1996 to 2004), I was also a Member of the Technical Advisory Board of SandCraft, Inc. in Santa Clara, California. I was the Director of the Intel Research Pittsburgh Lab at Intel Corporation in Pittsburgh, Pennsylvania from 2004 to 2007.

12. I have authored 21 journal articles and 71 conference papers. I am also an inventor on five patents.

13. I have published a number of papers in the top research conferences in the fields of data storage, caching, operating systems and distributed systems (including a paper that won the Best Paper Award at the USENIX 2nd Symposium on Operating Systems Design and Implementation (OSDI '96)). *See, e.g.,* Charles Garrod, Amit Manjhi, Anastasia Ailamaki, Bruce Maggs, Todd Mowry, Christopher Olston, and Anthony Tomasic, *Scalable Query Result Caching for Web Applications*, in Proceedings of the 34th International Conference on Very Large Databases (VLDB) (2008); Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, Jiexi Lin, Lin Ma, Prashanth Menon, Todd Mowry, Matthew Perron, Ian Quah, Siddharth Santurkar, Anthony Tomasic, Skye Toor, Dana Van Aken, Ziqi Wang,

Yingjun Wu, Ran Xian, and Tieying Zhang, *Self-Driving Database Management Systems*, in Proceedings of the 2017 Conference on Innovative Data Systems Research (CIDR) (2017); Amit Manjhi, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Anthony Tomasic, *Holistic Query Transformations for Dynamic Web Applications*, in 2009 IEEE 25th International Conference on Data Engineering (ICDE) (2009); and Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, Anthony Tomasic, and Haifeng Yu, *Invalidation Clues for Database Scalability Services*, in Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering (ICDE) (2007).

14. I have also been an active member of the Parallel Data Lab at Carnegie Mellon University since 1997, which is often described as “academia’s premiere data systems research center.”

15. I am the recipient of several honors and awards: the Arthur Samuel Thesis Award (awarded by the Stanford Computer Science department to the top two Ph.D. theses in a given year), several IBM Faculty Development Awards (1996, 1997, 1998, 2000, 2001, 2002, and 2003), several Best Paper Awards (the Second Symposium on Operating Systems Design and Implementation in 1996; the 20th International Conference on Data Engineering (ICDE) in 2004), the Alfred P. Sloan Research Fellowship (awarded to researchers in recognition of distinguished

performance and a unique potential to make substantial contributions to their field), the Most Thought-Provoking Idea Award in 2004 (awarded by the Architectural Support for Programming Languages and Operating Systems (ASPLOS)), the TR100 Award in 1999 (awarded by MIT's Technology Review magazine to the top 100 most promising young innovators in science and technology), and I became an ACM Fellow in 2016.

16. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM). I was the Editor-in-Chief of ACM Transactions on Computer Systems (the premier journal for computer systems research) from 2013 to 2018. I was an Associate Editor for the journal prior to that, since 2001. I was the Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) in 2010. I was the Co-Program Chair of the International Conference on Parallel Architectures and Compilation Techniques (PACT) in 2001. I have been on the programming committee in various years for ASPLOS, the International Symposium on Computer Architecture (ISCA), the International Symposium on Microarchitectures, and the Workshop on Architectural and System Support for Improving Format.

17. Overall, I have over 30 years of experience in the field of computer science and, specifically, computer architecture, operating systems, and document storage and retrieval.

18. In light of the foregoing, I consider myself to be an expert in the field of computer science, and I believe that I am qualified to provide an opinion as to what a person of ordinary skill in the art would have understood, known, or concluded regarding the subject matter of the '573 patent at the time of its alleged invention.

III. MATERIALS CONSIDERED

19. In forming my opinions, I have reviewed the following documents:

Exhibit	Description
EX-1001	U.S. Patent No. 10,924,573 (“the ’573 patent”)
EX-1002	U.S. Patent Publication No. 2016/0094471 to Newtown et al. (“Newton-471”)
EX-1003	U.S. Patent No. 7,133,905 to Dilley (“Dilley”)
EX-1004	U.S. Patent Publication No. 2006/0271972 to Pai (“Pai”)
EX-1005	U.S. Patent Publication No. 2005/0198250 (“Wang”)
EX-1007	Prosecution History of the ’573 patent
EX-1008	U.S. Patent No. 10,218,806
EX-1009	U.S. Patent No. 9,762,692
EX-1010	U.S. Patent No. 8,930,538
EX-1011	U.S. Provisional Patent Application No. 61/042,412 (“the ’412 provisional”)
EX-1012	U.S. Patent No. 8,667,175 to Thomas et al.
EX-1013	U.S. Patent Publication No. 2006/0015574 to Seed et al.
EX-1019	Duane Wessels, Web Caching, O’REILLY (Torkington et al. 1st ed. 2001)

EX-1020	Lv, Z., Yang, et al., Multimedia Delivery Grid: A Novel Multimedia Delivery Scheme, In: Li, M., Sun, XH., Deng, Qn., Ni, J. (eds) GRID AND COOPERATIVE COMPUTING. GCC 2003. LECTURE NOTES IN COMPUTER SCIENCE, vol 3032. Springer, Berlin, Heidelberg (2004), https://doi.org/10.1007/978-3-540-24679-4_46
---------	---

IV. LEGAL STANDARDS

20. In considering the subject matter of the '573 patent and its claims in light of the prior art and forming my opinions, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

21. I understand that the claims define the invention. I also understand that an unpatentability analysis is a two-step process. First, the claims of the patent are construed to determine their meaning and scope. Second, after the claims are construed, the content of the prior art is compared to the construed claims.

22. I understand that an invention, as defined by the claims of the patent, must not be anticipated or rendered obvious by the prior art.

A. Claim Construction

23. I understand that the United States Patent and Trademark Office interprets claim terms in an *inter partes* review proceeding under the same claim construction standard that is used in a United States federal court. I understand that under this standard, the meaning of claim terms is considered from the viewpoint of a person of ordinary skill in the art ("POSITA") at the time of the alleged invention.

24. I have been informed that claim terms are generally given their ordinary and customary meaning as understood by one of ordinary skill in the art in light of the specification and the prosecution history pertaining to the patent. I understand, however, that claim terms are generally not limited by the embodiments described in the specification.

25. I understand that in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

B. Priority

26. I understand that for a patent's claim to receive the benefit of an earlier-filed application, every application in the priority chain between that earlier-filed application and the application in question must support every limitation of the claim. This means the parent application, and the intervening applications, must describe every limitation of every claim and that it is not enough for the patent application to simply render obvious the claims of the patent claims in question.

27. I further understand that to adequately describe a claim limitation, the patent's specification must describe the claimed invention in sufficient detail that a

person of ordinary skill in the art could reasonably conclude that the inventor had possession of the claimed invention at the time of filing.

C. Anticipation Under 35 U.S.C. § 102

28. I understand that a patent claim is anticipated and therefore unpatentable if each element of that claim is present in a single prior art reference.

D. Obviousness Under 35 U.S.C. § 103

29. I understand that a patent claim is invalid as obvious if the claimed invention would have been obvious to a person of ordinary skill in the art at the time the claimed invention was made. I understand that in an obviousness determination, the person of ordinary skill in the art is presumed to have knowledge of all material prior art. I understand that whether a claim is obvious is based upon the determination of several factual issues.

30. In considering obviousness, I understand that one must determine the scope and content of the prior art. I understand that, in order to be considered prior art to a patent, a reference must be reasonably related to the claimed invention of that patent. A reference is reasonably related if it is in the same field as the claimed invention or is from another field to which a person of ordinary skill in the art would look to solve a known problem.

31. I understand that one must determine what differences, if any, existed between the claimed invention and the prior art.

32. I understand that a patent claim composed of several limitations is not proved obvious merely by demonstrating that each of its limitations was independently known in the prior art. In evaluating whether such a claim would have been obvious, one may consider whether a reason has been identified that would have prompted a person of ordinary skill in the art to combine the elements or concepts from the prior art in the same way as in the claimed invention.

33. I understand that one may consider whether (1) the change is merely the predictable result of using prior art elements according to their known functions or the result of true inventiveness; (2) there is some teaching or suggestion in the prior art to make the modification or combination of limitations claimed in the patent; (3) the innovation applies a known technique that had been used to improve a similar device or method in a similar way; (4) the claimed invention would have been obvious to try, meaning that the claimed innovation is one of a relatively small number of possible approaches to the problem that could have been used by those of ordinary skill in the art with a reasonable expectation of success; (5) the invention merely substitutes one known element for another known element in order to obtain predictable results; (6) the invention merely applies a known technique to a known device, method, or product to yield predictable results; or (7) known work in the field may have prompted variations of use of the same inventions in the same or

different fields due to market forces or design incentives that would have been predictable to a person of ordinary skill in the art.

34. I understand that one must be careful not to determine obviousness using the benefit of hindsight; many true inventions might seem obvious after the fact. I understand that the determination should be based on the position of a person of ordinary skill in the art at the time the claimed invention was made and should not reflect what is known today or what is learned from the teaching of the patent.

E. Secondary Considerations of Non-Obviousness

35. I understand that certain secondary considerations may be considered in evaluating obviousness in order to prevent hindsight bias. These secondary considerations include the commercial success of products that practice the patent, long-felt need for the patented technology, failure by others to solve the problem addressed by the patent, initial skepticism by others in the industry, industry recognition and praise of the patented products, and efforts by others to copy the patented technology.

36. I have been informed that these secondary considerations of non-obviousness must be balanced against the strength of the *prima facie* case of obviousness. Where the invention represents no more than the predictable use of prior art elements according to their established functions, secondary considerations are inadequate to establish non-obviousness.

37. I have also been informed that the patentee must establish a nexus between any secondary considerations and the claimed invention. Where a secondary consideration results from something other than what is both claimed and novel in the patent, there is no nexus to the merits of the claimed invention.

38. I have been informed that, for example, the nexus requirement as it applies to evidence of commercial success requires that the patentee must show that the sales of a product that allegedly practices the claimed invention must be a direct result of the unique characteristics of the claimed invention, as opposed to other economic and commercial factors that are unrelated to the patented invention. I understand this to mean that if the product's commercial success is due to an element that existed in the prior art or that is not claimed by the patent, then no nexus exists.

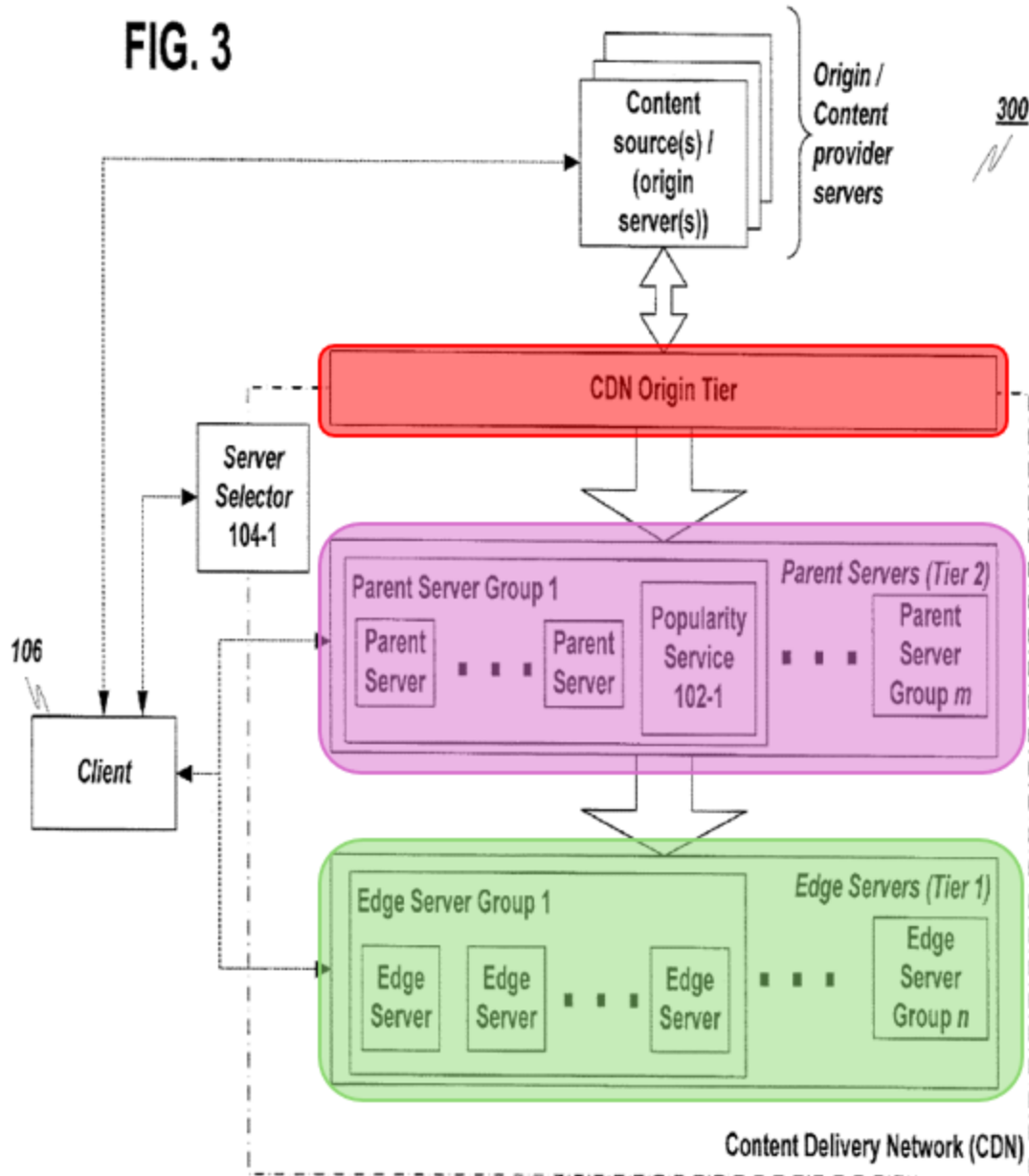
39. I have been further informed that, for example, the nexus requirement as it applies to the long-felt need element must likewise be supported by evidence. Where the differences between the prior art and the claimed invention are minimal, it cannot be said that any long-felt need was met by the patented invention or that the patent solved any technological problems that were unaddressed by others.

40. I have been informed that evidence of industry praise must also show a nexus with the claimed invention, and that self-serving statements by the patentee regarding the merits of the claimed invention are generally not given weight.

V. THE '573 PATENT

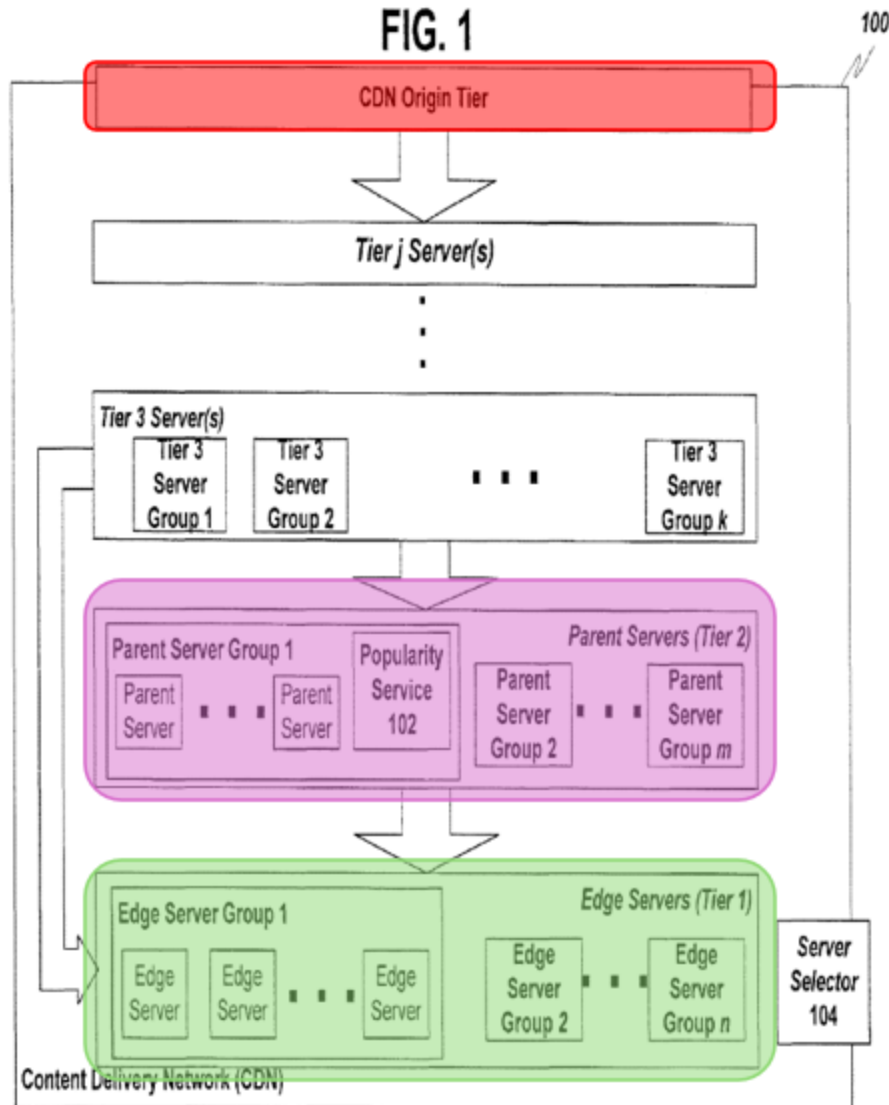
A. Overview of the '573 patent

41. The '573 patent is generally directed to a content delivery network (“CDN”) that includes a popularity service. EX-1001, Abstract. A CDN is a network of servers distributed geographically to deliver web content to users. EX-1001, 2:28-36; EX-1012, 1:13-19. The '573 patent describes a CDN organized into hierarchical tiers, where tier 1 servers represent edge servers, sitting closest to the clients (end-users), the origin tier represents those servers sitting furthest from the clients, and a parent tier could sit in between. EX-1001, 2:60-65, 3:53-57, 4:19-31; EX-1013, ¶[0010] (“edge servers are typically closer (by various measures of distance) to end users” and thus “facilitate more cost-effective and efficient delivery of [certain] objects”).



EX-1001, Fig. 3 (annotated showing **edge servers**, **parent servers**, and **origin tier**).

42. The '573 patent also describes a server selector system that directs user requests to the appropriate edge server within the CDN based on factors like proximity to the user and server load. EX-1001, 4:47-55.



Id., Fig. 1 (annotated).

43. The edge servers receive incoming requests for content (like a web page or a video) based on the server selector system and attempt to handle these requests directly when possible. EX-1001, 5:59-6:17. Indeed, each of these servers maintains

a cache¹ of frequently accessed content in order to improve delivery speed. EX-1001, 6:11-17. Thus, when a client requests a resource from the system, the first server will receive the request and check whether the content is readily available from the first server or a nearby peer server (e.g., another edge server in the same group). *Id.* If the resource is available, the content is served to the client from the first server. *Id.*

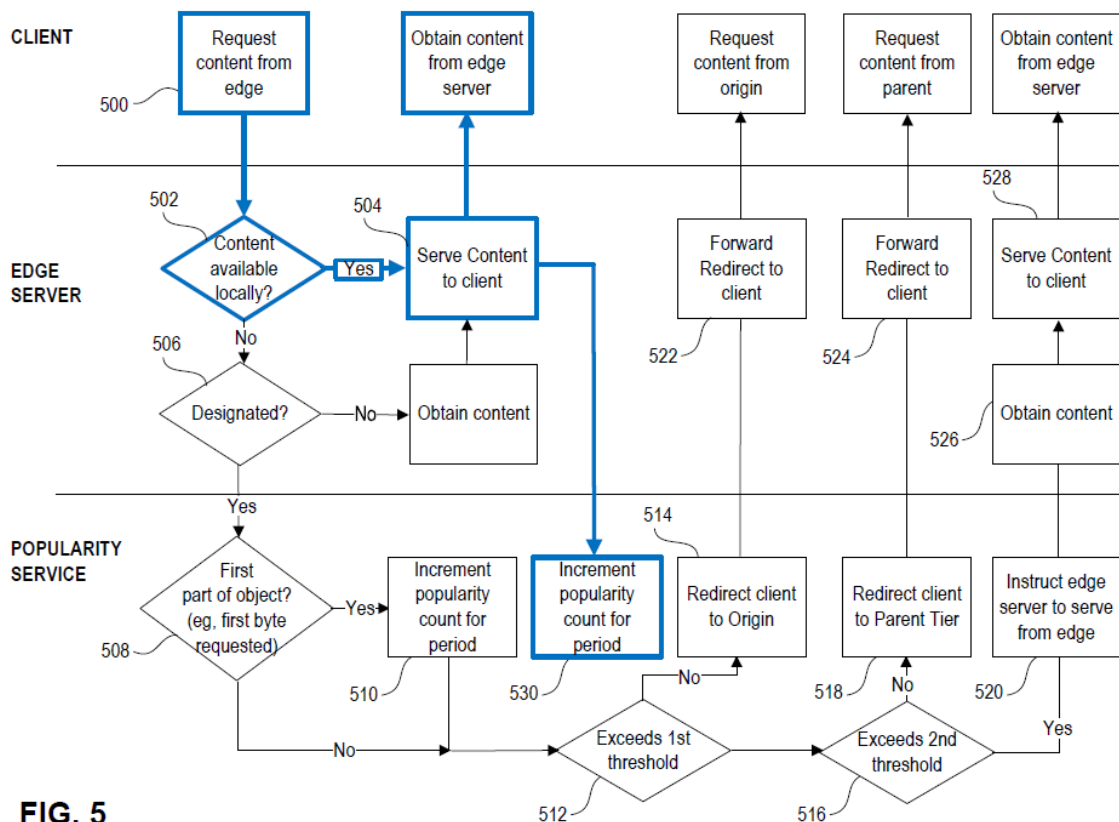


FIG. 5

¹ A cache stores or manages a copy of content stored on origin servers so that future requests for that content may be served faster. EX-1019, 15 (stating “caching refers to the storage of recently retrieved computer information for future reference”).

Id., Fig. 5 (sharpened and annotated).

44. When the resource is unavailable from the first server or peer server—for instance for less frequently accessed content—the '573 patent will evaluate the content's popularity using a “popularity service.” EX-1001, 6:27-34. The '573 patent provides three exemplary responses for the popularity service in determining the popularity of a resource. *Id.*

45. First, for unpopular resources (e.g., below a minimum popularity threshold), the request is redirected to an origin server or the CDN's origin tier. EX-1001, 6:65-2. Second, when a resource is some-what popular (e.g., between the minimum and a higher popularity threshold), the client's request is redirected to a parent server. EX-1001, 7:3-17. Third, when a resource is popular (e.g., exceeding higher popularity threshold), the first server will retrieve the content and serve the request itself. EX-1001, 6:53-7:30.

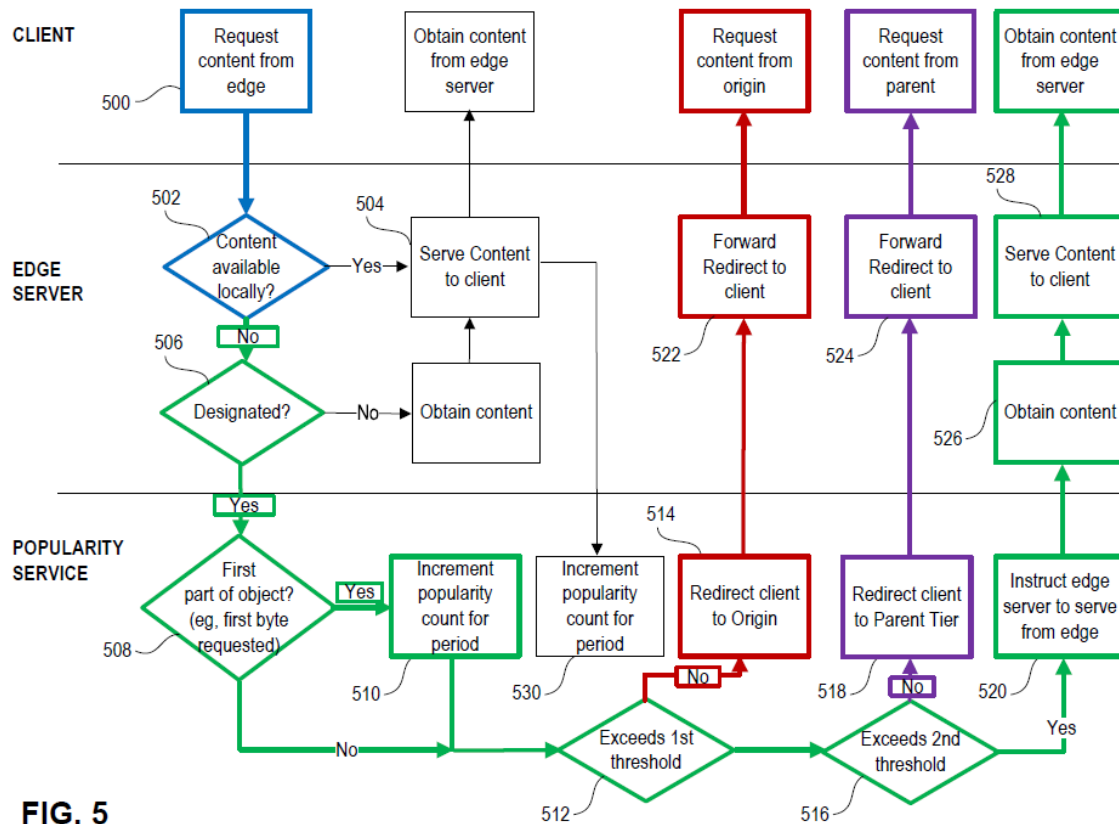


FIG. 5

Id., Fig. 5 (annotating in green the process for popular content being served from edge, annotating in purple the process for mid-level popular content being served from a parent server, and annotating in red the least popular content being served from the origin server or the CDN's origin tier).

46. By tracking the popularity of content, the system can adjust where resources are stored within the CDN and minimize response time for popular content while more efficiently managing unpopular (i.e., more infrequently requested) content. EX-1001, 7:35-54. Thus, the '573 patent sought to use its tiered server

system and popularity server to speed up content delivery and preserve memory space at the edge of the internet for popular content.

B. Summary of the Prosecution History of the '573 patent

47. I have reviewed the prosecution history of the application that led to the '573 patent. *See* EX-1007.

48. Application No. 16/284,791 (the "'791 application"), which issued as the '573 patent, was filed on February 25, 2019. Ex-1001, Cover. The '791 application claims priority as a continuation of U.S. Application No. 15/645,584 (now U.S. Patent No. 10,218,806), which claims priority as a continuation U.S. Application No. 12/880,324 (now U.S. Patent No. 9,762,692), which claims priority as a continuation-in-part to U.S. Application No. 12/408,681 (now U.S. Patent No. 8,930,538), which ultimately claims priority to Provisional Application No. 61/042,412 (the "'412 provisional"), filed on April 4, 2008. As further explained below, neither the '412 provisional nor the intervening three applications to which the '573 patent claims priority support the claims of the '573 patent. *See infra* §V.C. Despite this, the application was examined under pre-AIA provisions. EX-1007, 17.

49. In a first office action, Examiner issued a nonstatutory double patenting rejection, *inter alia*, finding various claims (which were substantially similar to those ultimately issuing as the Challenged Claims) unpatentable over the claims of U.S. Patent No. 10,348,848 to Newton et al. ("Newton-848"). Examiner further rejected

various claims as obvious over U.S. Patent No. 7,552,223 to Raju et al. (“Raju”) in view of U.S. Patent Application No. 2004/0221019 to Swildens et al. (“Swildens”) and rejected others as obvious over Raju and Swildens in further view of U.S. Patent No. 7,024,466 to Outten et al. (“Outten”). EX-1007, 76.

50. Examiner did not, however, issue an obviousness rejection for then-pending claim 2, which recited “receiving *an instruction to not cache* the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network.” EX-1007, 276 (emphasis added).

51. In response to this rejection, Applicant filed a terminal disclaimer, overcoming the double patenting rejections based on Newton-848. EX-1007, 46. Regarding the § 103 rejections, Applicant amended the independent claims to incorporate claim 2’s “instruction to not cache” limitation. EX-1007, 46-47. Applicant accordingly argued that the claims were patentable over the cited references in view of the amendment. EX-1007, 47.

52. Examiner then issued a Notice of Allowance stating that the “prior art of record fails to teach or suggest ‘receiving an *instruction to not cache* the resource at the first server.’” EX-1007, 17 (emphasis added). Specifically, Examiner differentiated Raju (“[t]he closest prior art”), explaining that while “Raju teaches a policy for managing caching content at the edge,” a determination of “whether [to]

cache a content item is different from ‘sending an instruction to not cache the content item’ as recited in the claims.” EX-1007, 17. Examiner similarly differentiated Swildens (“[t]he other closest prior art”), finding that although Swildens “teaches a popularity service” and caches content “based on the popularity value,” accessing “the popularity value is different from ‘sending an instruction to not cache the content item’ as recited in the claims.” EX-1007, 17. Thus, the Challenged Claims were allowed in light of this “instruction to not cache” limitation.

53. But as discussed below, Newton-471, as well as the other prior art not applied by Examiner, discloses this limitation.

C. Priority Date of the ’573 patent

54. In my opinion, the priority date for the claims of the ’573 patent is February 25, 2019, the filing date of the application that led to the ’573 patent. That is the date that the claims reciting new material, such as the “instruction to not cache” limitation was added.

55. Indeed, it is my opinion that no claim of the ’573 patent is entitled to claim priority to the ’412 provisional or any intervening application because each prior application fails to provide written description support for at least the “instruction to not cache” limitations found in independent claims 1, 11, and 19.

56. The specification of the ’573 patent has no support for an “instruction to not cache”; the limitation appears only in the claims. *See generally* EX-1001. And

nothing in the '573 patent's chain resolves this deficiency. The '573 patent is a continuation of now U.S. Patent 10,218,806 (the "'806 patent"), which is a continuation of now U.S. Patent 9,762,692, which is a continuation-in-part of now U.S. Patent 8,930,538 (the "'538 patent"), which claims priority to the '412 provisional.

57. The '573 patent substantively shares a disclosure with the '806 patent. The '806 patent substantively shares a disclosure with the '692 patent. *Compare* EX-1008 *with* EX-1009.

58. The '692 patent, a continuation-in-part, added new material not present in the '538 patent—namely Figure 7, and its accompanying description. Figure 7 provides a flowchart for “processing steps associated with a multi-tiered content-delivery network.” EX-1009, 2:3-4, 11:55-56, Fig. 7. These processing steps include “obtain[ing] a request from a client” and determining if “the resource is available.” EX-1009, 11:55-61. If “the resource is not available” and “is popular” it will “obtain the resource from an origin server and/or a parent server” and serve it to the client. EX-1009, 62-12:2. If the resource is not available and “is determined not to be popular,” “the client is directed to a second server in a second tier of servers.” EX-1009, 12:3-9. Figure 7 further provides an explanation for partitioning a content provider's library across multiple servers in the second tier of servers and directing the client to the appropriate server, storing the portion of the requested resource. EX-

1009, 12:17-28. Besides Figure 7 and the related disclosure, the '692 patent substantively shares a disclosure with the '538 patent. Notably, none of the new matter in the '692 patent discusses or provides support for a specific instruction to not cache portions of a resource at the edge.

59. Continuing through the chain of priority, the '538 patent substantively shares a disclosure with the '412 provisional application that is the earliest claimed priority application. But neither the '412 provisional nor, by extension, the intervening three applications to which the '573 patent claims priority support the “instruction not to cache” limitation found in each challenged claim. Thus, no priority documents within the chain include a specific instruction not to cache a resource or a portion of a resource at the edge.

60. At most, the '412 provisional and intervening applications discuss a determination of whether to cache content at the first server. EX-1011, ¶[0012]. For instance, the '412 provisional provides examples when it will “obtain and cache[] the resource.” EX-1011, ¶[0044] (explaining that when “the edge server 108 receives a ‘follow me’ redirect, it obtains and caches the resource (at 526) and serves it to the client”). But the specification is silent as to sending or receiving instructions to not cache a resource. And as the Examiner noted, a determination of “whether [to] cache a content item is different from ‘sending an instruction to not cache the content item’ as recited in the claims.” EX-1007, 17.

61. The specification contemplates a “rule base” set by the subscriber to prevent content from being cached, but this does not disclose a specific “instruction not to cache the portion of the resource at the edge server,” as claimed. EX-1011, ¶[0062]. For example, the ’412 provision provides that “a subscriber may not want to pay for certain content to be served from the edge, regardless of its popularity, and may set a rule accordingly” (i.e., “setting the thresholds for that particular content to prevent it from ever being cached at the edge”). EX-1011, ¶[0062].

62. The specification’s disclosure of a universal rule set by the subscriber amounts only to a determination of whether to cache content. A POSITA would understand that there is some content that a subscriber would likely never want to cache or pay to store and, therefore, would set up a universal rule to designate this type of content as uncacheable regardless of its popularity. This suggests an overriding rule that may be implemented into the system but does not disclose the necessary support for directing or receiving specific instructions regarding the CDN’s caching behavior.

63. For instance, a POSITA would have understood that content would only be cacheable if it can be used to answer a future request. EX-1019, 43 (§2.2 Is It Cacheable?) (explaining that an administrator can define a rule set to prevent dynamic content from being cached since it would not be used for future requests, thereby saving resources and increasing performance by not caching them). This

would be beneficial because, due to storage and cost limitations, not everything can be cached. *See* EX-1013, ¶[0011] (“It is typically not feasible to store all objects on the edge servers. The main difficulty is due to the fact that many such objects are very large (typically on the order of 10 MB (10,000,000 bytes)—in the neighborhood of 500 MB for movies.”).

64. The same is true for the ’412 provisional’s description that “HTTP POST requests” “should not be cached.” EX-1011, ¶[0052]. A POSITA would understand that “POST requests” fall into the category of resources a subscriber would likely never want to cache or pay to store. By way of example, these can be things that are generated dynamically and would become stale if they were to be cached, such as personalized recommendations in a shopping experience or the time of day. Therefore, these resources would logically “not be cached.” But this again amounts to a determination of whether to cache content and not to specific “instruction[s] not to cache the portion of the resource at the edge server,” as claimed. EX-1007, 17.

65. Thus, none of the priority documents provide written description support for the negative “instruction to not cache” limitations, and each claim in the ’573 patent is therefore entitled to a priority date no earlier than the ’573 patent’s filing on February 25, 2019.

66. Accordingly, I understand that under the actual priority date of February 25, 2019, all of the references are prior art under 35 U.S.C. § 102(a)(1). But even if the '573 patent were entitled to its claimed priority date of April 4, 2008, I understand that the Ground 2 references remain prior art under pre-AIA 35 U.S.C. § 102(b).

D. Claim Construction of Terms in the '573 patent

67. As discussed above, I have been informed that, for purposes of *inter partes* review, the standard for construction of terms in a patent's claims is the same as that applied in federal district court litigation.

68. I understand that a district court has yet to construe the claim terms of this patent. At this time, I do not see the need to construe any terms of the challenged claims for the purposes of this proceeding. Accordingly, in preparing my analysis I have applied the ordinary and customary meaning that each claim term would have to a person of ordinary skill in the art in light of the '573 patent's specification and prosecution history.

69. If any claim terms are differently construed, I reserve the right to reassess my opinions under those constructions.

VI. PERSON OF ORDINARY SKILL IN THE ART

70. I have been informed that patentability must be analyzed from the perspective of a "person of ordinary skill in the art" in the same field as the

patent-in-suit at the time of the invention. I also understand that several factors are considered in assessing the level of ordinary skill in the art, including (1) the types of problems encountered in the art; (2) the prior art solutions to those problems; (3) the rapidity with which innovations are made; (4) the sophistication of the technology; and (5) the education level of active workers in the field.

71. I refer to a person of ordinary skill in the art here as a “POSITA.” A POSITA at the time of the alleged invention (including either April 4, 2008, or February 25, 2019), would have had at least a bachelor’s degree in computer science, electrical engineering, or a related field, and at least two years of work or research experience in the field of content delivery management or networks. Work experience can substitute for formal education and additional formal education can substitute for work experience. I at least meet this definition of a POSITA, now and as of the priority date of the ’573 patent.

72. I am also familiar with the knowledge of a POSITA as of the priority date of the ’573 patent (whether that is found to be April 4, 2008 or Feb. 25, 2019). I am able to opine on how a POSITA would have understood the disclosure and claims of the ’573 patent, the disclosures of the prior art, the various motivations to combine the prior art, and what combinations would have been obvious to a POSITA.

VII. OPINIONS ON UNPATENTABILITY

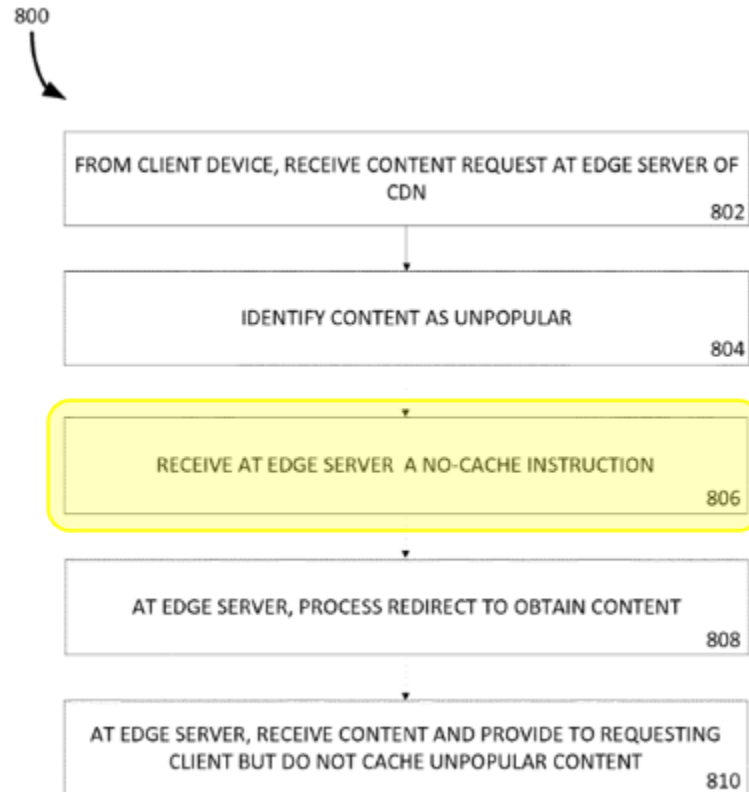
73. In the analysis that follows, I identify the following prior art and combinations of prior art that, in my opinion, render obvious the claims of the '573 patent:

Ground	Prior Art	Basis	Claims
1	Newton-471 (EX-1002)	§ 102	1-20
2	Dilley (EX-1003), Pai (EX-1004), and Wang (EX-1005)	§ 103	1-20

A. Ground 1 Anticipates Claims 1-20

1. Overview of Newton-471 (EX-1002)

74. Newton-471 is a published patent application that includes largely the same written description and figures of the '573 patent while including additional written support and figures for the description of the “instruction to not cache” limitations. *Compare* EX-1002 *with* EX-1001. Specifically, the following Figure 8 from Newton-471 and its corresponding description regarding an “instruction not to cache” is absent from the '573 patent and its priority applications:



EX-1002, Fig. 8 (annotated).

75. Figure 8 of Newton-471 identifies that the edge server receives a specific no-cache instruction (step 806) and thus will receive the content and provide that content to the requesting client without caching the unpopular content at the edge (step 810). *Id.*

76. Newton-471 further explains that:

In operation 806, the edge server 704 contacts the mid-tier 708 or other server hosting the content to receive the content in operation 806. In addition to returning the requested content, the mid-tier server 708 may also return a cache or no-cache instruction. In such an instance, the mid-tier 708 may have the content or may need to retrieve the content from the CDN origin 710, the customer origin 714, a different tier or otherwise. Regardless, the mid-tier 708, in conjunction with the

popularity service 716, determines or otherwise identifies the popularity of the content. For unpopular content, the mid-tier 708 returns a no-cache instruction to the content server 706. . . . In addition, the edge server 706 does not cache the content in response to the no-cache instruction.

EX-1002, ¶¶[0098]-[0099]. The '573 patent and its priority applications fail to include this (or any other description) regarding a no-cache instruction.

77. And because the '573 patent cannot claim priority before February 25, 2019, Newton-471, which was published on March 31, 2016, is § 102(a)(1) prior art, as explained above.

2. Claim 1

a. 1(preamble): “A method of content delivery in a content delivery network comprising:”

78. Newton-471 discloses this limitation.

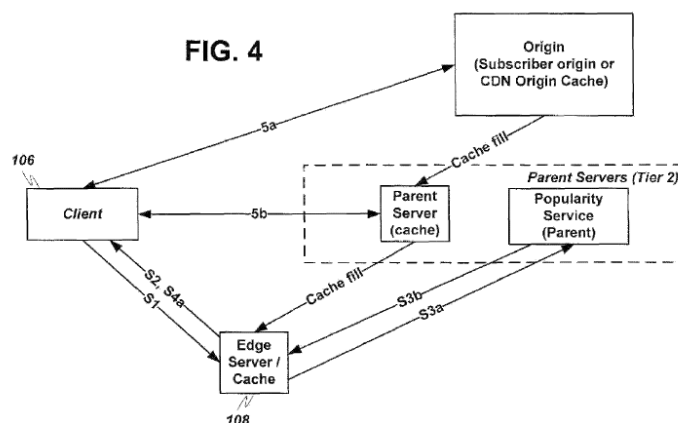
79. Newton-471 includes largely the same figures and written description as the '573 patent and thus similarly discloses the delivery and management of content in a content delivery network. EX-1002, ¶[0010] (“One implementation of the present disclosure may take the form of *a method for content delivery in a content delivery network.*”) (emphasis added).

80. Further, like the '573 patent, Newton-471 claimed “[a] method of content delivery in a content delivery network comprising...” EX-1002, claim 1; *see also id.*, claims 11 and 19.

- b. 1(a) “receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;”

81. Newton-471 discloses this limitation and explains its “method includes the operations of receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network.” EX-1002, ¶¶[0010]-[0012], [0048], [0096], [0100] (providing edge servers receive client requests); *see also id.*, claims 1, 11, and 19.

82. Newton-471 also provides that a “client 106 requests content from an edge server 108 (at 500 in FIG. 5). The request from the client 106 arrives at edge server 108 (51 in FIG. 4). The edge server 108 checks to see if the object is present (locally or on a peer) and fresh (at 502).” EX-1002, ¶[0048]. Thus, Newton-471 discloses this limitation.



EX-1002, Fig. 4.

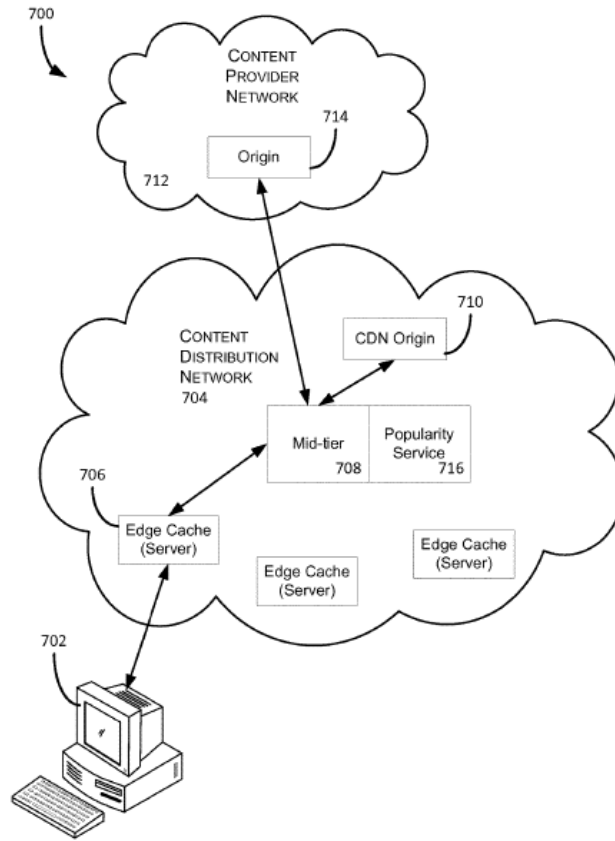
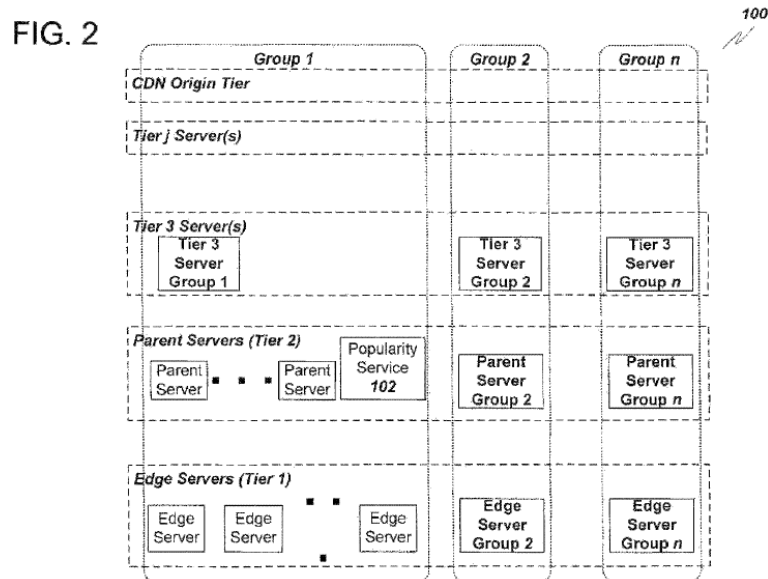
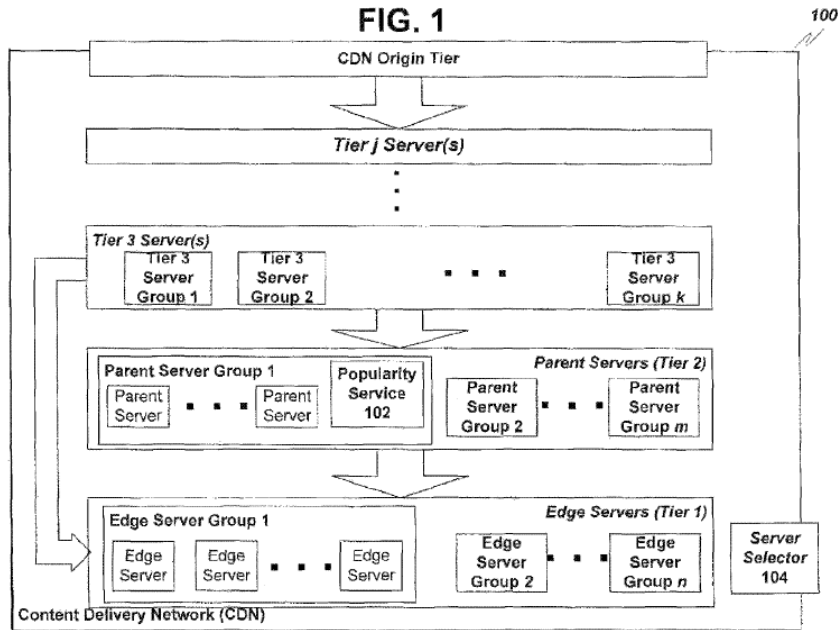


FIG. 7

EX-1002, Fig. 7 (depicting client 702 sending a request for content to the edge server).

83. Newton-471 also explains that its hierarchical CDN “may have one or more tiers of servers,” including, “Tier 1 [through] Tier j.” EX-1002, ¶¶[0029]. “Each tier of servers may comprise a number of servers organized into server groups,” such that the first tier of servers can include a plurality of edge servers, for example. EX-1002, ¶¶[0029]-[0030].



EX-1002, Fig 1 (top), Fig. 2 (bottom).

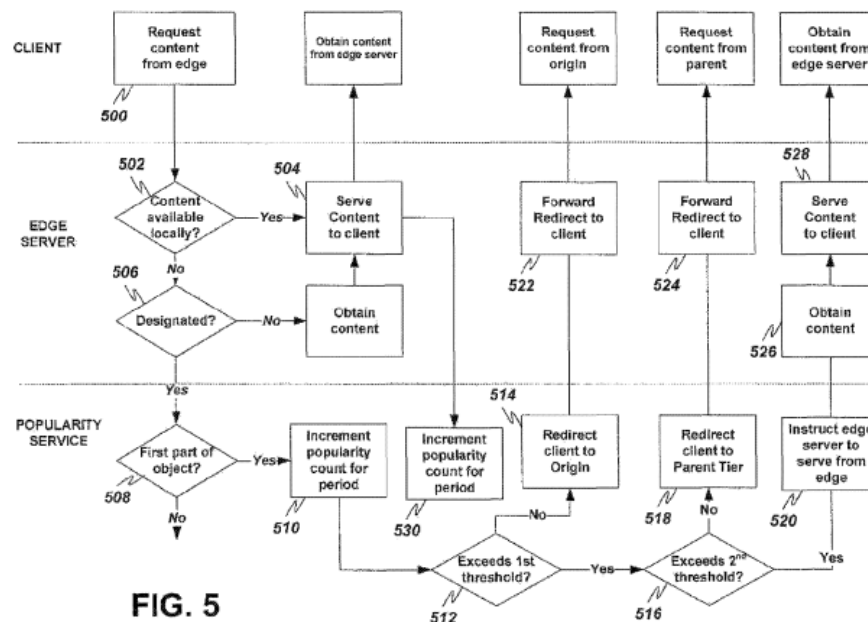
84. Further, like the '573 patent, Newton-471 claimed “receiving, at a first server of a first tier of servers of the content delivery network, a request from a

requesting device for a resource available from the content delivery network.” EX-1002, claim 1 *see also id.*, claims 11 and 19.

c. 1(b) “accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource;”

85. Newton-471 discloses this limitation and explains that its “method includes the operations” of “accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource.” EX-1002, ¶¶[0010]-[0012].

86. Newton-471’s Figure 5 and supporting description is identical to the ’573 patent and details the claimed popularity service:



EX-1002, Fig. 5.

87. Newton-471 continues that “[i]f the object is not available on the edge server 108 or on a peer, the edge server 108 ascertains whether this object is served based on its popularity” and “the request is sent to the popularity service 102.” EX-1002, ¶[0050]; *see also id.*, ¶¶[0096]-[0098], [0100]. “[T]he popularity service 102 determines (as described below) if the object is currently popular,” (i.e., determines a popularity designation associated with the requested resource). EX-1002, ¶¶[0053]-[0057]. Indeed, “the popularity service 102 returns one of three possible responses:” (1) “the object has not reached a first/minimal level of popularity (at 512),” (2) “the object’s popularity has exceeded the first/minimal level of popularity but has not yet exceeded a second, mid-tier threshold (at 516),” or (3) “the object’s popularity has exceed the mid-tier threshold (i.e., the object is popular).” EX-1002, ¶¶[0053]-[0057]. Thus, Newton-471 discloses this limitation.

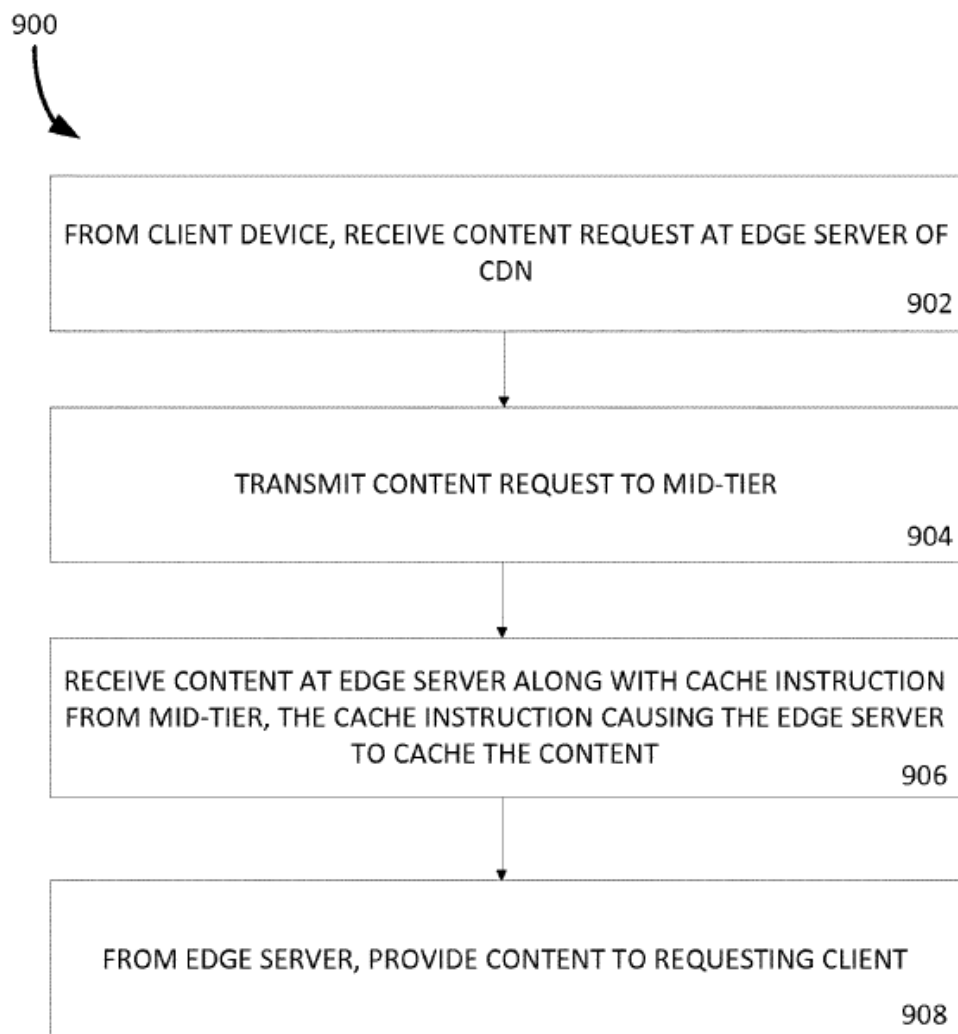
88. Further, like the ’573 patent, Newton-471 claimed “accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource.” EX-1002, claim 1; *see also id.*, claims 11 and 19.

d. 1(c) “requesting the resource from a second server of the content delivery network;”

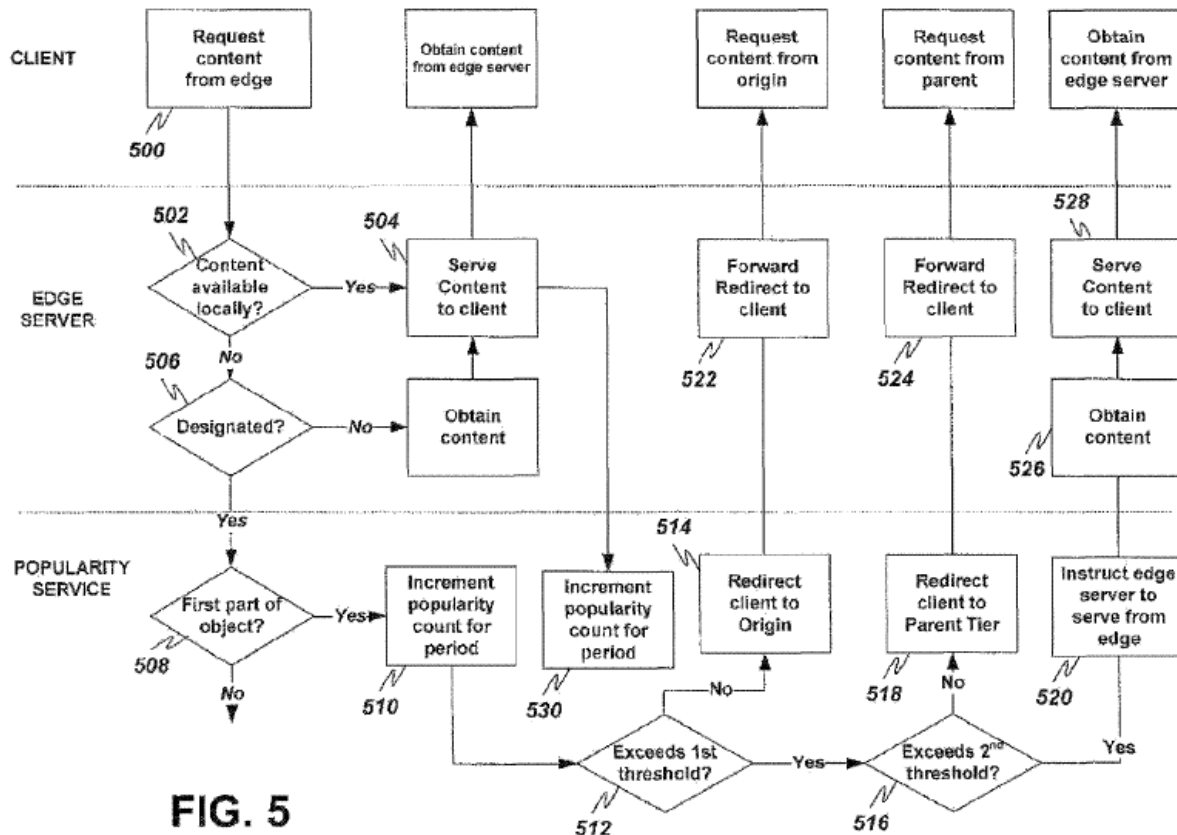
89. Newton-471 discloses this limitation and explains that its “method includes the operations” of “requesting the resource from a second server of the

content delivery network.” EX-1002, ¶¶[0010]-[0012]; *see also id.*, claims 1, 11, and 19.

90. It does this in at least two ways. First, Newton-471 explains that after receiving a request “[t]he edge server 108 checks to see if the object is present (locally or on a peer).” EX-1002, ¶[0048]. In this instance, the peer server would be a second server of the content delivery network. Second, Newton-471 explains that if the content is not available at the first server, the request may be “redirect[ed] to a parent or origin server,” as illustrated in Figure 9 and Figure 5 (below). EX-1002, ¶[0061]; *see also* EX-1002, ¶¶[0055]-[0057], [0096].



EX-1002, Fig. 9.



EX-1002, Fig. 5 (flowchart depicting content requested from origin or parent server based on popularity).

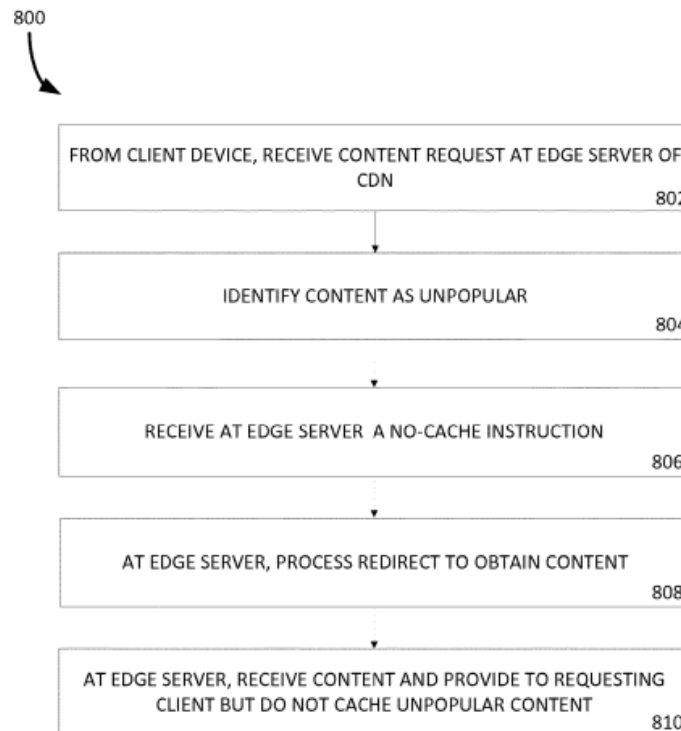
91. Further, like the '573 patent, Newton-471 claimed “requesting the resource from a second server of the content delivery network.” EX-1002, claim 1; *see also id.*, claims 11 and 19.

- e. **1(d) “processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network;”**

92. I do not have an opinion on whether the '573 patent contains written description support for this limitation. But if it does, Newton-471 discloses this

limitation and explains that its “method includes the operation[]” of “processing, at the first server of the first tier of servers, a redirect command from the second server of the content delivery network to obtain the resource from a content server of the content delivery network.”² EX-1002, ¶[0010]; *see also id.*, claims 1, 11.

93. Newton-471 discloses that “[t]he edge server 706 may process the redirect [instruction],” received from the second server, as shown in operation 808 of Figure 8. EX-1002, ¶¶[0097]-[0099]; *see also id.*, ¶¶[0095]-[0096], [0101].



EX-1002, Fig. 8.

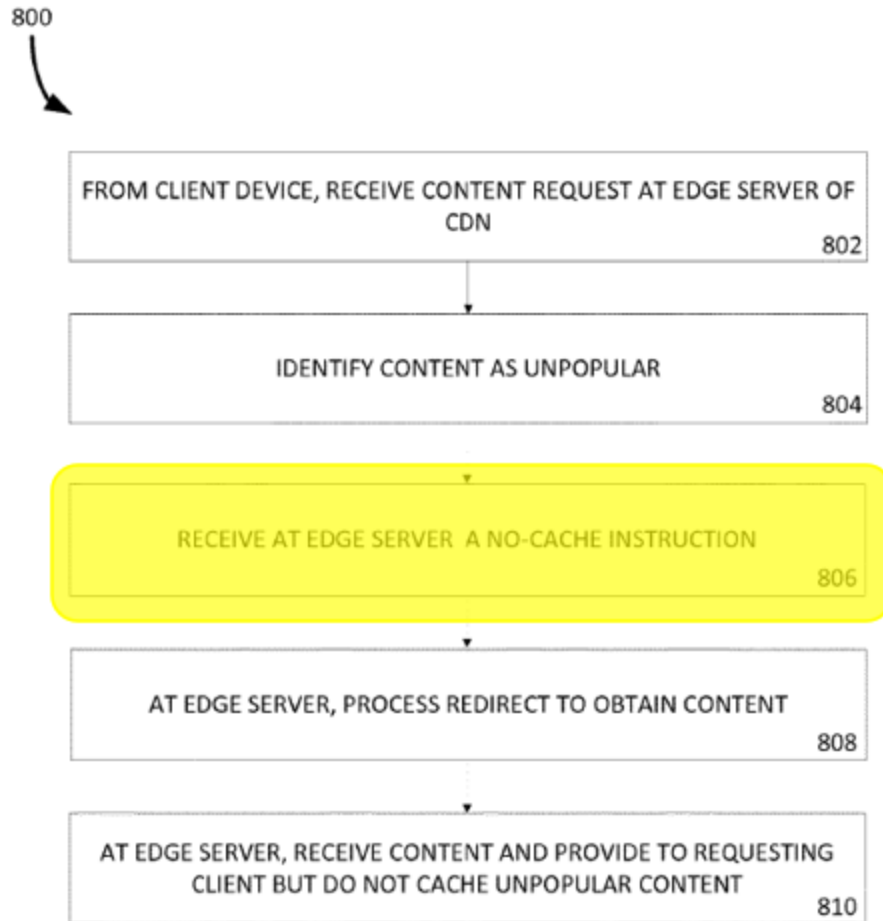
² While Newton-471’s Summary discloses “a redirect command,” it later claims “a redirect instruction.” *See* EX-1002, claim 1 (“processing . . . a redirect instruction from the second server...”).

94. Newton-471 further explains that “[i]f the object’s popularity has exceeded the first/minimal level of popularity but has not yet exceeded a second, mid-tier threshold (at 516): the popularity service sends the edge server an instruction (e.g., HTTP 302) to redirect the client’s request to a parent server.” EX-1002, ¶¶[0055]-[0057]; *see also id.*, ¶[0061].

95. Further, like the ’573 patent, Newton-471 claimed “processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network.” EX-1002, claim 1; *see also id.*, claim 11.

f. 1(e) “receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network; and”

96. While it is my opinion that the ’573 patent does not provide support for this limitation, *supra* §V.C, Newton-471 does. Indeed, Newton-471 discloses this limitation because it provides a figure and description regarding the specific “to not cache” instruction. For instance, Newton-471’s Figure 8 specifically depicts a method of serving content from the CDN “where the content is not cached.” EX-1002, ¶¶[0095]-[0096]; *see also id.*, ¶[0012]. This method includes step 806, which provides that the edge server will receive a “no-cache instruction” and will accordingly serve, but not cache, the requested content. EX-1002, ¶¶[0098]-[0099].



EX-1002, Fig. 8 (annotated).

97. Newton-471 discloses that when an edge server does not have the requested content, it “contacts the mid-tier 708 or other server hosting the content to receive [it].” EX-1002, ¶[0098]. “In addition to returning the requested content, the mid-tier server 708 may also return a *cache or no-cache instruction*. In such an instance, the mid-tier 708 may have the content or may need to retrieve the content from the CDN origin” or elsewhere. EX-1002, ¶[0098]. Specifically, Newton-471 states “[f]or unpopular content, the mid-tier 708 returns a no-cache instruction” and

“the edge server 706 does not cache the content in response to the no cache instruction.” EX-1002, ¶[0099].

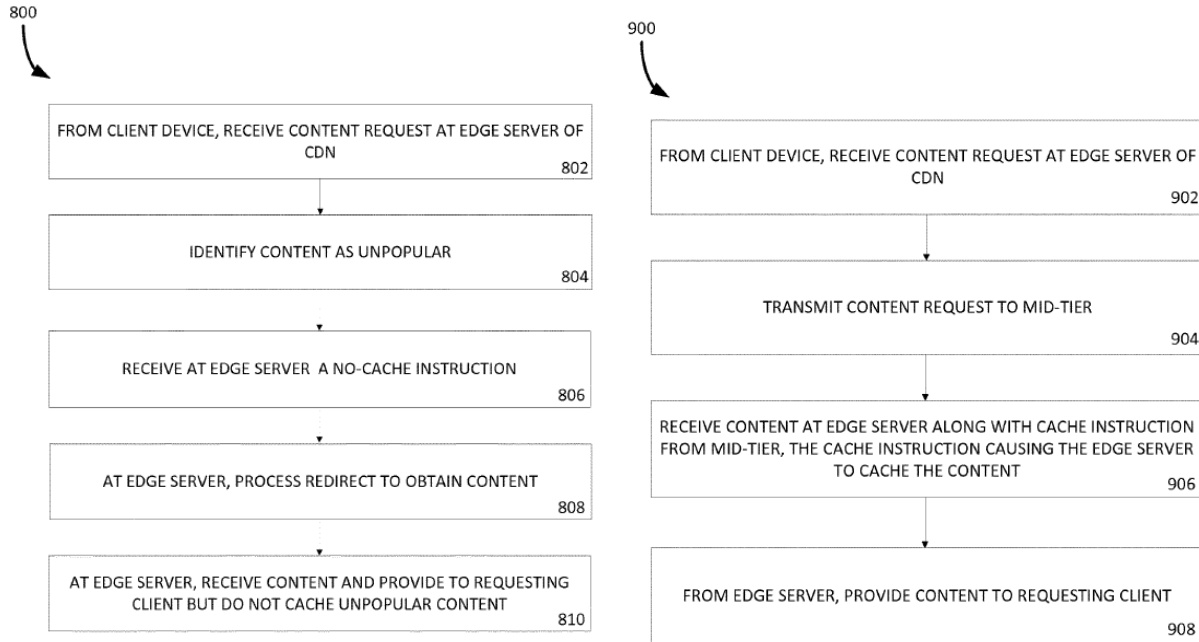
98. Thus, while the ’573 patent and its priority documents are silent as to this limitation, Newton-471 discloses it.

99. Further, like the ’573 patent, Newton-471 claimed “receiving an instruction to not cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.” EX-1002, claim 2; *see also id.*, claim 19.

g. 1(f) “providing the obtained resource to the requesting device.”

100. Newton-471 discloses this limitation and explains that its “method includes the operations” of “providing the obtained resource to the requesting device.” EX-1002, ¶¶[0010]-[0012]; *see e.g., id.*, claim 1. Indeed, Newton-471’s system is intended to provide the requested content to the client (i.e., requesting device). EX-1002, ¶¶[0047]-[0048], [0052], [0058].

101. As further examples, each of Figures 8 and 9 conclude with providing content to the requesting client:



EX-1002, Figs. 8 (left), 9 (right); *see also id.*, ¶¶[0099]-[0101].

102. Further, like the '573 patent, Newton-471 claimed “providing the obtained resource to the requesting device.” EX-1002, claim 1.

3. Claim 2: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”

103. Newton-471 discloses this limitation.

104. Newton-471 explains that the popularity designation associated with the requested resources indicates the popularity of the resource, including whether that resource is or is not popular. EX-1002, ¶¶[0096]-[0097] (explaining that the popularity service determines if the requested resource is unpopular); *see also id.*, ¶¶[0054]-[0057] (explaining that the popularity service determines if (1) an “object

has not reached a first/minimal level of popularity,” deeming it unpopular; (2) an “object’s popularity has exceeded the first/minimal level of popularity but has not yet exceeded a second mid-tier threshold,” deeming it somewhat popular; or (3) an “object’s popularity has exceeded the mid-tier threshold (i.e., the object is popular)”), Fig. 8 (step 804).

105. Like the ’573 patent, Newton-471 claimed a method “wherein the popularity designation associated with the resource indicates that the resource is not popular.” EX-1002, claim 2; *see also id.*, claim 12.

4. Claim 3

- a. **3(a) “The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular,”**

106. –Newton-471 discloses this limitation.

107. As explained above, Newton-471 explains that the popularity designation associated with the requested resources indicates the popularity of the resource, including whether that resource is or is not popular. EX-1002, ¶¶[0054]-[0057] (explaining that the popularity service determines if (1) an “object has not reached a first/minimal level of popularity,” deeming it unpopular; (2) an “object’s popularity has exceeded the first/minimal level of popularity but has not yet exceeded a second mid-tier threshold,” deeming it somewhat popular; or (3) an “object’s popularity has exceeded the mid-tier threshold (i.e., the object is

popular)"); *see also* EX-1002, ¶[0100] (explaining that the server “may determine that the content is popular based on the popularity service”).

108. Further, like the '573 patent, Newton-471 claimed “wherein the popularity designation associated with the requested resource indicates that the requested resource is popular.” EX-1002, claim 3; *see also id.*, claim 13.

- b. 3(b) “the method further comprising: receiving an instruction to cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.”**

109. Newton-471 discloses this limitation.

110. Newton-471 explains that “[i]f the content is or has become popular as determined by the popularity service 716, then the content is returned to the edge server 706 with a cache instruction.” EX-1002, ¶[0101]; *see also id.*, ¶[0092], claims 3, 13. Newton-471 further explains that the instruction to cache is received when the resource is obtained from the content server because Newton-471 explains that “[r]egardless of from which component of the CDN 704 the content is received, the edge server 706 receives the content in operation 906 and, in response to the cache instruction, caches the popular content at the content server.” EX-1002, ¶[0101].

111. However, based on this claim’s dependency, this claim limitation appears to contradict the claim limitations from which it depends. For instance, claim 1 recites “receiving an instruction *to not cache* the portion of the resource at the first

server of the first tier of servers when the portion of the resource is obtained from the content server.” *See* EX-1001, claim 1 (limitation 1(e)) (emphasis added). Claim 3, which depends from claim 1, now recites “receiving an instruction *to cache* the resource at the first server of the first tier of servers when the resource is obtained from the content server.” *See* EX-1001, claim 3 (emphasis added). Thus, I understand that claim 3 creates an internal inconsistency by simultaneously requiring an instruction to *not cache* and an instruction *to cache* the same resource at the same server when the resource is obtained from the content server. However, for purposes of this proceeding, I assume this limitation can be reconciled with claim 1.

112. Further, like the ’573 patent, Newton-471 claimed “receiving an instruction to cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.” EX-1002, claim 3; *see also id.*, claim 13.

5. Claim 4: “The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.”

113. Newton-471 discloses this limitation. *See e.g.*, EX-1002, ¶¶[0093]-[0094] (explaining the mid-tier server can be a “metro mid-tier server 708” and providing the benefits of using such a “metro mid-tier server”).

114. Like the '573 patent, Newton-471 claimed a method “wherein the second server is a metro mid-tier server of the content delivery network.” *See* EX-1002, claim 4; *see also id.*, claim 14.

6. Claim 5: “The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.”

115. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(c), when the second server is a peer server of the first tier of servers. *Supra* §VII.A.2.d.

116. Like the '573 patent, Newton-471 claimed a method “wherein the second server is a peer server of the first tier of servers of the content delivery network.” EX-1002, claim 5; *see also id.*, claim 15.

7. Claim 6: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”

117. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(b), claim 2, and limitation 3(a). *Supra* §§VII.A.2.c, VII.A.3, VII.A.4.a; *see also* EX-1002, Fig. 5. Indeed, Newton-471 provides three potential outcomes in response to a popularity determination: the object is (1) unpopular, falling below a minimal popularity threshold; (2) mid-tier popular,

exceeding the minimal popularity threshold but not a second, mid-tier threshold; or (3) popular, exceeding the mid-tier threshold. EX-1002, ¶¶[0054]-[0057], [0063].

118. Like the '573 patent, Newton-471 claimed a method “wherein the popularity designation associate with the resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.” EX-1002, claim 6; *see also id.*, claim 16.

8. Claim 7: “The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.”

119. Newton-471 discloses this limitation.

120. Newton-471 explains that the popularity service can be “associated with one or more of the server groups in one or more tiers.” EX-1002, ¶[0039]. It provides an “exemplary embodiment” wherein “some of the parent server groups have popularity services 102 associated therewith.” EX-1002, ¶[0039]. Thus, Newton-471 provides that the popularity service is associated with the second server, when the second server is a parent server, but it further contemplates that “the popularity service 102 may be integrated into one or more of the servers in the group, [and i]n general, the popularity service 102 may be associated or integrated into any server or server group of the CDN.” EX-1002, ¶[0039]; *see also id.*, ¶¶[0064], [0077], Figs. 1-4.

121. Like the '573 patent, Newton-471 claimed a method “wherein the popularity service is associated with the second server of the content delivery network.” *See* EX-1002, claim 7; *see also id.*, claim 17.

9. Claim 8: “The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.”

122. Newton-471 discloses this limitation for at least the same reasons as provided above for claim 7. *Supra* §VII.A.8. Newton-471 suggests that the popularity service may be associated into any server. EX-1002, ¶[0039].

123. Like the '573 patent, Newton-471 claimed a method “wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.” EX-1002, claim 8; *see also id.*, claim 18.

10. Claim 9: “The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.”

124. Newton-471 discloses this limitation for at least the same reasons as above for limitations 1(a) and 1(c) wherein the second server is a parent server. *Supra* §§VII.A.2.b, d. As explained for limitation 1(a), Newton-471 describes a hierarchical CDN with multiple tiers of servers including, “Tier 1, Tier 2, Tier 3, [through] Tier j.” EX-1002, ¶[0029]-[0030].

125. Like the '573 patent, Newton-471 claimed a method “wherein the content delivery network further comprises a second tier of servers distinct from the

first tier of servers, and wherein the second server is in the second tier of servers.”

EX-1002, claim 9; *see also id.*, claim 11.

11. Claim 10

- a. 10(a) “The method as recited in claim 3 further comprising: receiving, at the first server of the first tier of servers of the content delivery network, an additional request for the resource available from the content delivery network;”**

126. Newton-471 discloses this limitation.

127. Newton-471 explains that the edge server may “cache popular content to serve the content in response to a future request” (i.e., an additional request for the resource). EX-1002, ¶[0095]; *see also* EX-1002, ¶[0059] (explaining that “once content has been cached at the edge server, the edge server” will notify “the popularity service every time it gets another request for that content” and will determine “that it has the requested content”). Thus, Newton-471 discloses receiving an additional request for the resource available from the CDN.

128. Like the ’573 patent, Newton-471 claimed a method further comprising “receiving, at the first server of the first tier of servers of the content delivery network, an additional request for another portion of the resource available from the content delivery network.” EX-1002, claim 10.

- b. 10(b) “determining that the resource is cached at the first server of the first tier of servers; and”**

129. Newton-471 discloses this limitation.

130. As explained above for limitation 10(a), Newton-471 explains that the edge server will determine, in response to another request for content it has already cached, that it has the requested content available in its cache. EX-1002, ¶¶[0048], [0059], [0095]-[0096].

131. Like the '573 patent, Newton-471 claimed a method further comprising “determining that the another portion of the resource is cached at the first server of the first tier of servers.” EX-1002, claim 10.

c. 10(c) “providing the cached resource from the first server of the first tier of servers in response to the additional request for the resource.”

132. Newton-471 discloses this limitation.

133. As explained above for limitation 10(a), Newton-471 explains that the edge server will determine, in response to another request for content it has already cached, that it has the requested content available and will serve that content. EX-1002, ¶[0059] (“then, in addition to serving the content (at 504) . . . it also instructs the popularity service to increment the objects [sic] popularity count”); *see also id.*, ¶¶[0095]-[0096]; *Supra* §VII.A.11.a.

134. Like the '573 patent, Newton-471 claimed a method further comprising “providing the cached another portion of the resource from the first server of the first tier of servers in response to the additional request for the another portion of the resource.” EX-1002, claim 10.

12. Claim 11

- a. 11(preamble): “A content delivery network comprising:”**

135. Newton-471 discloses this limitation for the same reasons provided above for limitation 1(preamble). *Supra* §VII.A.2.a.; *see also* EX-1002, claim 11.

- b. 11(a) “a first tier of servers comprising a first plurality of servers, an edge server of the first tier of servers receiving a request from a requesting device in communication with the first server for a resource available from the content delivery network;”**

136. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(a). *Supra* §VII.A.2.b; *see also* EX-1002, claim 11.

- c. 11(b(i)) “a second tier of servers comprising a second plurality of servers,”**

137. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(c) and claim 9. *Supra* §§VII.A.2.d, VII.A.10.

138. Like the '573 patent, Newton-471 claimed “a second plurality of servers.” EX-1002, claim 11.

- d. **11(b(ii)) “a first server of the second plurality of servers receiving a request from the edge server for the content and, in response, transmitting a redirect instruction intended for the requesting device to the edge server to obtain the resource from a content server and transmitting an instruction to not cache the portion of the resource at the edge server when the portion of the resource is obtained from the content server; and”**

139. Newton-471 discloses this limitation for at least the same reasons provided above for limitations 1(c)-1(e). *Supra* §§VII.A.2.d-f.

140. Like the '573 patent, Newton-471 claimed “first server of the second plurality of servers receiving a request from the edge server for the portion of the resource and, in response, transmitting a redirect instruction to the edge server to obtain the portion of the resource from a content server” and “transmit[ting] an instruction to not cache the [portion of the] resource at the [edge server] when the [portion of the] resource is obtained from the content server.” EX-1002, claims 11 and 12.

- e. **11(c) “a popularity service tracking a popularity designation associated with the requested resource;”**

141. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(b). *Supra* §VII.A.2.c.

142. Like the '573 patent, Newton-471 claimed “a popularity service tracking a popularity designation associated with the resource.” EX-1002, claim 11.

- f. **11(d) “wherein the edge server processes the redirect instruction to obtain the content from the content server and provides the obtained content to the requesting device.”**

143. Newton-471 discloses this limitation for at least the same reasons provided above for limitations 1(d) and 1(f). *Supra* §§VII.A.2.e-g.

144. Like the '573 patent, Newton-471 claimed “wherein the edge server processes the redirect instruction to obtain the portion of the resource from the content server and provides the obtained portion of the resource to the requesting device.” EX-1002, claims 9, 11.

- 13. Claim 12: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”**

145. Newton-471 discloses this limitation for at least the same reasons provided above for claim 2. *Supra* §VII.A.3.

146. Like the '573 patent, Newton-471 originally claimed “wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.” EX-1002, claim 12.

14. Claim 13

- a. 13(a) “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular and”**

147. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 3(a). *Supra* §VII.A.4.a.

148. Like the '573 patent, Newton-471 originally claimed “wherein the popularity designation associated with the requested resource indicates that the requested resource is popular.” EX-1002, claim 13.

- b. 13(b) “the first server of the second plurality of servers transmits an instruction to cache the resource at the content server of the first tier of servers when the resource is obtained from the content server.”**

149. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 3(b). *Supra* §VII.A.4.b.

150. Similar to claim 3, this claim limitation appears to contradict the claim limitations from which it depends. For instance, claim 11 recites “transmitting an instruction *to not cache* the portion of the resource at the edge server when the portion of the resource is obtained from the content server.” *See* Limitation 11(b(ii)) (emphasis added). Claim 13, which depends from claim 11, now recites “transmits an instruction *to cache* the resource at the content server of the first tier of servers when the resource is obtained from the content server.” *See* Claim 13 (emphasis added). Thus, I understand that claim 13 is simultaneously requiring an instruction

to *not cache* and an instruction *to cache* the same resource at the same server when the resource is obtained from the content server. However, for purposes of this proceeding, I assume this limitation can be reconciled with claim 11.

151. Like the '573 patent, Newton-471 originally claimed “the first server of the second plurality of servers transmits an instruction to cache the resource at the content server of the first tier of servers when the resource is obtained from the content server.” EX-1002, claim 13.

15. Claim 14: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a metro mid-tier server of the content delivery network.”

152. Newton-471 discloses this limitation for at least the same reasons provided above for claim 4. *Supra* §VII.A.5.

153. Like the '573 patent, Newton-471 claimed “wherein the first server of the second plurality of servers is a metro mid-tier server of the content delivery network.” EX-1002, claim 14.

16. Claim 15: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a peer server of the first tier of servers of the content delivery network.”

154. Newton-471 discloses this limitation for at least the same reasons provided above for claim 5. *Supra* §VII.A.6.

155. Like the '573 patent, Newton-471 claimed “wherein the first server of the second plurality of servers is a peer server of the first tier of servers of the content delivery network.” EX-1002, claim 15.

17. Claim 16: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”

156. Newton-471 discloses this limitation for at least the same reasons provided above for claim 6. *Supra* §VII.A.7.

157. Like the '573 patent, Newtown-471 claimed “wherein the popularity designation associated with the requested resources comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.” EX-1002, claim 16.

18. Claim 17: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the first server of the second plurality of servers of the content delivery network.”

158. Newton-471 discloses this limitation for at least the same reasons provided above for claim 7. *Supra* §VII.A.8.

159. Like the '573 patent, Newton-471 claimed “wherein the popularity service is associated with the first server of the second plurality of servers of the content delivery network.” EX-1002, claim 17.

19. Claim 18: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the content server of the first tier of servers of the content delivery network.”

160. Newton-471 discloses this limitation for at least the same reasons provided above for claim 8. *Supra* §VII.A.9.

161. Like the '573 patent, Newton-471 claimed “wherein the popularity service is associated with the content server of the first tier of servers of the content delivery network.” EX-1002, claim 18.

20. Claim 19

a. 19(preamble): “A method of content delivery in a content delivery network comprising:”

162. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(preamble). *Supra* §VII.A.2.a.

163. Like the '573 patent, Newton-471 claimed “[a] method of content delivery in a content delivery network comprising.” EX-1002, claim 19.

b. 19(a) “receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;”

164. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(a). *Supra* §VII.A.2.b.

165. Like the '573 patent, Newton-471 claimed “receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting

device for a resource available from the content delivery network.” EX-1002, claim 19.

- c. **19(b) “accessing a popularity service associated with the first server of the first tier of the content delivery network to determine a popularity designation associated with the requested resource;”**

166. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(b) and claim 8. *Supra* §§VII.A.2.c, VII.A.9.

167. Like the ’573 patent, Newton-471 claimed “accessing a popularity service associated with the first server of the first tier of the content delivery network to determine a popularity designation associated with the requested resource.” EX-1002, claim 19.

- d. **19(c) “receiving an instruction to not cache the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network;”**

168. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(e). *Supra* §VII.A.2.f.

169. Like the ’573 patent, Newton-471 claimed “receiving an instruction to not cache the resource at the first server of the first tier of servers.” EX-1002, claim 19.

- e. **19(d) “requesting the resource from a second server of the content delivery network based at least on the popularity designation associated with the requested resource; and”**

170. Newton-471 discloses this limitation for at least the same reasons provided above for 1 limitations 1(b)-1(c), claim 2, and limitation 3(a). *Supra* §§VII.A.2.c-d, VII.A.3-4.a; *see also* EX-1002, claim 19. For instance, Newton-471 provides that the request to a second server of the CDN is based at least on whether the popularity designation is unpopular, popular, or somewhere in-between. *See also* EX-1002, ¶¶[0054]-[0057], [0063], Fig. 5.

171. Like the '573 patent, Newton-471 claimed “requesting the resource from a second server of the content delivery network based at least on the popularity designation associated with the requested resource.” EX-1002, claim 19.

- f. **19(e) “providing the obtained resource to the requesting device.”**

172. Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(f). *Supra* §VII.A.2.g.

173. Like the '573 patent, Newton-471 claimed “providing the obtained resource to the requesting device.” EX-1002, claim 19.

- 21. **Claim 20: “The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.”**

174. Newton-471 discloses this limitation.

175. Newton-471 explains that the content server may be an edge server (i.e., a peer server of the first server). EX-1002, ¶[0090] (“the content server (e.g., edge server)”); *see also* EX-1002, ¶[0091] (“In one example, the content server may be an edge server”), claim 20. Thus, under this circumstance, Newton-471 discloses this limitation for the same reasons provided above for claims 5 and 15. *Supra* §§VII.A.6, VII.A.16. Alternatively, Newton-471 explains that “[i]t is also possible that the content server may be a device in one of the other tiers discussed above, such as a parent server or origin server.” EX-1002, ¶[0091]. Thus, under the circumstance where the content server is a parent server, Newton-471 discloses this limitation for the same reasons provided above for claim limitation 1(c), claim 4, claim limitation 11(b(ii)), claim 14, and claim limitation 19(d). *Supra* §VII.A.2.d, VII.A.5, VII.A.12.d, VII.A.15, VII.A.20.e. Newton-471 also discloses this limitation under the circumstance where the content server is an origin server because Newton-471 discloses a CDN with only one tier, including only edge servers and origin servers. EX-1002, ¶[0034]. Thus, in such a configuration, Newton-471 discloses that the content server would be the origin server.

176. Like the '573 patent, Newton-471 claimed “wherein the second server of the content delivery network is a content server storing the requested content.” EX-1002, claim 20.

B. Ground 2 Renders Obvious Claims 1-20

1. Overview of Dilley (EX-1003)

177. Dilley discloses an improved content delivery network using a tiered cache hierarchy to efficiently distribute content to end-users. EX-1003, Abstract. Dilley explains that its system reduces the load on and insulates the origin server from requests for popular content, thereby improving the performance of the CDN. EX-1003, 1:8-10, 2:12-31.

178. Dilley's system employs a tiered hierarchy of servers, including edge servers (sitting closest to the clients), parent servers, and origin servers. EX-1003, 2:32-36, 3:67-4:2, 4:63-5:15. The system receives a request and routes it to the nearest edge server. EX-1003, 7:2-6, 8:3-10. If the edge server has the requested content, it will directly serve that content to the user. EX-1003, 8:3-10. Otherwise, the edge server will request the content from another server, e.g., a peer, parent, or origin server. EX-1003, 8:11-48.

179. Content is cached and fetched from the different levels of servers (edge, parent, or origin) based on specific criteria defined by the content's metadata, for example. EX-1003, 6:10-40, 7:6-40. Indeed, Dilley uses metadata tags to control how content is distributed and from where it should be served. *Id.* More specifically, Dilley uses a cache hierarchy, where content is served from either the origin server,

intermediate parent servers, or directly from edge servers depending on the content's characteristics (such as popularity, size, and content type). EX-1003, 6:4-7, 9,57-62.

180. Dilley discloses that in operation, the system will receive a content request from an end user and will route that request to the nearest edge server within the CDN. EX-1003, 7:2-6, 8:3-10. If the edge server has the requested content, it will directly serve that content to the requesting user. EX-1003, 8:3-10. Otherwise, if the edge server cannot fulfill the client's request (either because it does not have the content or the content is stale), the edge server will request the content from another server using the cache hierarchy. EX-1003, 8:11-21. As part of the cache hierarchy, the edge server can determine from where to request the content and can send the request to a parent server or to the origin server accordingly. EX-1003, 8:11-48.

181. The cache hierarchy operates based on metadata associated with the content resource. EX-1003, 8:23-48. For instance, the content's embedded metadata indicates whether the cache hierarchy should be used and whether the content should be served from a nearby edge or parent server or retrieved from the origin server. *Id.* As one example, the metadata could indicate that the cache hierarchy should be used and that the edge server should first request the content from the intermediate parent server region, and the parent server will either return the content from its cache or,

if that parent server does not have the content, will escalate the request to the origin server. *Id.*

182. In Dilley's system, the origin server is contacted only when necessary. This ensures that the origin server is not overwhelmed with frequent or redundant requests. Dilley identifies that this is especially beneficial when dealing with flash-popular content or large files. EX-1003, 1:8-10, 2:32-36, 2:67-3:2, 9:57-65. By serving these popular or large file content types from the edge or parent servers, the system reduces latency and delivers requested content quickly, even if the edge server does not initially have the content cached. EX-1003, 6:4-10. The tiered approach also improves overall content delivery efficiency and allows the CDN to handle large volumes of requests with reduced congestion on the origin servers. EX-1003. 5:65-6:15.

183. Dilley also discloses that its system allows for a subset of objects to be serviced through the cache hierarchy, while it will indicate that others should be fetched from the origin server directly. EX-1003, 4:23-26, 8:23-48. These decisions are controlled through the embedded metadata designations of each content item and can be set at the object level or for all objects belonging to a particular content provider. EX-1003, 4:23-26, 6: 10-15, 8:23-48. Thus, Dilley's hierarchical approach provides for flexibility depending on the needs of the content provider and allows

for the management or organization of content based on metadata attributes specified by the content provider.

2. Overview of Pai (EX-1004)

184. Pai discloses a system and method for popularity-based on-demand media distribution. EX-1004, Abstract. Pai's system dynamically manages and distributes media content based on the popularity of the media content. EX-1004, Abstract, ¶[0013]. For instance, Pai's design facilitates media storage and delivery to users by adjusting the storage locations of content (such as at storage level 1, 2, or 3) in a network environment based on the content's popularity. EX-1004, ¶¶[0013], [0016].

185. Pai explains that popularity is assigned to each content item and is regularly updated based on a calculation of popularity values. EX-1004, ¶[0026]. Using these popularity values, Pai will dynamically redistribute the media content between different storage levels (i.e., level 1, 2, and 3) on the network based on the popularity value. EX-1004, ¶¶[0018], [0039]-[0044]. Pai explains that this approach helps ensure that more frequently accessed content is readily available and able to be delivered quickly, while less popular content can be stored further from users, occupying less valuable storage capacity (i.e., slower, long-term storage options). EX-1004, ¶¶[0017], [0020].

186. Using Pai's system, after a user requests a media item, Pai's system will retrieve the content from the relevant media source based on the content's current location within the system. EX-1004, ¶¶[0013], [0020]. Pai explains that as content is requested, its popularity is tracked and incremented, and this information is then used to dynamically adjust the content's popularity value. EX-1004, ¶¶[0026], [0029]. Once the popularity value exceeds (or falls below) a certain threshold, it will be redistributed to a higher-level (or lower-level) media source. Ex-1004, ¶¶[0038]-[0045].

187. This dynamic redistribution process ensures that the most popular content is stored at the media source that will allow it to be retrieved quickly. EX-1004, ¶[0017]. Content with medium popularity will be redistributed to level 2 sources, such as mid-tier storage options, and items with low popularity are moved to level 3 media sources, where they are stored on slower, long-term storage sources. *Id.*

188. Accordingly, Pai discloses a dynamic and efficient approach to managing and delivering media content based on content popularity. By continuously assessing the popularity of each media item and dynamically redistributing content to the appropriate storage levels, the system ensures fast retrieval times for users requesting popular media content, while balancing cost-

effective storage options for less frequently requested media items. EX-1004, ¶¶[0017], [0020].

3. Overview of Wang (EX-1005)

189. Wang discloses a hierarchical scalable integrated service network with level 1, level 2, and level 3 service managers responsible for the different tiers of the network. EX-1005, Abstract, ¶[0022]. Information is aggregated from lower to higher levels of the network and supports content caching at multiple levels, such that content can be cached at local, data center, or global levels, for instance. EX-1005, ¶¶[0008], [0059].

190. Wang explains that when a request is received the system will check the first level cache to determine if the content is available. EX-1005, ¶[0123]. If the content is not found, the request escalates up the service hierarchy from level 1 to level 2 and then level 3 (if the content is not available at level 2). *Id.*

191. In the hierarchical caching system, Wang explains that its system will identify certain items that should not be cached by the lower-level service managers. EX-1005, ¶[0123]. This is based on cache control attributes, which a POSITA would have understood could be used for things that should not be cached, such as dynamic content (i.e., dynamic or personalized information that may be session-specific, user-specific, or require real-time updates), private user information (to prevent sensitive data from being stored in caches), or for certain content types that might not be suited

for caching (including unpopular content that would otherwise take up valuable cache storage space if it was to be cached and thus should be fetched fresh from the origin server).

4. Analogous Art

192. In my opinion, each of Dilley, Pai, and Wang is analogous art to the '573 patent. The '573 patent states that the “invention relates to content delivery, to content delivery networks (CDNs), and to frameworks and systems using CDNs.” EX-1001, 1:46-48. Like the '573 patent, each of Dilley, Pai, and Wang are in the same field of endeavor as each is directed to networks for storing and delivering content. EX-1003, Abstract (“A tiered distribution service is provided in a content delivery network (CDN),” using “a cache hierarchy”); EX-1004, Abstract (discussing a “[p]opularity-based on-demand media distribution” system); EX-1005, Abstract (“A network system manages hierarchical service and content distribution via a directory enabled network to improve performance of the content delivery network with a hierarchical service network infrastructure design.”).

193. In my opinion, Dilley, Pai, and Wang are also analogous art because they are reasonably pertinent to the problem that the '573 patent purports to solve. In particular, the '573 patent discusses a tiered CDN that uses a popularity service to manage long-tail content and thereby adjust where content is stored within the CDN's tiers to efficiently deliver and manage resources. EX-1001, 10:5-41.

194. Dilley, Pai, and Wang are reasonably pertinent to this same problem.

Dilley discloses providing “a tiered distribution method and system in a content delivery network to ensure that end users can obtain desired content quickly and reliably, while effectively buffering web site infrastructure, thereby ensuring that an origin site is not overwhelmed with requests for popular content.” EX-1003, 2:26-31. Dilley does this by storing its popular (i.e., most frequently requested content) at edge servers, thereby efficiently storing content within the CDN to improve delivery speed and reliability. EX-1003, 2:37-56, 5:4-21. Pai similarly discloses a system and “techniques for distributing on-demand media content within an on-demand media system based on popularity,” such that “more popular media content may be maintained by a media source (e.g., a server or server cluster) located physically closer to the client devices.” EX-1004, ¶ [0013]. This accomplishes the same goal of improving reliability and speed of delivery. Finally, Wang also seeks to improve “server congestion and Internet traffic jams” in a content delivery network by “provid[ing] a network system having multiple levels” (i.e., “a hierarchical service network infrastructure design”). EX-1005, ¶ [0003].

5. Combining Dilley and Pai

195. A POSITA would have found it obvious to implement or modify Dilley’s content delivery network using a popularity service, as disclosed in Pai.

196. First, a POSITA would have found it obvious to combine Dilley (EX-1003) with Pai (EX-1004) to provide an improved and dynamic tiered content delivery system that includes a popularity service. Combining Dilley's tiered cache architecture with Pai's introduction of real-time method for adjusting media content placement based on popularity would have led to a more efficient, scalable, and cost effective CDN. EX-1003, 1:30-32 (explaining that an effective CDN serves frequently accessed content from an optimal location), 2:26-31 (explaining that a tiered distribution method allows end users to obtain desired content quickly and reliably without overwhelming the origin server with requests for popular content). A POSITA would have understood that this combination would allow content to be placed dynamically in the appropriate cache, based on its popularity, and would improve cache management, content delivery speed, and resource optimization.

197. Dilley discloses a tiered distribution service provided in a content delivery network (CDN) having a set of surrogate origin (namely, "edge") servers organized into regions that provide content delivery on behalf of participating content providers. EX-1003, Abstract. Dilley uses a tiered cache hierarchy to store content in various levels of a cache network, storing more frequently accessed content in levels closer to the requesting clients (such as, at edge servers) and storing less frequently requested content in further caches (such as, at parent servers or origin servers). EX-1003, 2:12-17, 2:23-25, 2:63-3:2. This improves content

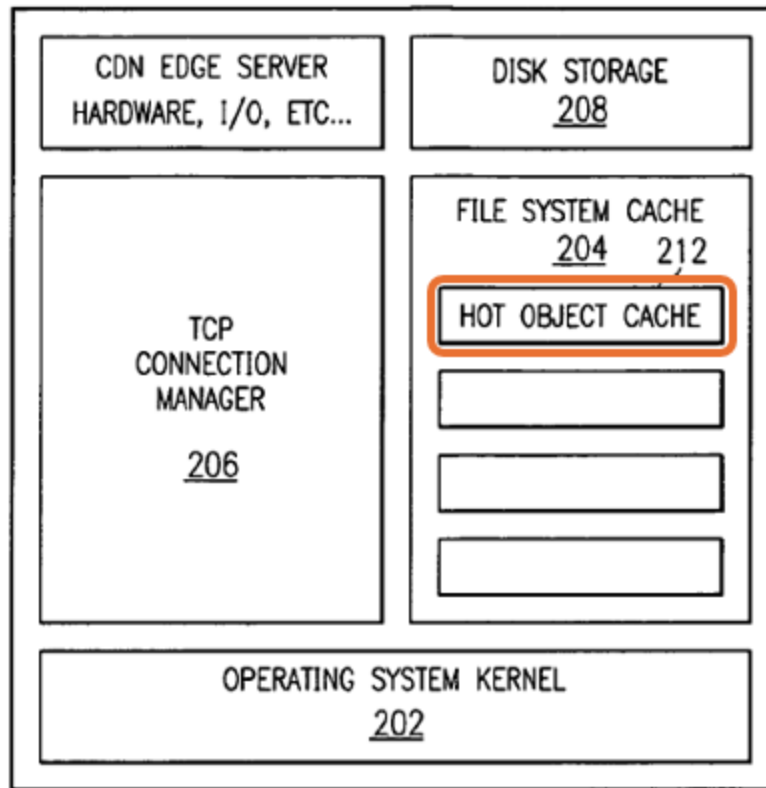
retrieval time for end users for popular content while providing cost effective storage options for the content provider. EX-1003, 5:65-6:10.

198. As Dilley explains, its system establishes a cache hierarchy such that when “a given object request cannot be serviced in the given edge region,” the request is provided to another server or tier, selected “preferably as [a] function of metadata associated with the given object request.” EX-1003, 2:42-56.

199. A POSITA would have understood that using “object specific metadata” to facilitate control over a tiered distribution of content in a CDN could be done using the popularity of the object. EX-1003, 2:23-25. Indeed, Dilley at least suggests using the popularity of objects to distribute content within a CDN and “reduce traffic to the origin servers for very popular” objects. EX-1003, 1:8-10, 2:26-31; *see also* EX-1003, 4:47-53, Fig. 2 (describing prior art configuration with “hot” object cache).

200. Managing popular objects in a cache was well known prior to Dilley. At Figure 2, Dilley provides a typical prior art machine configuration for a CDN content server which includes software to create and manage a “hot” object cache 212 for popular objects being served by the CDN. EX-1003, 4:47-53.

FIG. 2
(Prior Art)



EX-1003, Fig. 2 (annotated).

201. Dilley teaches or at least suggests a popularity service to protect the content origin servers from being overwhelmed by requests for popular objects. EX-1003, 5:65-6:10. This is because Dilley seeks to optimize the distribution of content across the tiered cache hierarchies based on different content types or access frequencies (i.e., popularity). EX-1003, 2:67-3:2, 5:65-6:10.

202. Pai further describes a popularity-based on-demand media distribution system that distributes media based on popularity such that more popular media

content is served from one on-demand media source and less popular media content is served from another on-demand media source. EX-1004, ¶[0003]. “For example, more popular media content may be maintained by a media source (e.g., a server or server cluster) located physically closer to client devices from which the media content may be requested,” such as an edge server, “[while l]ess popular media content may be maintained by a media source located physically further away from the client devices,” such as a parent or origin server. EX-1004, ¶[0013].

203. Pai also describes a system to identify the popularity of content by calculating popularity values and using these popularity values to dynamically redistribute the media content between different storage levels (i.e., level 1, 2, and 3) based on the popularity value. EX-1004, ¶¶[0013], [0029]. Pai does this to ensure that the most popular content is stored in the most accessible media source while the less popular content is stored in a media source reflective of its access frequency. EX-1004, ¶[0013].

204. In view of Dilley’s suggestion of a popularity service, a POSITA would have found it obvious to implement Pai’s dynamic popularity service, which includes multiple levels and popularity thresholds to more effectively manage objects of different popularity values.

205. A POSITA would have thus understood that Dilley and Pai were compatible and support the same content storage and delivery functionality. Indeed,

each system seeks to optimize content delivery while providing cost effective storage management solutions.

206. A POSITA would have accordingly implemented what Pai expressly discloses by providing a dynamic popularity-based media distribution system with multiple levels to Dilley's popularity service (e.g., providing a cache hierarchy for "popular content" to reduce the amount of traffic on content provider origin servers, EX-1003, 5:65-6:10; and finding it "particularly advantageous for flash-popular content" to be "distribute[d] closer to the requesting end users," EX-1003, 9:57-65). EX-1004, ¶[0013].

207. Pai provides an exemplary popularity service that would work well in Dilley's system. Indeed, Pai describes what Dilley refers to as "popularity." Dilley describes that content can be stored in various places based on popularity, and Pai provides specificity on how such a system would work.

208. The combination would have been nothing more than combining known elements in the prior art according to understood principles to yield predictable results. Both Dilley and Pai describe content delivery networks having many common features. For example, each uses a hierarchical structure of servers to deliver content to requesting clients such that when a requested object is not available from a first content server, the object may be retrieved from another tier of server. EX-1003, 5:4-21; EX-1004, ¶¶[0013], [0017], [0039]. Given the similarities

and overlap in their teachings, a POSITA would have had a high expectation of success in the combination.

209. A POSITA would have been motivated to combine Dilley and Pai, for example by supplementing Dilley's functionality with Pai's dynamic popularity service, for the advantages Dilley contemplates, such as providing an efficient system that can store and deliver popular content more quickly and efficiently to end users. EX-1003, 1:30-32 (explaining that an effective CDN serves frequently accessed content from an optimal location), 2:26-31 (explaining that a tiered distribution method allows end users to obtain desired content quickly and reliable without overwhelming the origin server with requests for popular content). Thus, employing a multi-tiered dynamic popularity service, as taught by Pai and suggested by Dilley, would have improved Dilley's ability to handle high volumes of requests for different levels of popular content without overloading the origin servers and would have been able to balance the request across the hierarchy of servers. A POSITA would have also understood that the tiered cache hierarchy of Dilley would benefit from the real-time popularity calculations and dynamically adjusted content placement provided in Pai, resulting in better performance and more efficient use of resources.

210. Further, the combination would have allowed more content to be stored and more effectively managed based on the different levels of popularity. This would

have increased the ability of Dilley's system to direct a requesting client to the appropriate level of the cache hierarchy or to allow Dilley's system to retrieve content from the appropriate server, thereby increasing system performance. A POSITA would have understood that by using Pai's real-time popularity values to redistribute content, the system would ensure that only popular media items occupy the storage space at the faster storage levels (i.e. level 1 or edge server level). This would prevent unpopular content from occupying valuable storage space at these frequently accessed levels, and thus the system would optimize the use of its limited storage capacity.

211. Thus, a POSITA would have been motivated to combine Dilley and Pai because doing so would have reduced storage costs and reduced the retrieval latency of popular content items. EX-1003, 5:16-21; EX-1004, ¶[0002]. Moreover, the combined system would have allowed for improved scalability of the CDN because it would combine the existing tiered hierarchy of Dilley with the dynamic popularity-based distribution system of Pai, which would account for evolving media consumption patterns by clients.

212. Modifying Dilley to implement the popularity service detailed in Pai would have been well within the capabilities of a POSITA and have been nothing more than combining known features in the prior art according to known methods to yield predictable results.

6. Further Combining Dilley and Pai with Wang

213. A POSITA would have found it obvious to combine Dilley and Pai with Wang to provide a system that includes, in certain situations, instructions to not cache a resource at the edge when it is obtained from the content server.

214. A POSITA would have understood that Dilley-Pai was compatible with the teachings of Wang. Similar to Dilley and Pai, Wang provides for a tiered content distribution network. EX-1005, Abstract, ¶[0005] (describing a hierarchical service and content distribution system). And like Dilley and Pai, Wang discloses a three-level content distribution network where each level may store content of different popularity thresholds. EX-1005, ¶¶[0047]-[0049]. It discloses a “network system having multiple levels for improving performance of the content delivery network via a hierarchical service network infrastructure design.” EX-1005, ¶[0005].

215. Wang’s system establishes a framework for communication between service managers and devices, ensuring data integrity and reliability in a complex content delivery network. EX-1005, ¶¶[0022], [0047]-[0049]. The network is managed hierarchically, with Level 1, 2, and 3 service managers responsible for different tiers of the network. EX-1005, ¶¶[0047]-[0049]. Wang explains that the directory system supports content caching at multiple service levels. EX-1005, ¶[0123]. Content can be cached at level 1, 2, or 3, and can be retrieved dynamically. *Id.* As Wang explains, a client request for a resource is forwarded to the available

caching service engines. *Id.* If the content is not found at level 1, the request escalates up the service hierarchy, from level 1 to level 2 and then to level 3, to find the content. *Id.*

216. Wang's hierarchical caching system also includes a mechanism to identify certain resources that should not be cached by the network service engines. EX-1005, ¶[0123]. For instance, when the request is escalated to level 3, the service manager will return attributes of the resource, including an indication of whether that resource is "cacheable or not." EX-1005, ¶[0123]. Thus, if it is marked as not cacheable, a POSITA would have understood this to be an instruction to not cache that resource at level 1 when it is obtained from level 3.

217. A POSITA would have found it obvious to combine Dilley and Pai with Wang. For example, a POSITA would have combined Wang's identification of resources as not to be cached at lower levels to Dilley and Wang's hierarchical content delivery network such that certain content from level 3 would not be cached at level 1.

218. A POSITA would have been motivated to combine this feature with Dilley-Pai to avoid the cost and resource expenditure of caching content that, e.g., becomes quickly stale. Indeed, not everything can be cached in a CDN due to storage and cost limitations. *See* EX-1013, ¶[0011] ("It is typically not feasible to store all objects on the edge servers. The main difficulty is due to the fact that many such

objects are very large The storage and rack space required to accommodate often large and sometimes rarely requested objects at every edge server can be cost prohibitive as the number of customers grows and the number of their objects increases.”). A POSITA would have further understood that implementing Wang’s teachings would have improved system operation.

219. A POSITA would have understood that Dilley and Pai were compatible with the teachings of Wang. For example, Dilley similarly discloses that certain content may be served from the origin server because the requested content is unavailable or “un-cacheable.” EX-1003, 2:12-17. And each Dilley and Pai include a similar tiered cache hierarchy as that described in Wang. EX-1003, Fig. 3; EX-1004, Fig. 2. Thus, combining Wang’s instruction to not cache a resource at the edge when it is obtained from the origin would have been nothing more than combining known features in the art according to understood principles to yield predictable results. The result would have been a tiered CDN with a hierarchical cache that escalates a request to the next level (edge server to parent server to origin server) to find content and includes an instruction to not cache that content at the lower-level servers if it does not meet the criteria or thresholds set by the CDN, as explained in Pai. EX-1004, Fig. 6. A POSITA would have had a reasonable expectation of success in the combination because including an identification of cacheable or not was known and understood in the art. This combination would have improved the

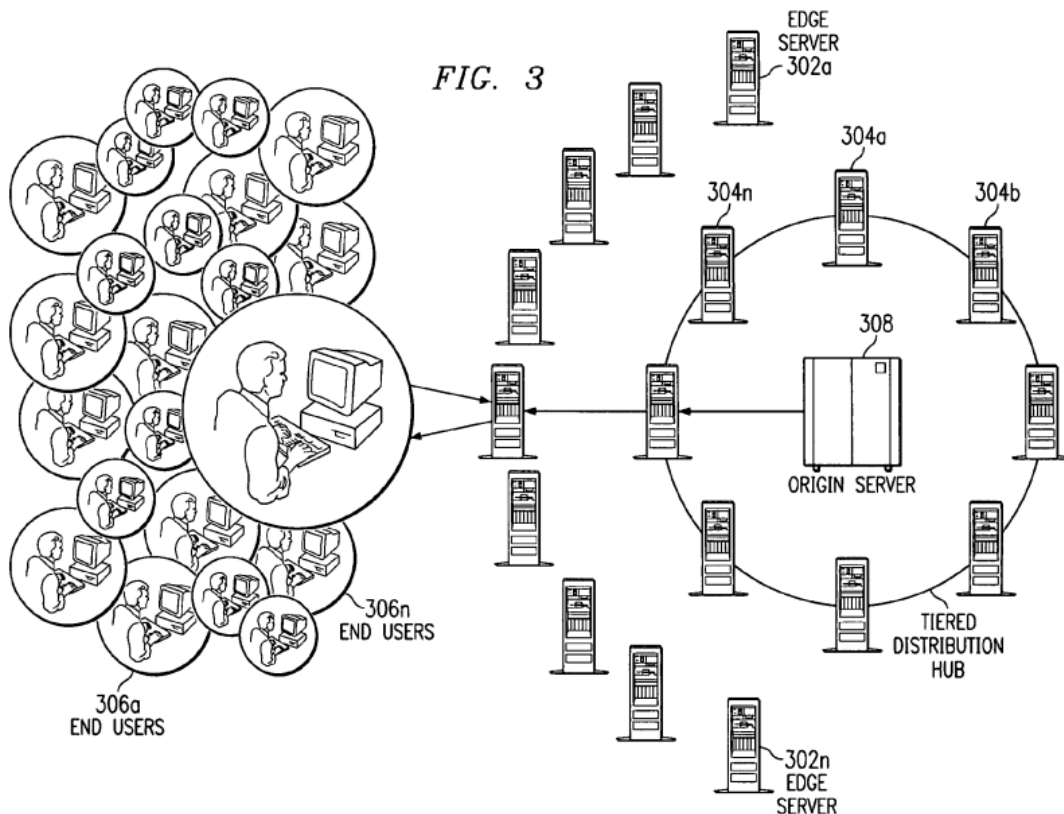
capacity, scalability, and performance of the Dilley and Pai system because it would selectively cache information at the lower levels without filling each cache merely because the content was requested.

7. Claim 1

a. 1(preamble): “A method of content delivery in a content delivery network comprising:”

220. The combination of Dilley, Pai, and Wang discloses this limitation.

221. Dilley is entitled a “[m]ethod and system for tiered distribution in a content delivery network,” and as the name implies, further discloses “[a] tiered distribution service” provided “in a content delivery network (CDN).” EX-1003, codes 54 and 57; *see also id.*, Abstract, 1:12-40.



EX-1003, Fig. 3 (depicting tiered distribution service in a CDN).

222. Dilley discloses improvements for a CDN by using a cache hierarchy to improve the distribution of content across multiple server levels, thereby reducing the load on the origin server, improving content delivery to user, and reducing storage costs for content providers. EX-1003, 2:57-3:2, 5:65-6:15.

223. Dilley therefore teaches a method of content delivery in a content delivery network, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

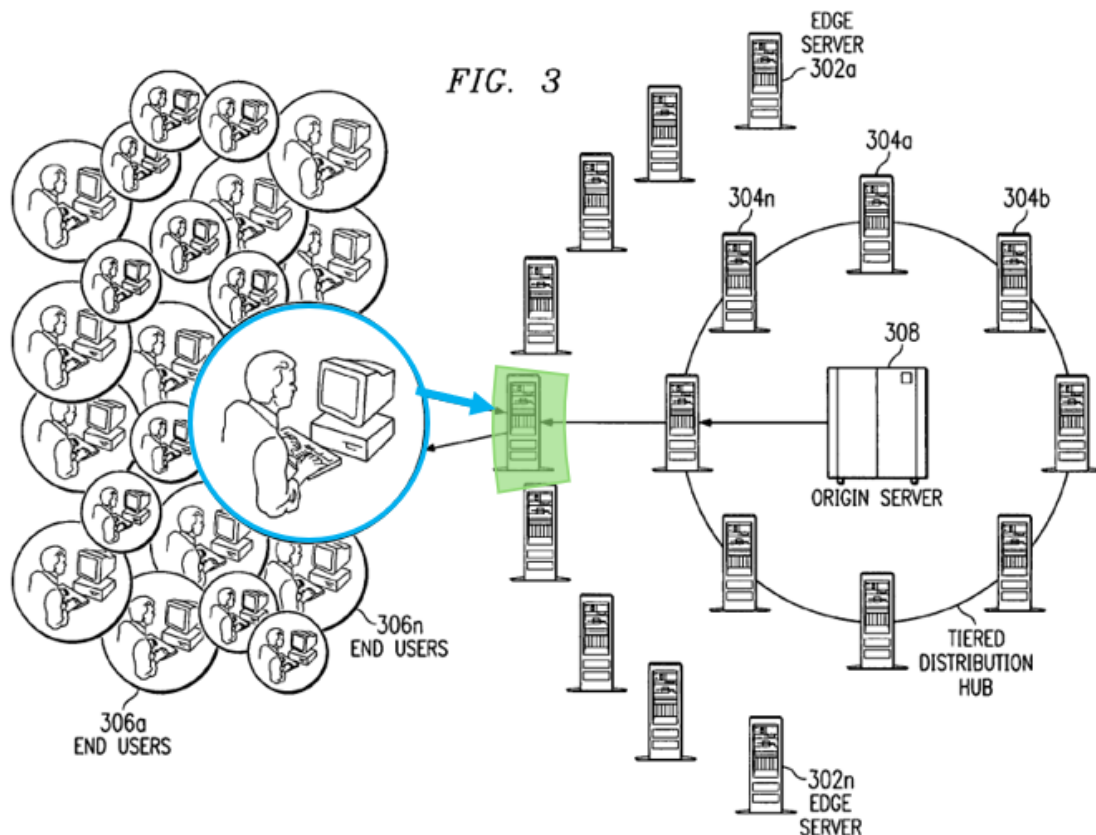
- b. 1(a) “receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;”**

224. The combination of Dilley, Pai, and Wang discloses this limitation.

225. Dilley teaches receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network.

226. Dilley’s CDN employs a tiered distribution structure, where content is cached and served to requesting users from edge servers, parent regions, and, when necessary, directly from the origin server. EX-1003, 4:63-5:21. Dilley explains that it’s tiered infrastructure of servers includes at least “a set of edge servers and a hierarchical set of ‘parent’ or hub servers,” where the edge server receives user requests for content. EX-1003, 4:63-5:21.

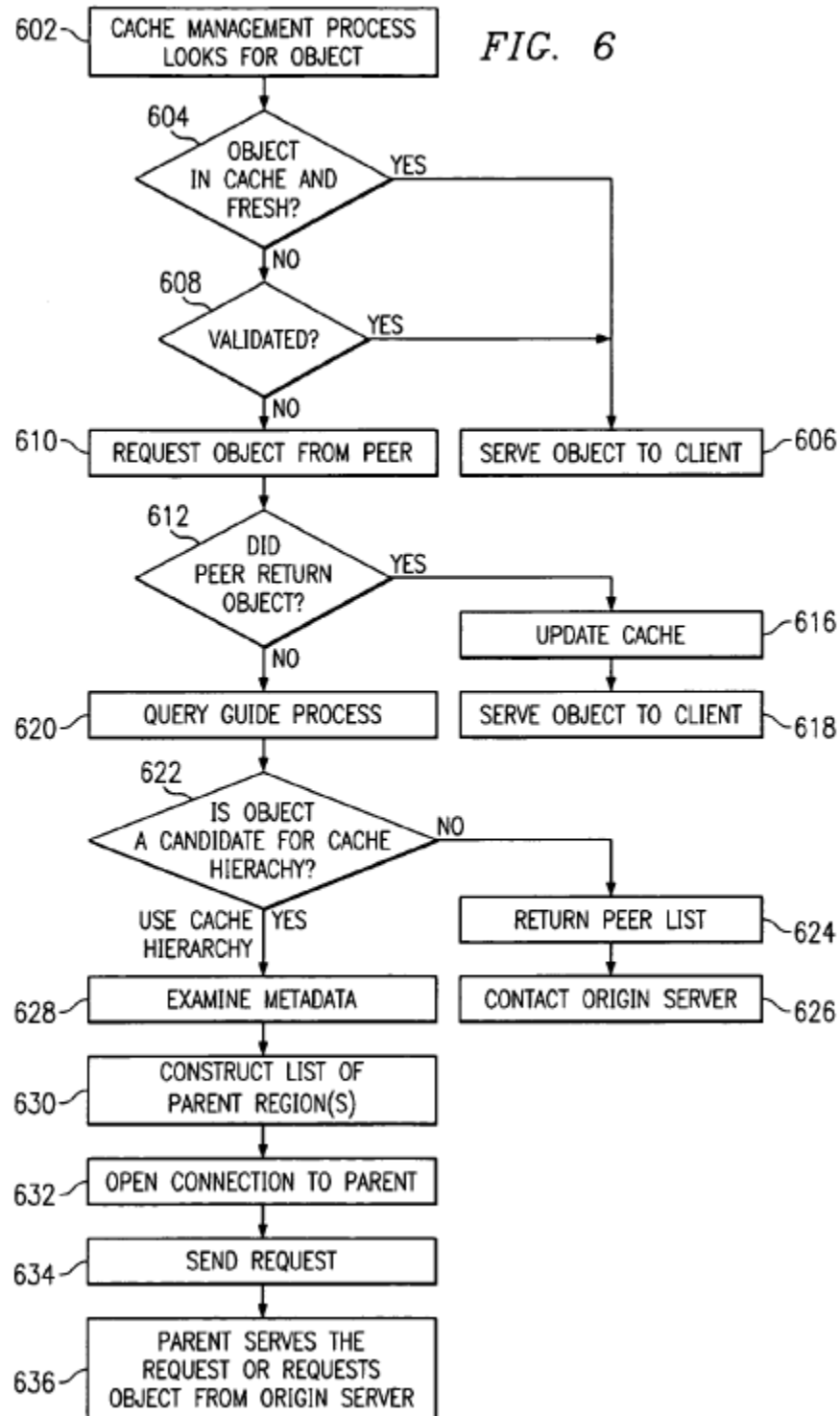
227. As an initial step, an end user or requesting device makes a content request to Dilley's system. This request is routed to an edge server of the CDN. EX-1003, 4:63-5:21. Dilley then discloses that the edge server "receives end-user requests for origin server content and serves this content" from its cache "if it can do so." EX-1003, 7:50-53. This is shown in Dilley Figure 3.



See, e.g., EX-1003, Fig. 3 (annotated, showing an edge server receiving the request from an end user).

228. Dilley further provides that "in a preferred operation of the present invention" an "edge server receives a client request for an object," resulting in the

commencement of the flowchart in Figure 6. EX-1003, 8:3-7 (explaining that “when a given edge server receives a client request for an object[, t]he routine begins”).



EX-1003, Fig. 6

229. Dilley also discloses that when a content request is made by an end user and received by an edge server, the edge server of the CDN will determine whether the requested content is present in its cache and is fresh (i.e., up-to-date or unexpired). EX-1003, 8:3-11. If it is present and fresh, the edge server will directly serve that content to the requesting user. *Id.* Otherwise, it will attempt to communicate with other servers, such as other edge, parent, or origin servers to obtain the content. EX-1003, 8:11-21.

230. Dilley therefore teaches receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

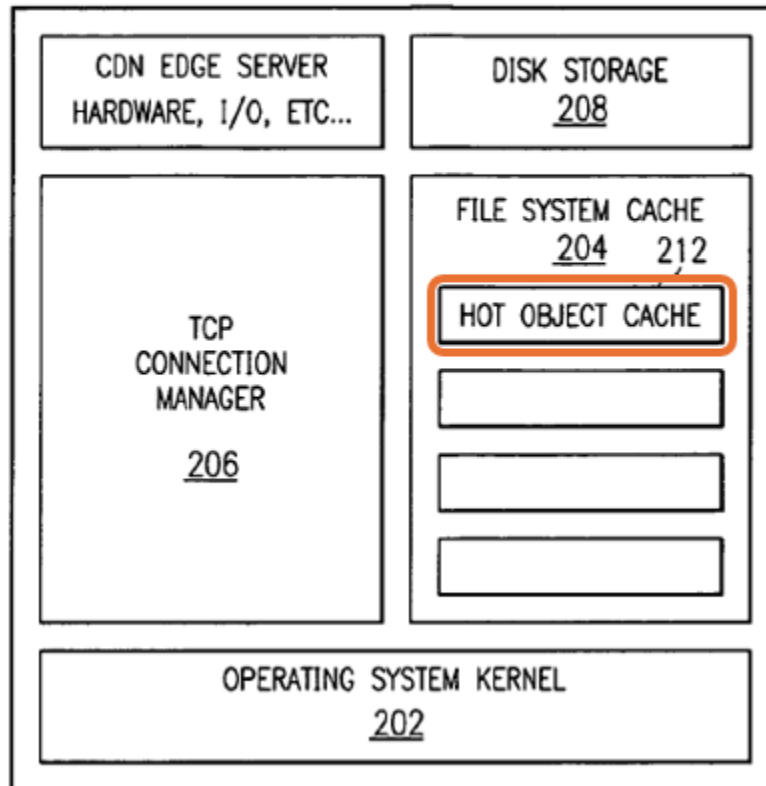
c. 1(b) “accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource;”

231. The combination of Dilley, Pai, and Wang discloses this limitation.

232. Dilley explains that user requests are received at the edge servers. The system then “funnel[s] requests that cannot be serviced in edge regions” (i.e., when “the object is not in cache”) through “parent clusters” or parent servers. EX-1003, 5:65-6:4. By serving these requests from the edge server cache or funneling these requests to parent servers, the Dilley system “reduces the amount of traffic on

content provider origin servers for very popular content.” EX-1003, 5:65-6:10. That is because Dilley teaches that CDN servers regularly include a popularity service that caches popular items at the edge. *Id.*; *see also* EX-1003, 4:47-62 (explaining that a typical “CDN content edge server” contains a “hot” object cache 212 to manage popular objects being served by the CDN).

FIG. 2
(Prior Art)



EX-1003, Fig. 2 (annotated) (identifying a hot object cache that identifies hot or popular objects, similar to the function performed by a popularity service).

233. Dilley further provides that its cache hierarchy “uses a hierarchy of servers to distribute content closer to the requesting end users,” which is “particularly advantageous for flash-popular content.” EX-1003, 9:57-65; *see also id.*, 1:7-10.

234. Dilley’s cache hierarchy operates based on metadata attributes associated with each content resource. EX-1003, 6:14-40, 7:31-35. The embedded metadata specifies certain criteria about the content such as the desired caching and delivery behavior, as well as the content’s popularity, size, and content type. EX-1003, 6:1-15. This metadata controls whether and where the content is cached and from where the content is served. EX-1003, 7:31-35.

235. Because Dilley explains that a key advantage of its system and method is to insulate origin servers from large quantities of requests for popular and/or large files, a POSITA would have understood that Dilley’s system includes a popularity service to determine the popularity of each content item and, based on the metadata of these items, would appropriately distribute the content across various tiers of the CDN hierarchy.

236. To the extent Dilley does not alone teach accessing a popularity service to determine a popularity designation associated with the requested resource, Pai does. As explained in Section VII.B.5 above, while Dilley teaches the advantages of using a hierarchical infrastructure to distribute popular content closer to the edge, it

leaves to a POSITA how the popularity service would work in application, and a POSITA would have been motivated to implement Pai's teachings regarding its popularity service in Dilley's system.

237. As discussed above in Section VII.B.2, Pai details a system and method for popularity-based on-demand media distribution. EX-1004, ¶[0003]. Pai discloses a popularity service that monitors and calculates a content's popularity and, based on this calculated popularity value, dynamically adjusts the content's storage location to improve the delivery of content to users while efficiently managing the limited storage capacity of the system. EX-1004, ¶¶[0013], [0020].

238. Pai discloses a system with a three-level media source infrastructure (level 1, 2, and 3). EX-1004, ¶[0020], Figs. 2, 6. Pai discloses that popular media items are distributed to level 1, which are quick and easily accessible storage sources. EX-1004, ¶¶[0020], [0039]-[0040]. Content with lower popularity values are assigned to level 2 or 3 media sources which have typically slower, longer-term storage capabilities. EX-1004, ¶¶[0020], [0042]-[0045]. Pai's system and dynamic distribution and redistribution based on content popularity ensures that popular content may be retrieved the fastest, while less popular content in either an intermediate level or high level may be stored more efficiently and may not need to be retrieved as quickly based on the lower volume of requests for this content. *Id.*

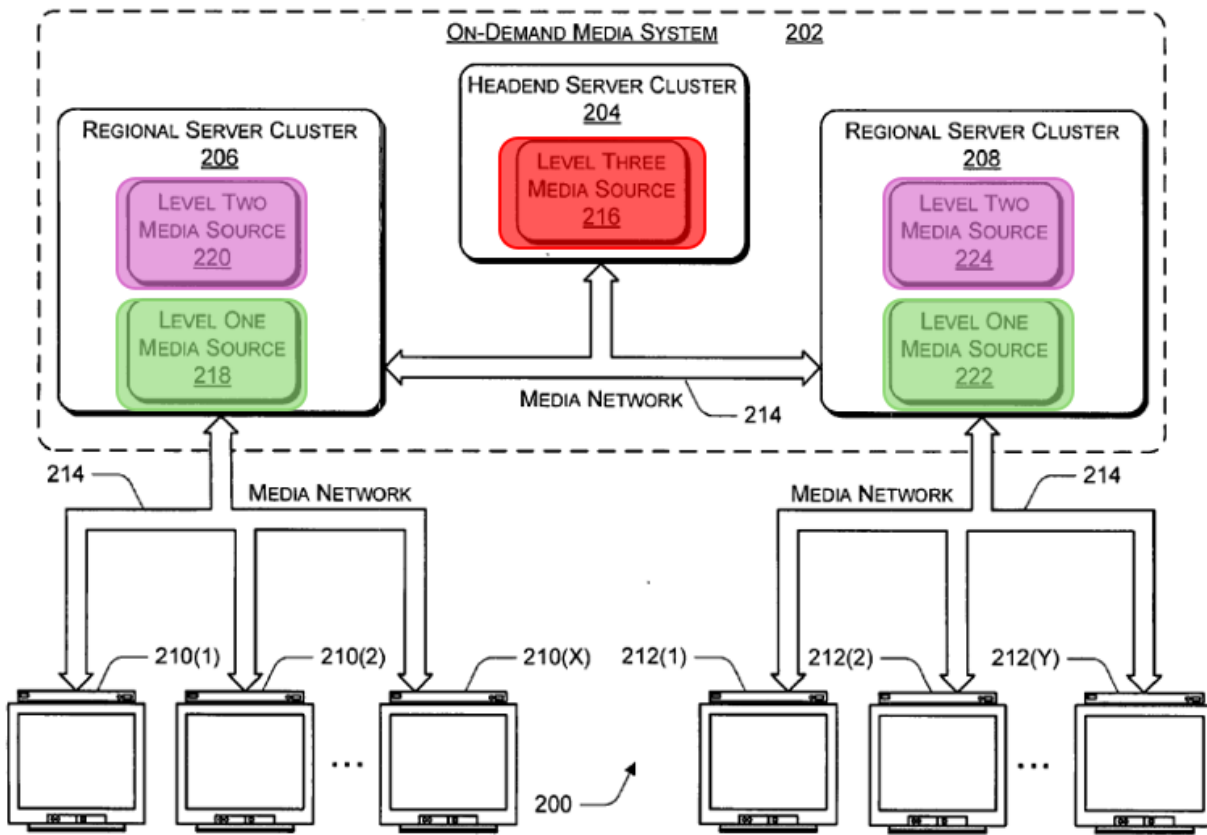


Figure 2

EX-1004, Fig. 2 (annotated showing level one, level two, and level three media sources for the most popular, less popular, and least popular content, respectively).

239. Pai explains that as media content is consumed by users, its popularity can change over time. EX-1004, ¶¶[0003], [0013], [0029]. It accordingly implements a dynamic redistribution process to ensure that the most popular content is stored closest to the edge in the fastest-accessed media source (level 1). EX-1004, ¶¶[0013], [0020]. To do this, the system continuously recalculates the popularity value of each media content based on user demand. EX-1004, ¶¶[0013], [0029]; see

e.g., id., claim 8. For instance, Pai explains that it tracks the requests made by users, and when a media item has been ordered by a user, its popularity value is updated accordingly, potentially triggering a redistribution to a lower level. EX-1004, ¶¶[0039], [0047].

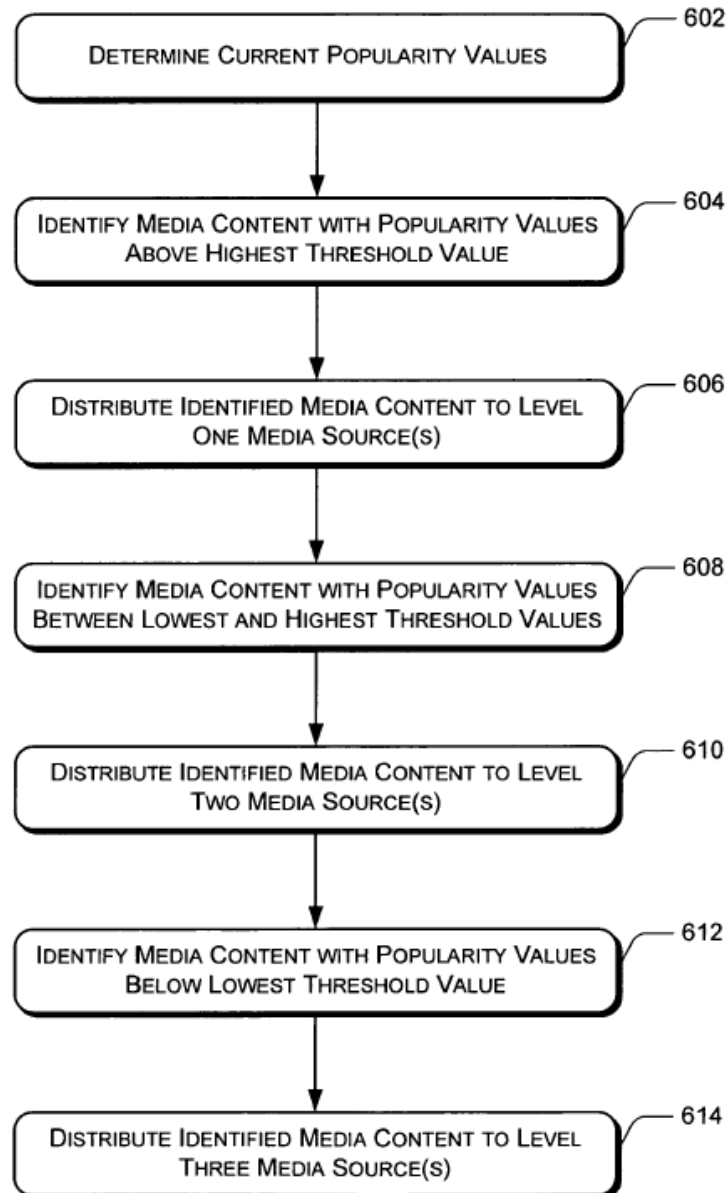


Figure 6

EX-1004, Fig. 6.

240. Through monitoring and dynamically adjusting a content's popularity value, Pai discloses accessing a popularity service to determine the popularity associated with the requested resource. EX-1004, ¶¶[0039], [0047]. Indeed, as Pai explains, using its popularity service, content is appropriately distributed throughout the system such that when a user requests a media resource, the system retrieves it from the relevant media source based on the content's current storage location (selected based on its current popularity). *See e.g.*, EX-1004, ¶¶[0039]-[0045].

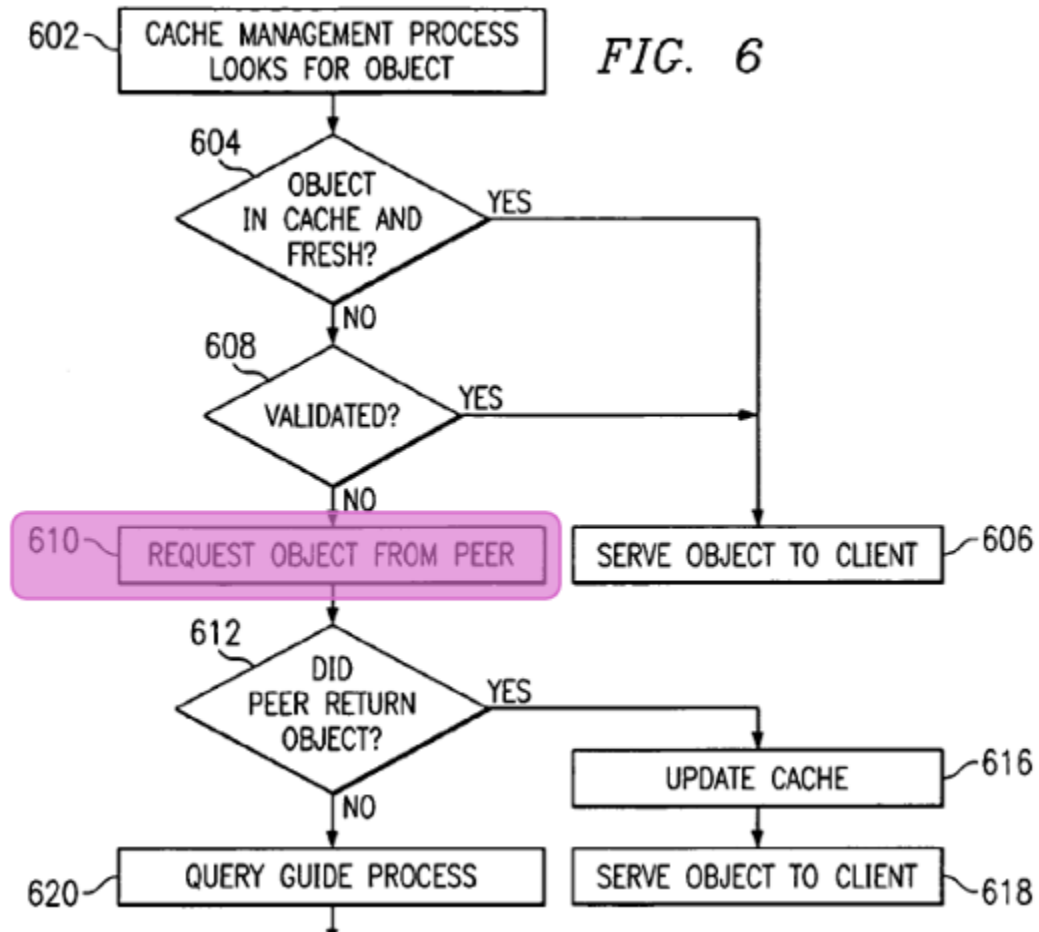
241. Dilley in combination with Pai therefore renders obvious accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource, and further combining Dilley and Pai with Wang, as discussed above, would not alter these teachings.

d. 1(c) “requesting the resource from a second server of the content delivery network;”

242. The combination of Dilley, Pai, and Wang discloses this limitation in at least two ways.

243. First, Dilley explains that when a request is received at an edge server, the server first looks for the object in the edge server cache (step 604). EX-1003, 8:5-11. If it is, the edge server will serve the object to the client. *Id.* Otherwise, the edge server will request the object from a different server (step 610), and then update the cache to include the requested content. EX-1003, 8:11-18. For instance, Dilley

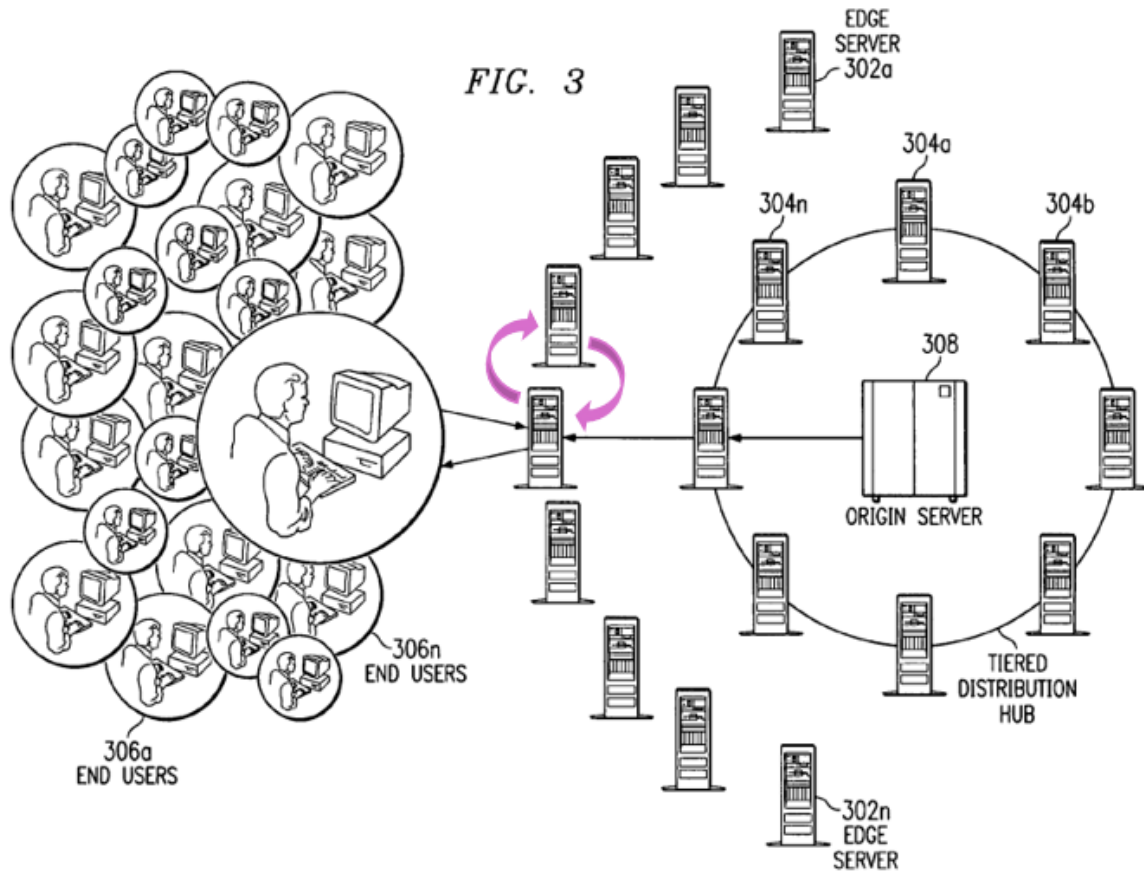
explains that “[i]f the objection is not in cache or needs to be validated, the edge server makes a request (e.g., via a cache sharing protocol) to other edge servers in the region (each a ‘peer’) to see if one of them has a fresh copy of the object.” EX-1003, 8:3-15.



EX-1003, Fig. 6 (abbreviated and annotated).

244. Thus, Dilley discloses specifically that the first server (i.e., the edge server receiving the request from the end-user) will request that resource from a

second server (i.e., a peer server of the same tier of servers) when the first edge server does not have the requested content in its cache. EX-1003, 5:22-30, 8:11-18.

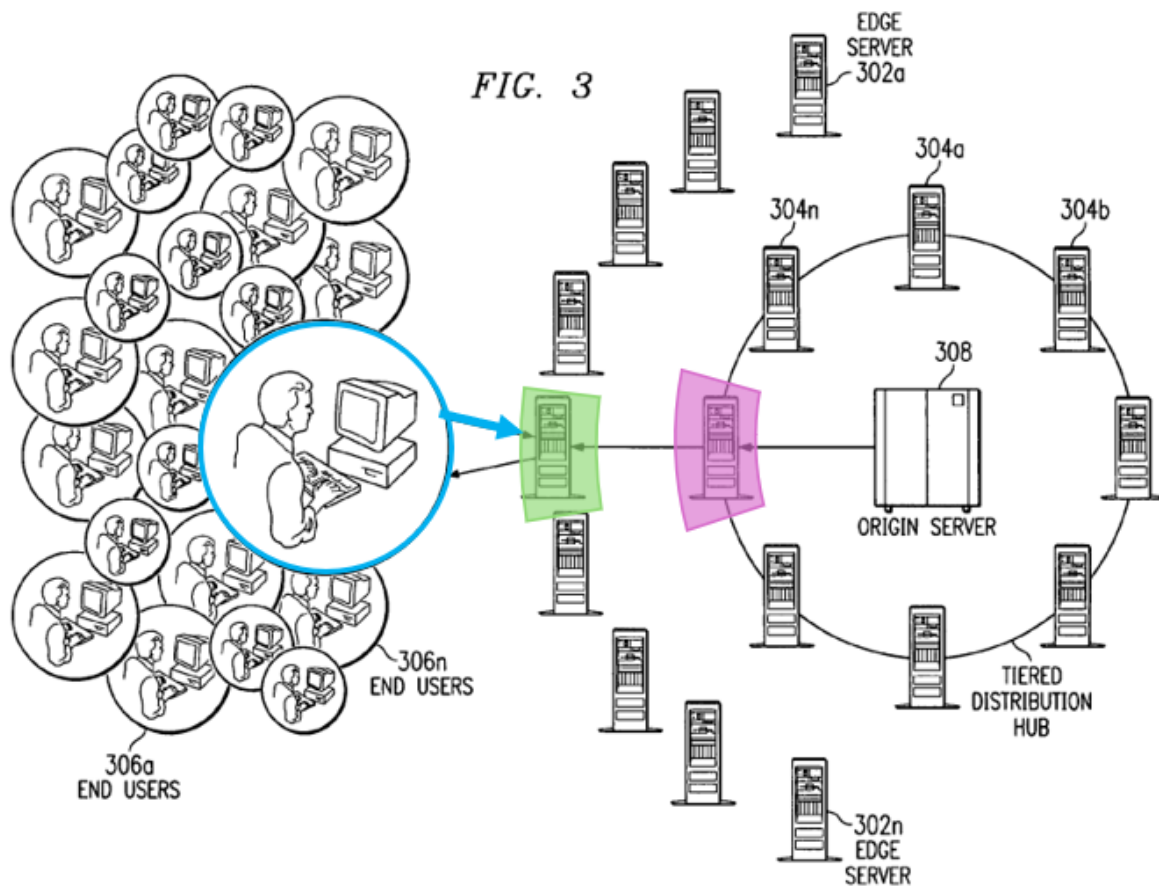


EX-1003, Fig. 3 (annotated to identify an exemplary communication among peer servers).

245. A POSITA would have further found it obvious that the second server is a peer server of the first tier of servers because upon a cache miss at the first server, requesting the content from a peer server in the same tier would reduce latency, ensuring quicker and more reliable delivery of the content and would have further insulated the parent and origin servers from a large volume of requests (in

furtherance of both the '573 patent's and Dilley's goals). EX-1003, 1:65-2:3, 6:4-10.

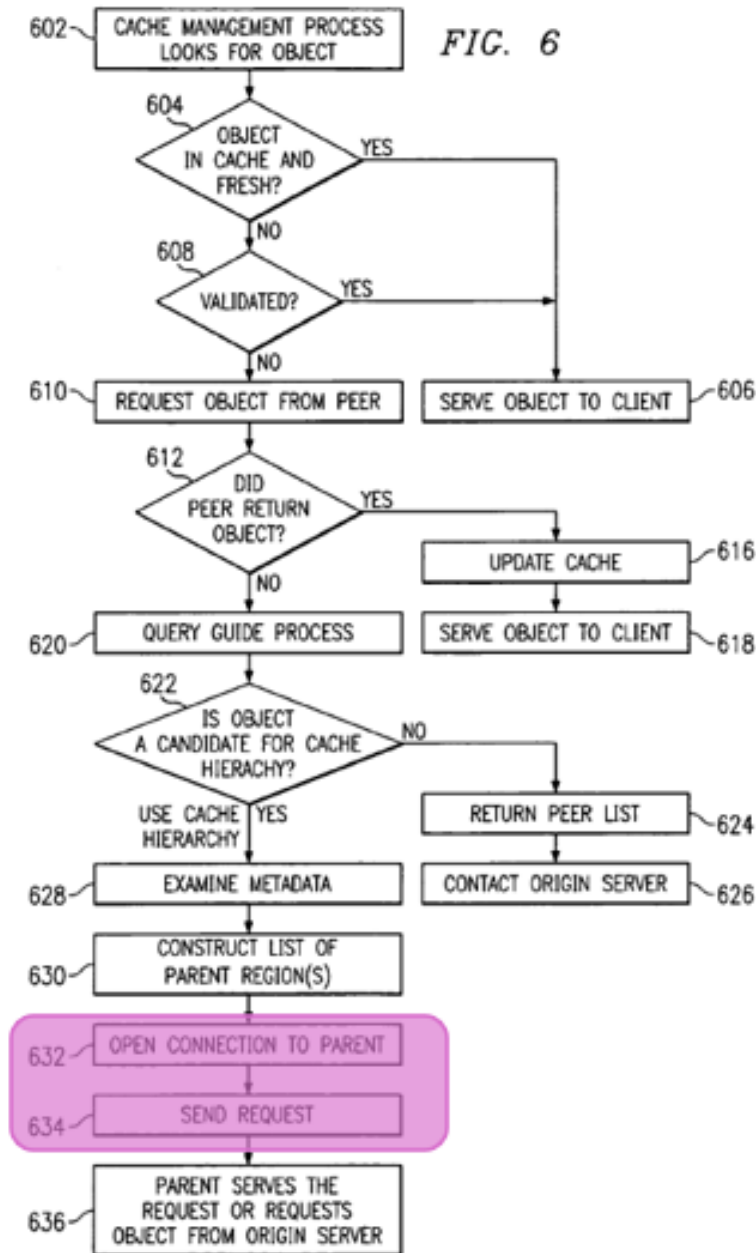
246. Second, Dilly discloses a method of using “a cache hierarchy[] in which requests for content that cannot be serviced in edge servers are funneled through intermediate nodes,” i.e., requesting a resource from a second server of the CDN. EX-1003, 2:5-17. As Dilley explains, its CDN comprises an “edge server region” of edge servers 302a-n, and either a single parent region or a subset of parent server regions containing parent servers 304a-n. EX-1003, 2:37-46.



EX-1003, Fig. 3 (annotated).

247. When it is determined that the edge server cannot service the end user's (306a-n) request for a resource, "instead of contacting the origin server, the request is provided to either the single parent region" or to one of the subsets of parent server regions "for handling," (i.e., a second server of the CDN). EX-1003, 2:46-56. In this instance, the request is serviced, if possible, by the second server (the parent server), and "is only forwarded on to the origin server if it cannot be serviced" by the parent server. EX-1003, 2:52-56; *see also id.*, 5:27-41.

248. This is further illustrated in Figure 6 of Dilley's method. Specifically, Dilley explains that the "edge server receives a client request for an object." EX-1003, 8:3-5. If the edge server does not have the requested object, it makes a request to another edge server in the region (i.e., a peer server) to see if a copy of the object is available (see step 610 below). EX-1003, 8:7-15. If the peer server is also unable to return the object, Dilley will communicate the request to a parent server of the CDN (see steps 632 and 634 below). EX-1003, 8:38-48 (explaining that the edger server "sends the request to a server in the parent region).



EX-1003, Fig. 6 (annotated).

249. Dilley therefore teaches at least two ways in which it requests the resource from a second server of the content delivery network, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

- e. **1(d) “processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network;”**

250. As in Ground 1, I do not have an opinion on whether the '573 patent contains written description support for this limitation. But if it does, the combination of Dilley, Pai, and Wang discloses this limitation.

251. As explained above, if Dilley's edge server does not have the requested content available, it will try to request that content from another server, such as a peer server or a parent server higher on the hierarchy. EX-1003, 5:4-21, 7:31-40. This second server will either serve the content from its cache or, if it's not available, instruct the first server to request the content from server even higher on the hierarchy, such as a parent or origin server. EX-1003, 7:31-40, 8:53-58.

252. While Dilley discloses that the content can be retrieved from the origin server when it is unavailable at the edge and parent servers, it does not provide specifics on how that retrieval happens. Instead, it leaves it open to a POSITA how the communications and instructions among the tiered servers would work.

253. A POSITA would have understood that, in the example when the second server is a parent server, Dilley's parent server could have sent a redirect instruction to the edge server, directing the edge server to obtain the resource from the origin content server instead. EX-1003, 7:15-30. A POSITA would have

understood this and have been motivated to implement the redirect instruction this way from the description in Wang. *See e.g.*, EX-1005, ¶¶[0123]-[0130].

254. Wang discloses a CDN managed hierarchically into three tiers (levels 1, 2, and 3), where each level supports content caching. EX-1005, ¶¶[0047]-[0049]. Wang explains that information is distributed from lower levels (level 1) to higher levels (level 3) to promote efficiency and avoid conflicts. EX-1005, ¶[0046]. For instance, when a user requests content, the system will select a level 1 server to retrieve the content. EX-1005, ¶[0123]. If the content is not found at level 1, the request escalates from level 1 to level 2 and eventually to level 3 to find the content. *Id.* In this instance, Wang explains that the level 1 server will process a redirect instruction to obtain the content from the origin server. EX-1005, ¶[0123] (explaining, for not-cacheable content, the level 1 server will redirect the request to the origin server), ¶[0124] (explaining the same redirect process to the origin server is supported for cacheable content).

255. A POSITA would have understood that these redirects were instructions to obtain the content from the origin content server. And a POSITA would have understood that Wang's redirect procedure would have added benefits because the edge server would then process the redirect instruction, e.g., by establishing communication with, and requesting or obtaining the content from the origin server. It would thus be expecting the response directly from the origin server.

Otherwise, if the parent server requests the content from the origin server, and the origin server delivers that content to the edge server, the edge server may not be expecting the resource from the origin (since it originally requested the content only from the parent server).

256. Having the parent server redirect the edge server to fetch the content from the origin server would decrease latency as it would remove the parent server from acting as a “middleman,” and would allow the edge server to communicate with and receive content directly from the origin server. Otherwise, the parent server would request content from the origin server, would need to receive that content, and then would need to pass that content back down the chain to the edge server, providing for more latency or room for error along the chain. Thus, a POSITA would have understood that processing at the edge server a redirect instruction from the parent server to obtain the content from the origin server would reduce delays and provide the end users with a quicker and smoother experience in obtaining the requested content.

257. In addition to decreasing latency, this process would allow the parent server to remain available to service other requests received by the CDN. For instance, if the parent server sends a redirect instruction to the edge server to get the content from the origin, the parent server is then available to receive other requests from edge servers seeking content that was not available in the edge server’s cache.

If the parent server was tasked with forwarding the request to the origin and playing “middleman” to receive the content from the origin and forward it to the edge server, it would be tied up and would not have the bandwidth to assist with other requests, thereby causing the CDN as a whole to service less requests.

258. Each of the above benefits would apply equally to the example where the second server is a peer server in which the peer server could have sent a redirect instruction to the edge server, directing the edge server to obtain the resource from a parent or origin server instead. Thus, a POSITA would have also found it obvious for the edge server to process a redirect instruction from a peer server when the peer did not have the requested resource.

259. A POSITA would have understood that having the second server (either a peer or parent server) send a redirect instruction to the edge server to receive the content from a content server would have been one of very finite design choices and would have been obvious in the context of at least Dilley and Wang.

260. Dilley and Wang therefore teach processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

- f. **1(e) “receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network; and”**

261. The combination of Dilley, Pai, and Wang discloses this limitation.

262. Dilley discloses a system that leverages a cache hierarchy, controlled by embedded metadata, to efficiently distribute content across a multi-tiered CDN. EX-1003, 5:65-6:15. As Dilley explains, its system reduces the number of requests sent to the origin server and improves the delivery speed for larger and more popular content. EX-1003, 6:4-10. To make this possible, Dilley cannot cache every requested item at the edge or parent servers because doing so would take up valuable storage space at those server levels. For instance, while Dilley supports content caching at multiple levels (edge, parent, and origin servers), a POSITA would have understood that cache storage is limited at each of these levels and is likely more restricted at edge servers. EX-1003, 9:50-56; EX-1004, ¶¶[0017], [0020]. Thus, as a POSITA would have understood, the system would need to implement rules to determine the content that should be and should not be cached once it is received from the origin server.

263. Dilley’s system uses embedded metadata to implement these rules. EX-1003, 6:14-15. As it explains, the default setting could be to not use the cache hierarchy, preserving the hierarchy for only certain content. EX-1003, 6:16-40.

Thus, the default metadata tag would indicate that the edge server is to go directly to the origin for certain requested content.

264. A POSITA would have understood that Dilley's system would use this rule (to not use the cache hierarchy, and therefore, not cache content) for content that should not be cached. A POSITA would have understood that some of these "not to be cached" items could include content that is dynamic (i.e., content that updates in real-time and would become immediately stale) or content that is personalized (either in the specific session or to the specific user).

265. A POSITA would have further understood that since Dilley's system is particularly interested in insulating the origin server from requests for flash-popular content, Dilley's system would operate to cache the popular content (indicated by its embedded metadata) at the parent or edge servers. EX-1003, 9:57-65; *see also id.*, 1:7-10. Thus, the default rule would be to not cache items unless they are indicated as popular items.

266. To the extent Dilley does not alone teach receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server, the Dilley-Pai-Wang combination teaches this limitation.

267. As explained above, when a request for content is made by a user, the system will select a level 1 server to retrieve the content. *Supra* §VII.B.7.e; EX-

1005, ¶[0123]. If the content is not found at this level 1 server, the request escalates up the server hierarchy from level 1 to level 2 and eventually to level 3 to find the content. *Id.*

268. Wang discloses that content items may include cache attributes, identifying whether they should be cached or not. EX-1005, ¶[0123]. The system can include content attributes that identify the content as not cacheable. *Id.* Specifically, Wang discloses that when the content is unavailable at levels 1 and 2 and is obtained from level 3, it may include an “indication” that it is not to be cached (i.e., a no-cache instruction). *Id.* A POSITA would have understood that this no-cache attribute attached to the content’s metadata would be an instruction for the level 1 server to not cache the resource at level 1 when it is obtained from the level 3 server (i.e., the origin server).

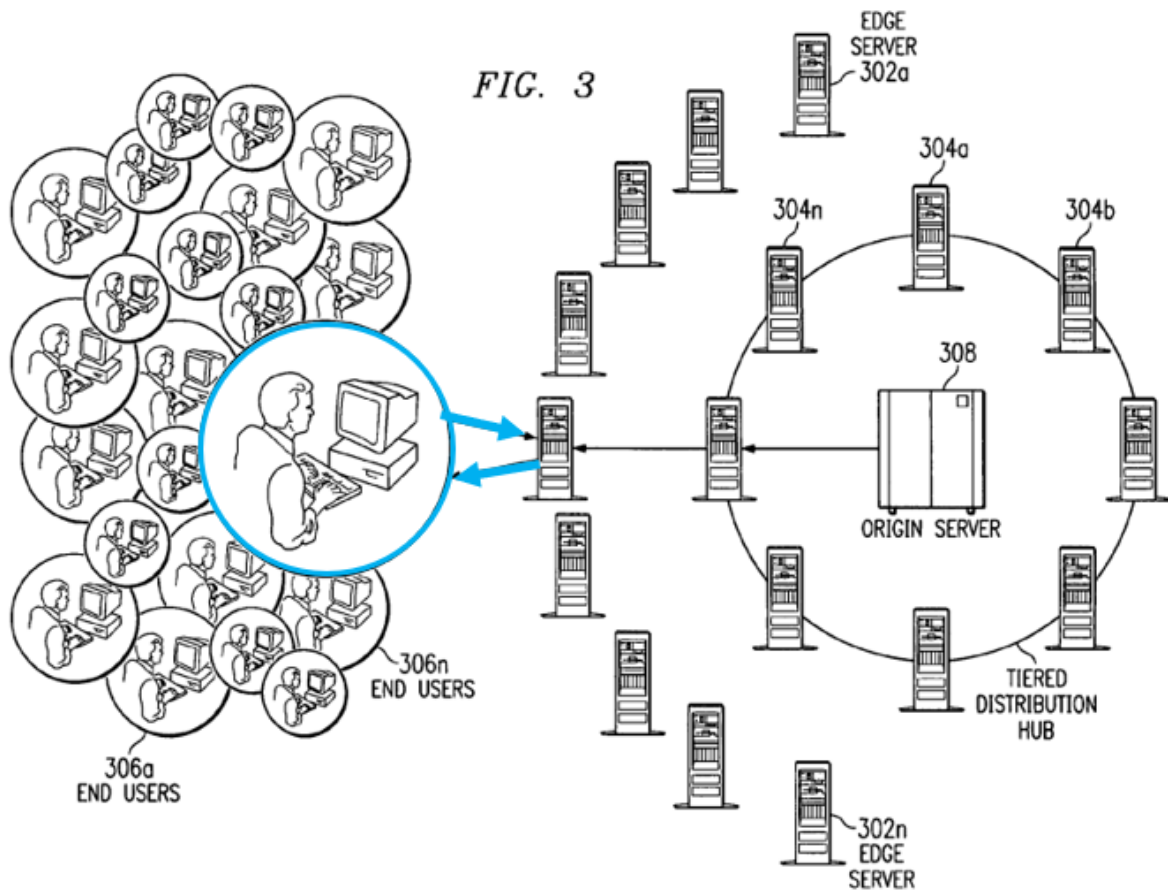
269. As above, a POSITA would have understood the no-cache instruction ensures that not every content item is cached at the lower levels. A POSITA would have also understood that there are some content items that should not be cached at the edge servers. These include not only dynamic or private content but could also include unpopular content in the context of Dilley, Pai, and Wang, where the storage capacity at the edge is more limited and would not want to be consumed by infrequently requested resources.

270. Dilley in combination with Pai and Wang therefore teaches receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network.

g. 1(f) “providing the obtained resource to the requesting device.”

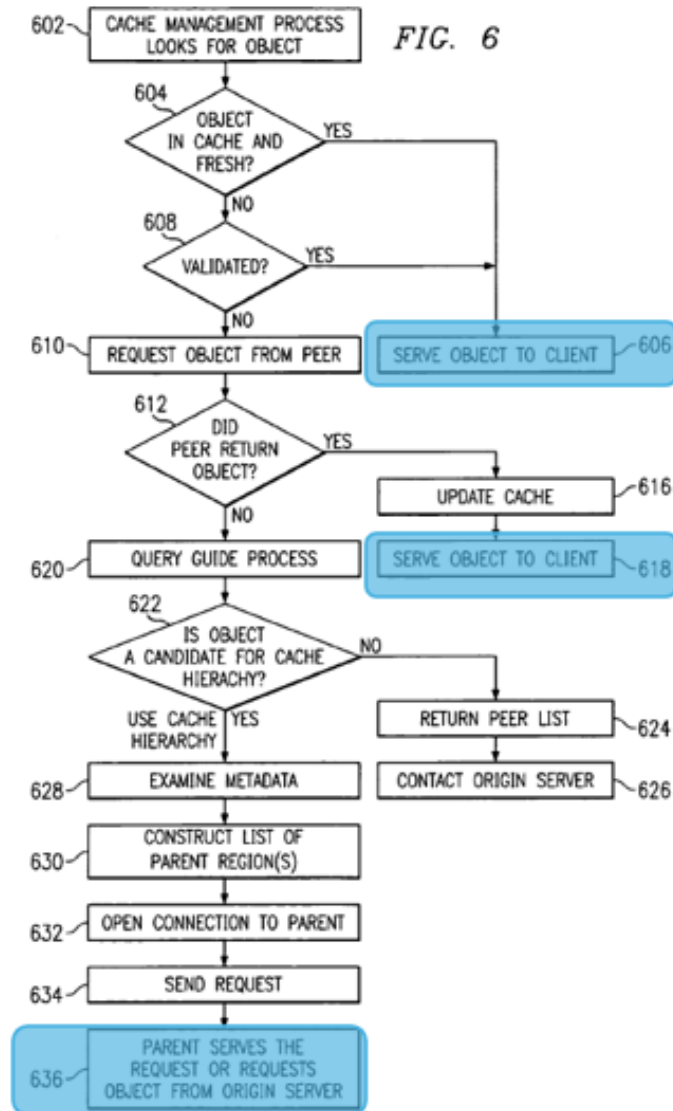
271. The combination of Dilley, Pai, and Wang discloses this limitation.

272. Dilley’s overall purpose is to distribute content to end-users. It uses a tiered cache hierarchy in a CDN to do so.



EX-1003, Fig. 3 (annotated).

273. While Dilley's process begins with an end-user making a request for content, it ends with that requested content being served to the client.



EX-1003, Fig. 6 (annotated) (steps 606 and 618, for example, concluding with “serve object to client”), 8:3-18, 8:53-58.

274. Dilley therefore teaches providing the obtained resource to the requesting device, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

8. Claim 2: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”

275. The combination of Dilley, Pai, and Wang discloses this limitation.

276. Dilley discloses a system that uses a cache hierarchy to store and serve popular objects from the edge, such “that the [origin] site is not overwhelmed with requests for popular content.” EX-1003, 5:16-21. For instance, Dilley’s “[c]ache hierarchy uses a hierarchy of servers to distribute content closer to the requesting end users,” resulting in a “particularly advantageous [functionality] for flash-popular content.” EX-1003, 9:57-63. Thus, a POSITA would have understood that Dilley makes a determination and indicates if a requested resource is not popular, and thus can be served from a higher tier server, such as the origin server.

277. Moreover, Pai discloses specifically that the popularity designation associated with the requested resource indicates that it is not popular. Pai describes a popularity-based media distribution system that distributes popular content from one location and less popular content from another. *See* EX-1004, Summary. To do this, Pai determines popularity values of content and identifies: (1) media content with popularity values above the highest threshold (step 604), (2) media content with

popularity values between lowest and highest threshold values (step 608), and (3) media content with popularity values below a lowest threshold value (step 612). EX-

1004, Fig. 6:

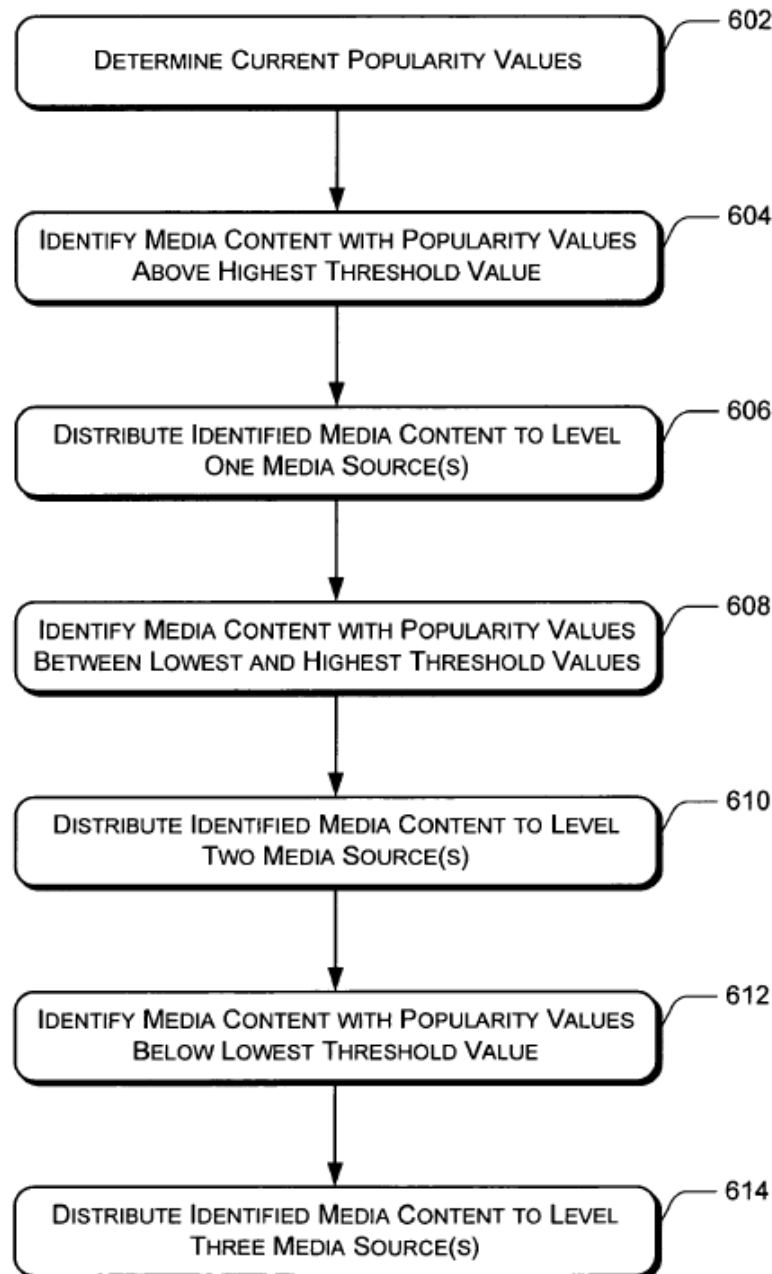


Figure 6

278. The media content identified as less popular will be maintained by a media source located physically further from the requesting client devices. EX-1004, ¶¶[0013], [0016] (“less popular media content may be served from media sources associated with another level” further from requesting devices), [0017], [0020] (“level three media source 216 is configured to maintain and serve the least popular media content”). Pai explains, as an example, that these unpopular items could be those with popularity values below the lowest popularity threshold, such as those assigned a value between one and two (on a scale of one to ten). EX-1004, ¶[0044] (“media content with popularity values below the lowest popularity threshold is identified”).

279. Pai also illustrates distributing of media content based on popularity in Figure 3 below:

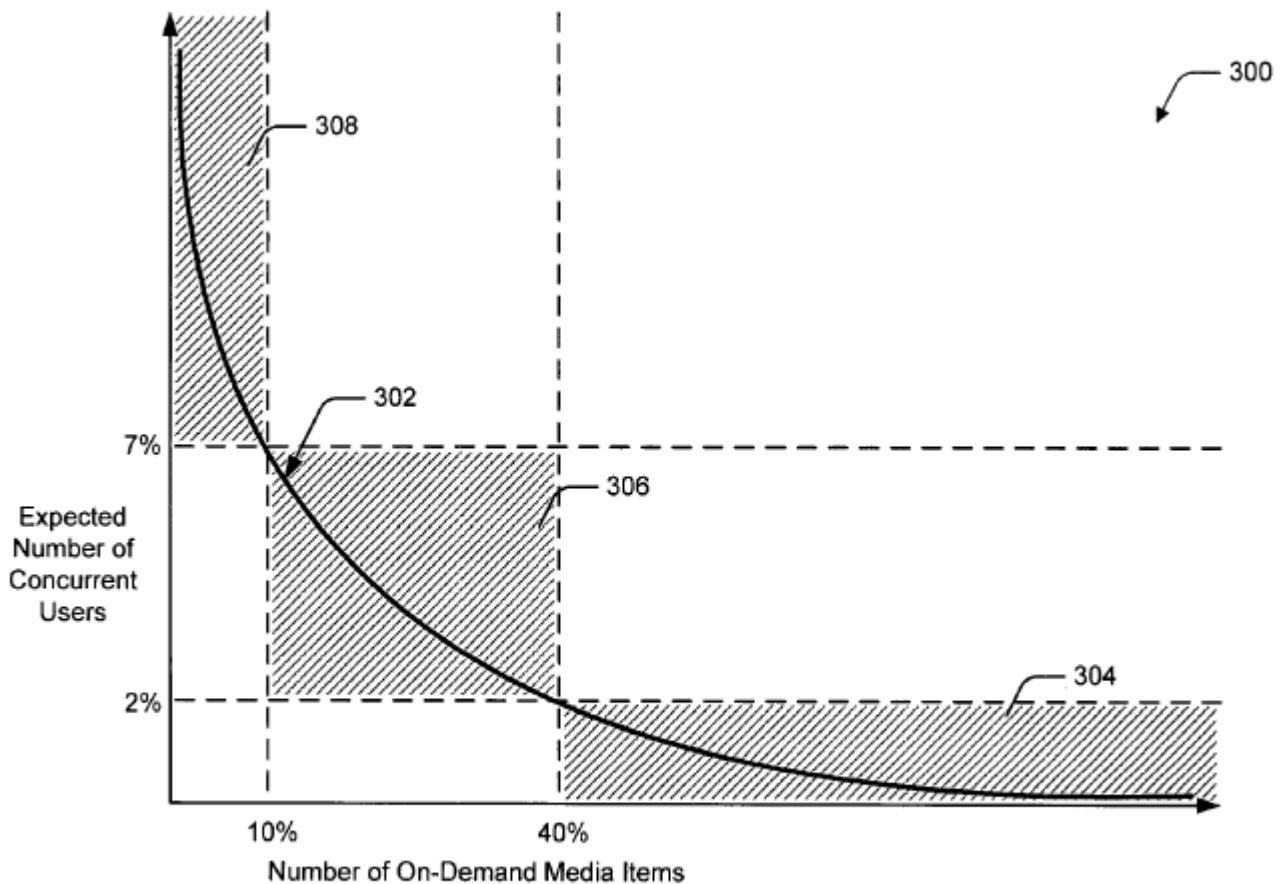


Figure 3

EX-1004, Fig. 3. Pai explains that it can use the determined number of concurrent users of a media item to determine the media item's popularity. EX-1004, ¶¶[0021]-[0023], [0039]. As one example, Pai explains that the shaded area 304 represents the largest number of media items that have the fewest number of concurrent users (i.e., were the least requested content items), and thus "shaded area 304 represents the least popular media content." EX-1004, ¶[0021].

280. Thus, in my opinion, Pai discloses designating that a requested resource is not popular so that it can appropriately distribute that content to the level three media source. A POSITA would have been motivated to implement Pai's popularity service in Dilley with a reasonable expectation of success, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

9. Claim 3

- a. 3(a) "The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular,"**

281. The combination of Dilley, Pai, and Wang discloses this limitation.

282. As explained above, Dilley discloses a system that uses a cache hierarchy to store and serve popular objects from the edge, such "that the [origin] site is not overwhelmed with requests for popular content." EX-1003, 5:16-21. For instance, Dilley explains that the "[c]ache hierarchy uses a hierarchy of servers to distribute content closer to the requesting end users," resulting in a "particularly advantageous [functionality] for flash-popular content." EX-1003, 9:57-63. Thus, a POSITA would have understood that Dilley makes a determination and indicates whether a requested resource is popular, and thus can be served from the edge servers, protecting the origin server from these requests for popular content. EX-1003, 5:65-6:7.

283. Moreover, Pai further discloses specifically that the popularity designation associated with the requested resource indicates that it is popular. *See, e.g.*, EX-1004, ¶[0013]. As further explained above, Pai not only identifies popularity values of content below a lowest threshold value (least popular content), but it also identifies media content with popularity values above a highest threshold value (indicating a popular resource). EX-1004, ¶¶[0040]-[0041] (“media with popularity values above a highest popularity threshold are identified” and distributed to level one media sources); *see also id.*, Fig. 3, ¶[0023] (identifying popular items at shaded area 308), Fig. 6 (identifying that a resource is popular at step 604 as shown above).

284. Thus, Pai specifically determines current popularity values of content as it is requested by users. EX-1004, ¶¶[0039], [0047]. Pai increments these popularity values when they are requested from users and uses these ever-changing popularity values to determine whether the content is popular (having a “popularity value[] above highest threshold value”), is unpopular (having a “popularity value[] below lowest threshold value”), or is in the middle (having a “popularity value[] between lowest and highest threshold values”). EX-1004, Fig. 6, ¶¶[0038]-[0045].

285. For instance, Pai discloses that “level two media sources 220 and 224 are configured to maintain and serve more popular media content,” while “level one

media sources 218 and 222 are configured to maintain and serve the most popular media content.” EX-1004, ¶[0020].

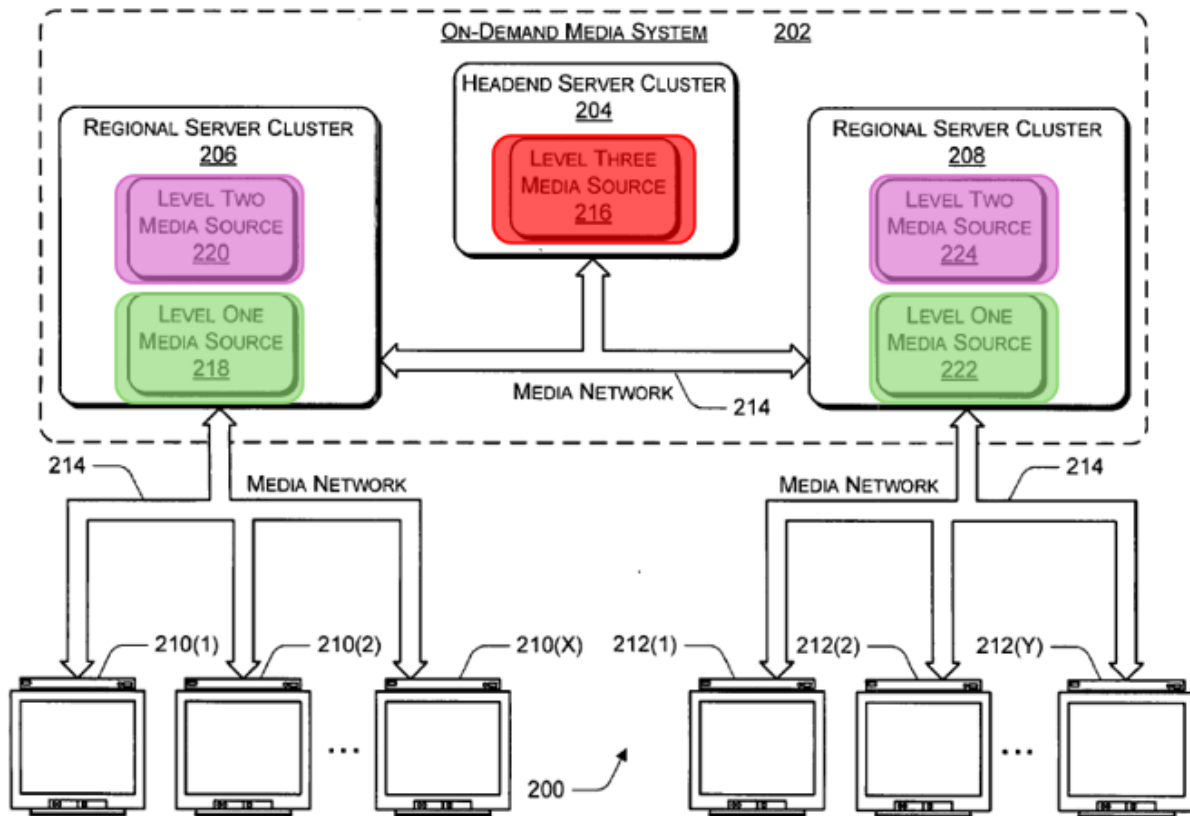


Figure 2

EX-1004, Fig. 2 (annotated).

286. In order to ensure that content is distributed to, maintained, and served from the appropriate media source or server, as Pai explains, a POSITA would have understood that it needs to determine that certain media content is designated as popular content. That way, the content can be dynamically distributed across the system, as Pai intends. EX-1004, ¶[0013].

287. Thus, in my opinion, Pai discloses designating that a requested resource is popular so that it can appropriately distribute that content to the level one media source. POSITA would have been motivated to implement Pai's popularity service teachings with Dilley with a reasonable expectation of success, and combining Dilley with Pai and Wang, as discussed above, would not alter these teachings.

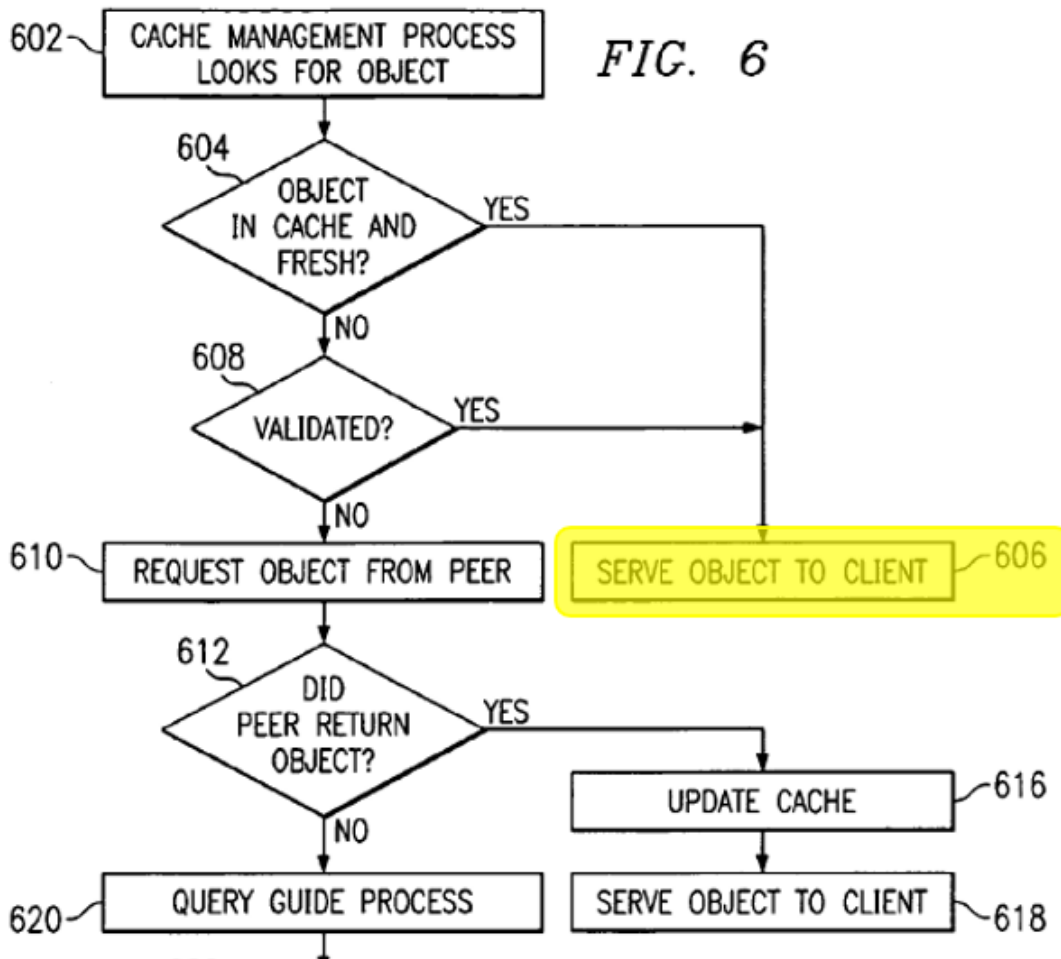
- b. **3(b) “the method further comprising: receiving an instruction to cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.”**

288. Dilley in combination with Pai and Wang discloses receiving an instruction to cache the resource at the first server of a first tier of servers when a resource is obtained from the content server of the content delivery network.

289. As stated above, however, based on this claim's dependency, this claim limitation appears to contradict the claim limitations from which it depends. For instance, claim 1 recites “receiving an instruction *to not cache* the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server.” *See* EX-1001, claim 1 (limitation 1(e)) (emphasis added). Claim 3, which depends from claim 1, now recites “receiving an instruction *to cache* the resource at the first server of the first tier of servers when the resource is obtained from the content server.” *See* EX-1001, claim 3 (emphasis added). Thus, I understand that claim 3 creates an internal inconsistency by

simultaneously requiring an instruction to *not cache* and an instruction to *cache* the same resource at the same server when the resource is obtained from the content server. However, for purposes of this proceeding, I assume this limitation can be reconciled with claim 1.

290. Because Dilley uses a cache hierarchy to store and serve popular objects from the edge, a POSITA would have understood that the servers in Dilley's hierarchy send and receive instructions to cache popular content at the edge for future requests. EX-1003, 5:16-21, 9:57-63. For instance, Dilley explains that when a request is received at an edge server, it looks for the object in the edge server cache (step 604). EX-1003, 8:3-11. If it is, it will serve the object to the client. *Id.* Otherwise, it will request the object from a different server (step 610), and then update the cache to include the requested content. EX-1003, 8:11-21.



EX-1003, Fig. 6 (annotated and abbreviated).

291. A POSITA would have understood that Dilley's system thus discloses receiving an instruction to cache items since it will update the cache with the content it did not previously have. A POSITA would have further understood that this would occur not only when content is received from peer servers, but also when popular content is retrieved from parent or origin servers, since that would in turn serve

Dilley's stated goal of serving popular content from the edge and insulating the origin server from requests for such popular materials. EX-1003, 1:8-10, 2:26-31.

292. Pai further confirms this understanding and provides additional motivation for a POSITA to include an instruction to cache popular resources in the Dilley system as claimed. It discloses that its system will “dynamically redistribute media content” among different levels “based on popularity.” EX-1004, ¶[0053]. As Pai explains, a “[m]edia delivery engine 424 is configured to manage delivery of media content to client devices based on received requests for the media content,” (i.e., the number of requests for a given resource is used “to update popularity values associated with the media content, which may then be used to redistribute the media content across multiple levels of media sources”). EX-1004, ¶[0030].

293. Thus, Pai discloses “a server may, based on popularity, dynamically redistribute media content among multiple levels of media sources,” so that the system will instruct the content to be cached at, e.g., the first server as it becomes more popular. EX-1004, ¶¶[0029], [0052]-[0054].

294. Accordingly, a POSITA would have understood that if content is retrieved from the content server of Pai's CDN and causes the popularity threshold to increase and deem the content “popular,” that the system will send an instruction to redistribute that content at a different media source, like the first server, based on

its new popularity score. EX-1004, ¶[0029]. Thus, the content will be cached at the edge for future requests.

295. Wang similarly includes content-specific attributes, including an “indication of cacheable or not.” EX-1005, ¶[0123]. Thus, this includes not only an instruction not to cache content, but also an instruction to cache certain content. For instance, “[i]f a cacheable content is indicated,” such as a popular content, “cache the content if ‘cacheable’ is also specified in the http response from the origin server.” EX-1005, ¶[0124].

296. A POSITA would understand that this cacheable attribute attached to the content’s metadata would be an instruction for the level 1 server to cache the resource at level 1 when it is popular and obtained from the level 3 server (i.e., the origin server).

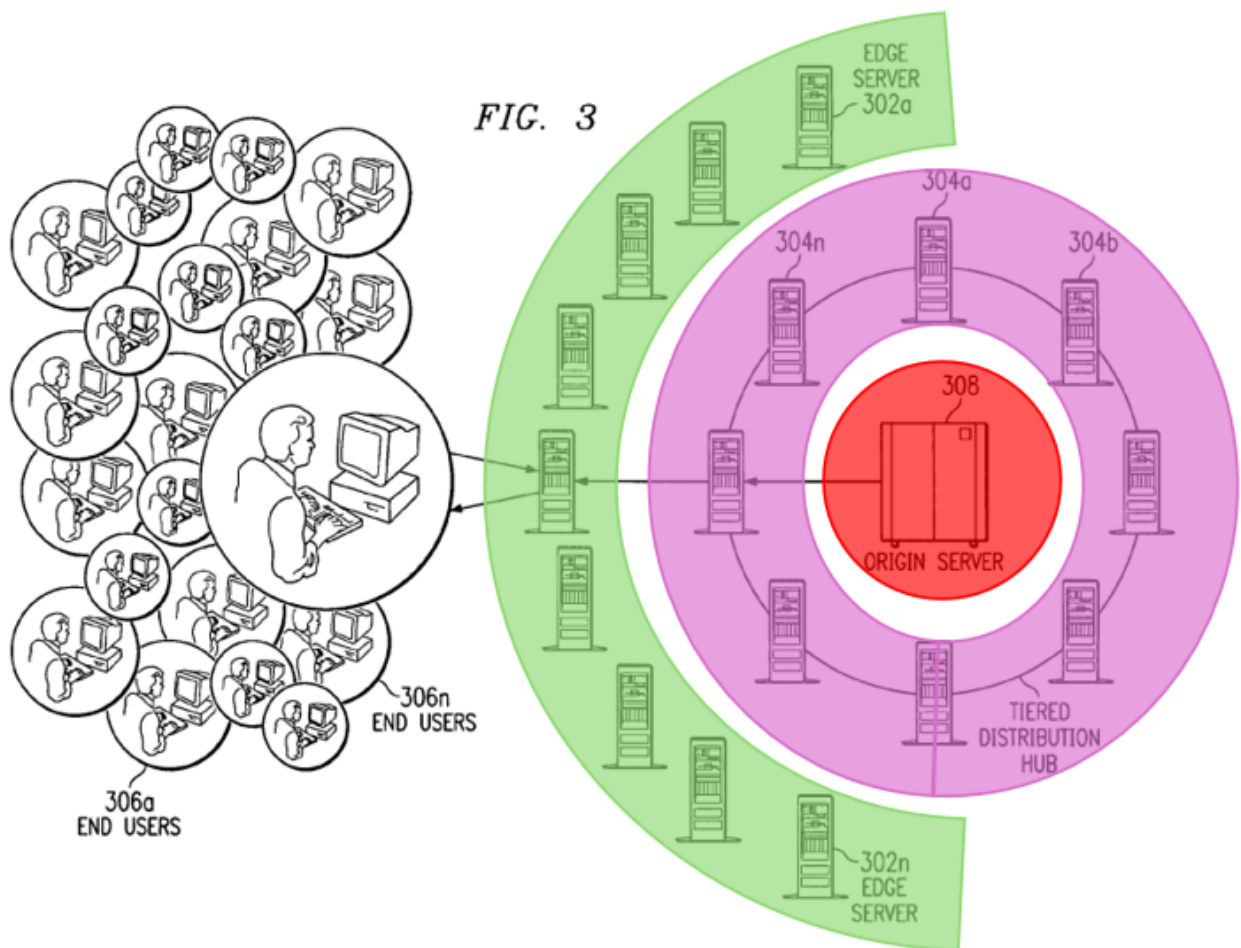
297. Similar to the above, a POSITA would have understood that this cacheable instruction would help ensure that the popularly requested content is cached at the lower levels.

298. The combined teachings of Dilley, Pai, and Wang, accordingly, disclose this limitation. A POSITA would have been motivated to implement these teachings to serve Dilley’s goals, (EX-1003, 1:8-10, 2:26-31), and to achieve the benefits described in Pai, (EX-1004, ¶¶[0013], [0016]-[0017]), and Wang (EX-1005, ¶[0123]); *see also* §VII.B.5.

10. Claim 4: “The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.”

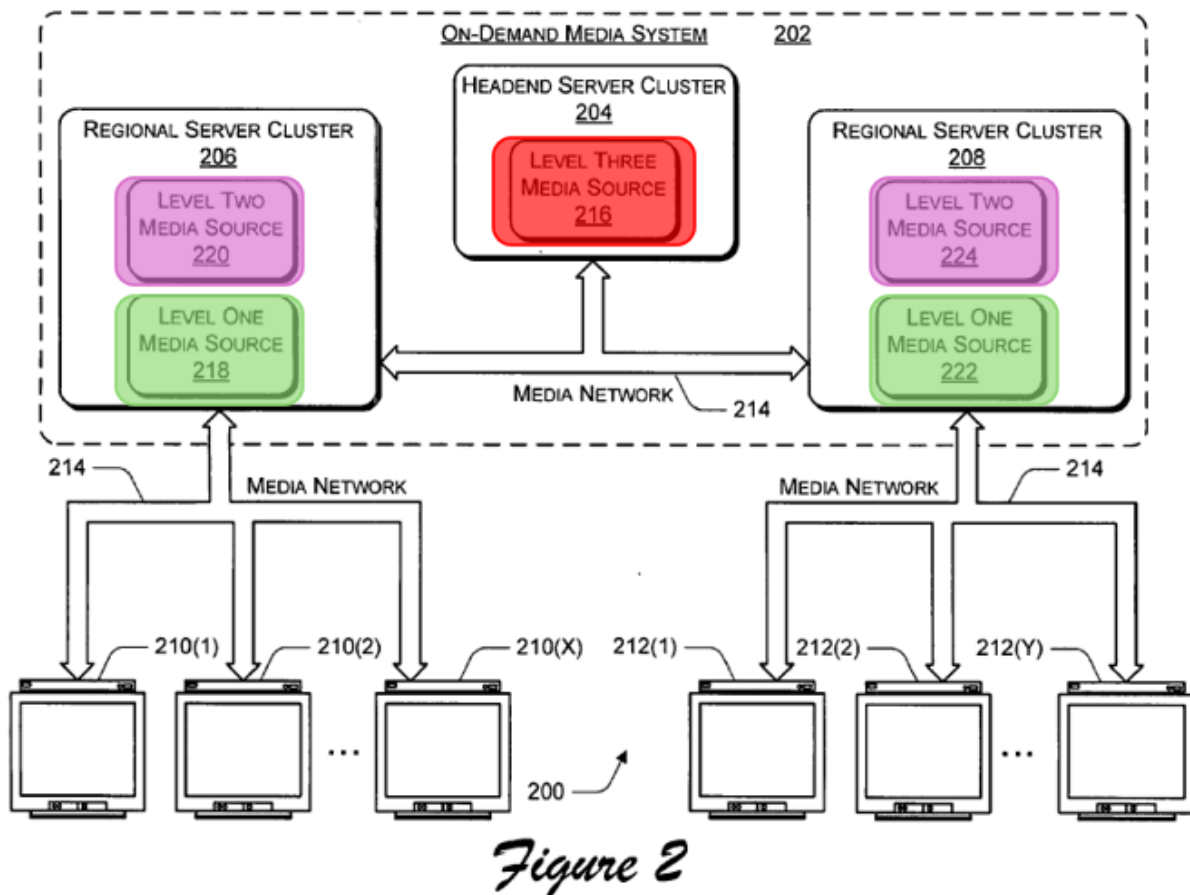
299. The combination of Dilley, Pai, and Wang discloses this limitation.

300. Dilley explains that the second server could be a parent server. *See* EX-1003, Fig. 4. A POSITA would have understood that a parent server, below in purple, is a mid-tier server as it sits in a middle tier between the edge and the origin.



EX-1003, Fig. 3 (annotated).

301. The same is true for Pai's level two media source, which is effectively the parent server, sitting in a middle tier between the level one media source (e.g., edge server) and the level three media source (e.g., origin server). EX-1004, ¶[0020].



EX-1004, Fig. 2 (annotated).

302. A POSITA would have further understood that a metro mid-tier server is a commonly used type of server that sits in a metropolis and is thus easily accessible by a large number of end-users. EX-1020, 2 (describing a three-tier hierarchical architecture with “initial media sources (original content providers),

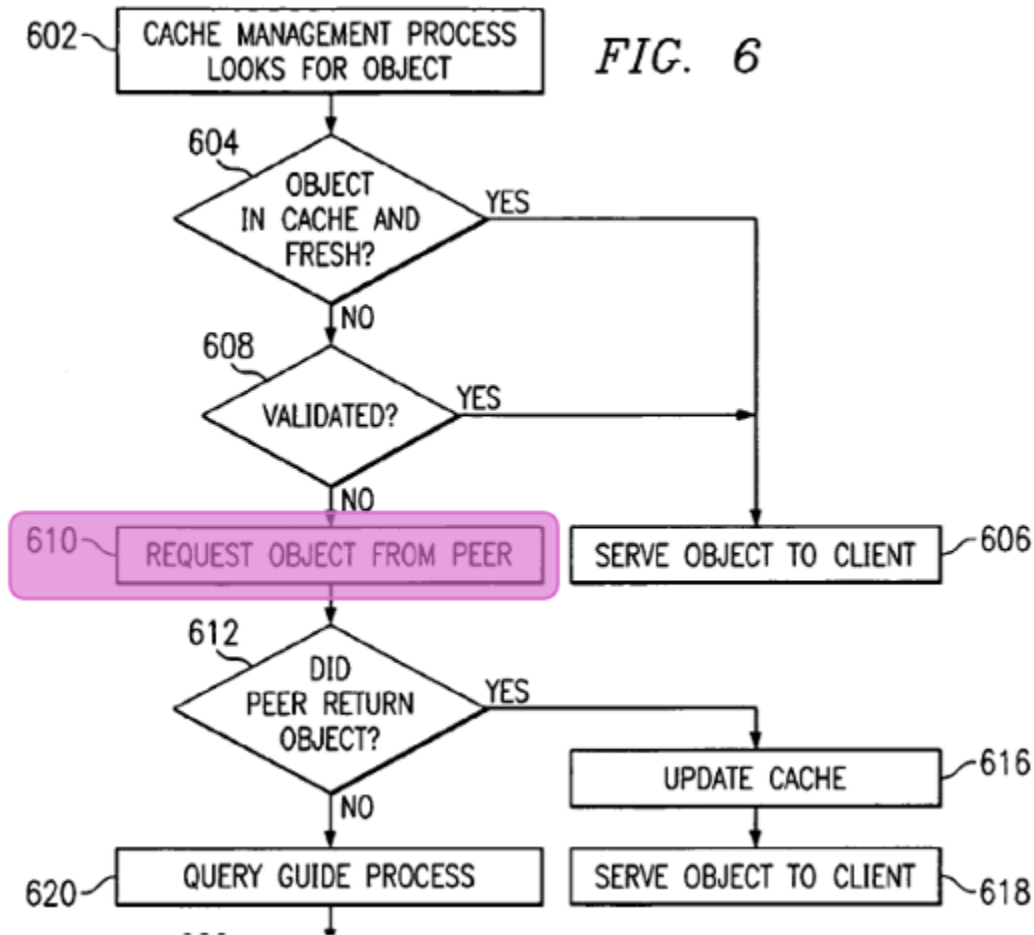
metropolitan media centers and regional media centers acting as hierarchical carriers for popular multimedia content”). As the Multimedia Delivery Grid: A Novel Multimedia Delivery Scheme explains, “[t]he function of metropolitan media centers is to aggregate a number of neighbored regional media centers, for instance of a metropolitan area.” *Id.* (further explaining that, on request, “content is directly provided from the regional media center to the consumer” if it is available; “[i]f the content cannot be obtained from there, the regional media center asks its associated metropolitan media center;” otherwise, if the content cannot be obtained from the metropolitan media centers, “content is obtained from the original content provider at the top of the distribution hierarchy”). A metro server is a logical choice (and is one of a finite number of choices) for this system, as it would serve the greatest number of end-users (located in crowded or compact metropolis areas). By being geographically close to a large number of end-users, the system can leverage the metro servers to reduce latency and efficiently and quickly serve requests from users located in the metropolis area. EX-1003, 6:7-10; EX-1004, ¶¶[0017], [0020].

303. In view of the teachings of Pai and Wang, a POSITA would have been motivated to implement Dilley’s parent servers as metro mid-tier servers. A POSITA would have recognized such servers as one of a finite set of options, and a POSITA would have recognized their appropriateness for Dilley’s system in view of Pai and Wang. *See supra* §§VII.B.5-6.

11. Claim 5: “The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.”

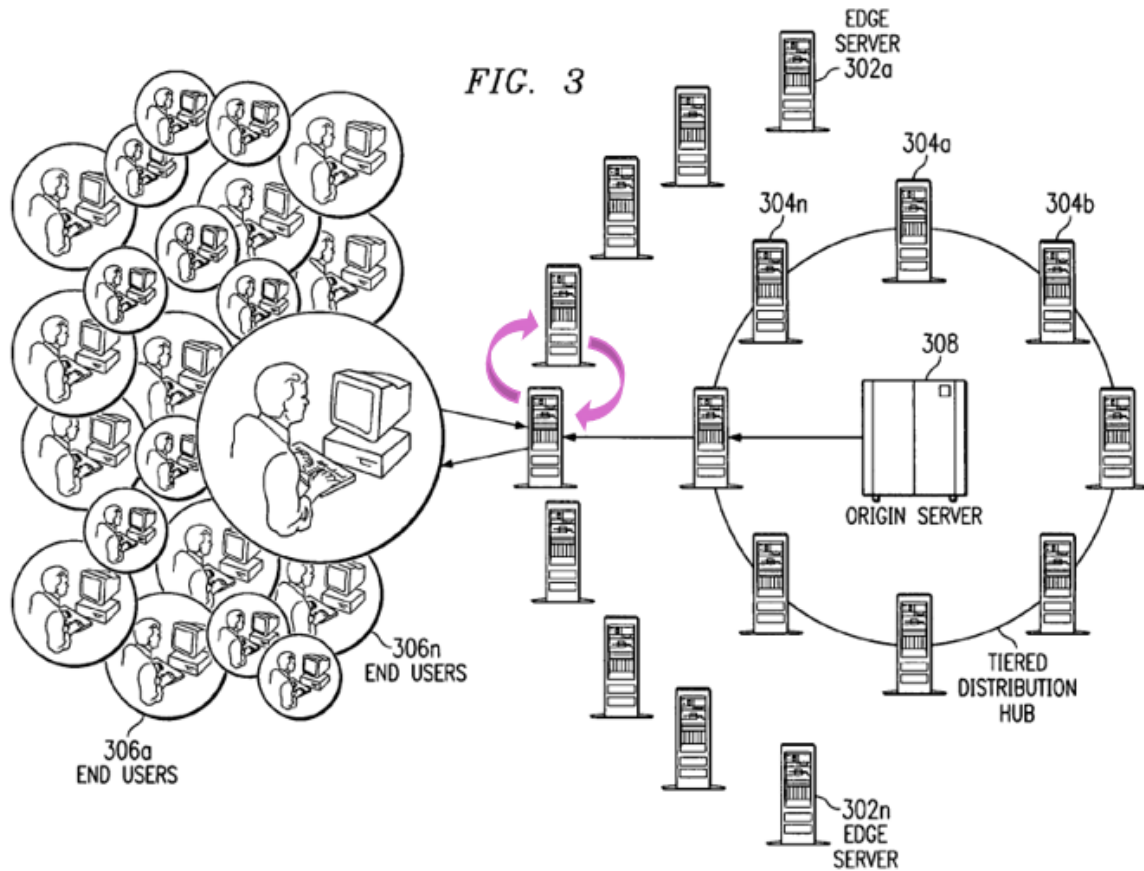
304. The combination of Dilley, Pai, and Wang discloses this limitation.

305. As explained above for limitation 1(c), Dilley discloses, in at least one example, that the second server is a peer server of the first tier of servers of the CDN. *Supra* §VII.B.7.d. Dilley explains that when a request is received at an edge server, the server first looks for the object in the edge server cache (step 604). EX-1003, 8:5-11. If it is, it will serve the object to the client. *Id.* Otherwise, it will request the object from a different server (step 610), and then update the cache to include the requested content. EX-1003, 8:11-18. For instance, Dilley explains that “[i]f the object is not in cache or needs to be validated, the edge server makes a request (e.g., via a cache sharing protocol) to other edge servers in the region (each a ‘peer’) to see if one of them has a fresh copy of the object.” EX-1003, 8:3-15.



EX-1003, Fig. 6 (abbreviated and annotated).

306. Thus, Dilley discloses that the first server (i.e., the edge server receiving the request from the end-user) will request that resource from a second server (i.e., a peer server of the same tier of servers) if the first edge server does not have the requested content in its cache. EX-1003, 5:22-30, 8:11-18.



EX-1003, Fig. 3 (annotated to identify communication among peer servers).

307. A POSITA would have further found it obvious for the second server to be a peer server of the first tier of servers because, upon a cache miss at the first server, requesting the content from a peer server in the same tier would reduce any latency, ensuring quicker and more reliable content delivery, and would have further insulated the parent and origin servers from a large volume of requests (in furtherance of the '573 patent's and Dilley's goals). EX-1003, 1:65-2:3, 6:4-10.

308. Further, as explained above in limitation 1(d), the second server (i.e., the peer server) would send the redirect instruction to the first server when the requested resource is not available at the second server.

12. Claim 6: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”

309. The combination of Dilley, Pai, and Wang discloses this limitation for at least the same reasons provided for claim 3. *See supra* §§VII.B.5, VII.B.9. For instance, Pai discloses a three levels of popularity values and will distribute and redistribute media across its system based on whether the popularity value of a given content falls below the lowest threshold, exceeds the lowest threshold (but fails to exceed the highest threshold), or exceeds the highest threshold. EX-1004, Fig. 6 (steps 604 through 612).

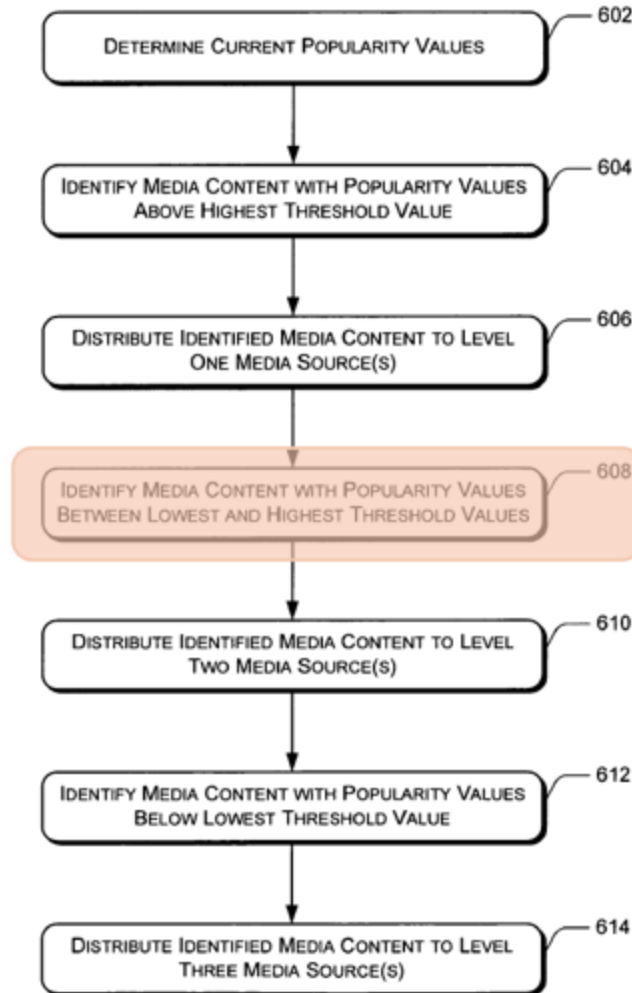


Figure 6

EX-1004, Fig. 6 (annotated).

310. Specifically, Pai explains that it will “identify media content with popularity values between lowest and highest threshold values” at step 608, shown above. EX-1004, ¶[0042]. A POSITA would have understood that this comprises an indication whether the current popularity value for a resource exceeds a first predetermined popularity threshold (i.e., exceeds the lowest threshold value) and is

thus prevented from being classified as the least popular content. EX-1004, Figs. 5-6.

311. Such content in Pai's system would be dynamically redistributed to the second level of media source, in-between the level one media source for the most popular content and level three media source for the least popular content. EX-1004, ¶¶[0013], [0020], [0038]-[0045]. A POSITA would have been motivated to implement Pai's teachings, as described above, with a reasonable expectation of success, and combining Pai with Dilley and Wang would not alter these teachings.

13. Claim 7: "The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network."

312. The combination of Dilley, Pai, and Wang teaches that the popularity service is associated with the second server of the CDN because a POSITA would have understood that the popularity service could be associated with any server in the CDN. As Pai explains, its system is comprised with several servers, each having many components. For instance, Pai explains that its on-demand media server 400 includes exemplary components such as the media distribution module 414, which includes the media scoring module 416 that "determines popularity scores associated with media content." EX-1005, ¶¶[0025]-[0026]. Thus, each media server 400 can be equipped with a popularity service and, as Pai explains, "on-demand media server 400 may represent a server associated with headend server cluster 204 [in Figure 2],

a server associated with one or more regional server clusters 206 and 208 [in Figure 2], or any combination thereof.” EX-1005, ¶[0025]. Thus, a POSITA would have understood that the popularity service could be associated with any edge server, parent server, and/or origin server in the CDN.

313. This is also consistent with the teachings of Dilley. Indeed, Dilley explains that in existing systems, each server can be equipped with a hot object cache “for popular objects being served by the CDN.” EX-1003, 4:47-62, Fig. 2. A POSITA would have understood that the hot object cache acts as a popularity service to determine if content is popular and should remain in the server’s cache.

314. And it is further consistent with the ’573 patent, which explains that those of skill in the art will know and understand “that in a multi-tier CDN [(such as those in Dilley, Pai, and Wang)], the popularity service may be located at any tier, or there may be popularity services at more than one tier.” EX-1001, 8:8-11.

315. Thus, a POSITA would have understood at the time of the ’573 patent that the popularity service could be associated with any of the servers in the CDN. A POSITA would have also recognized that associating the popularity service with the second server would have been one of a finite set of options, and a POSITA would have recognized the appropriateness of that structure for Dilley’s system.

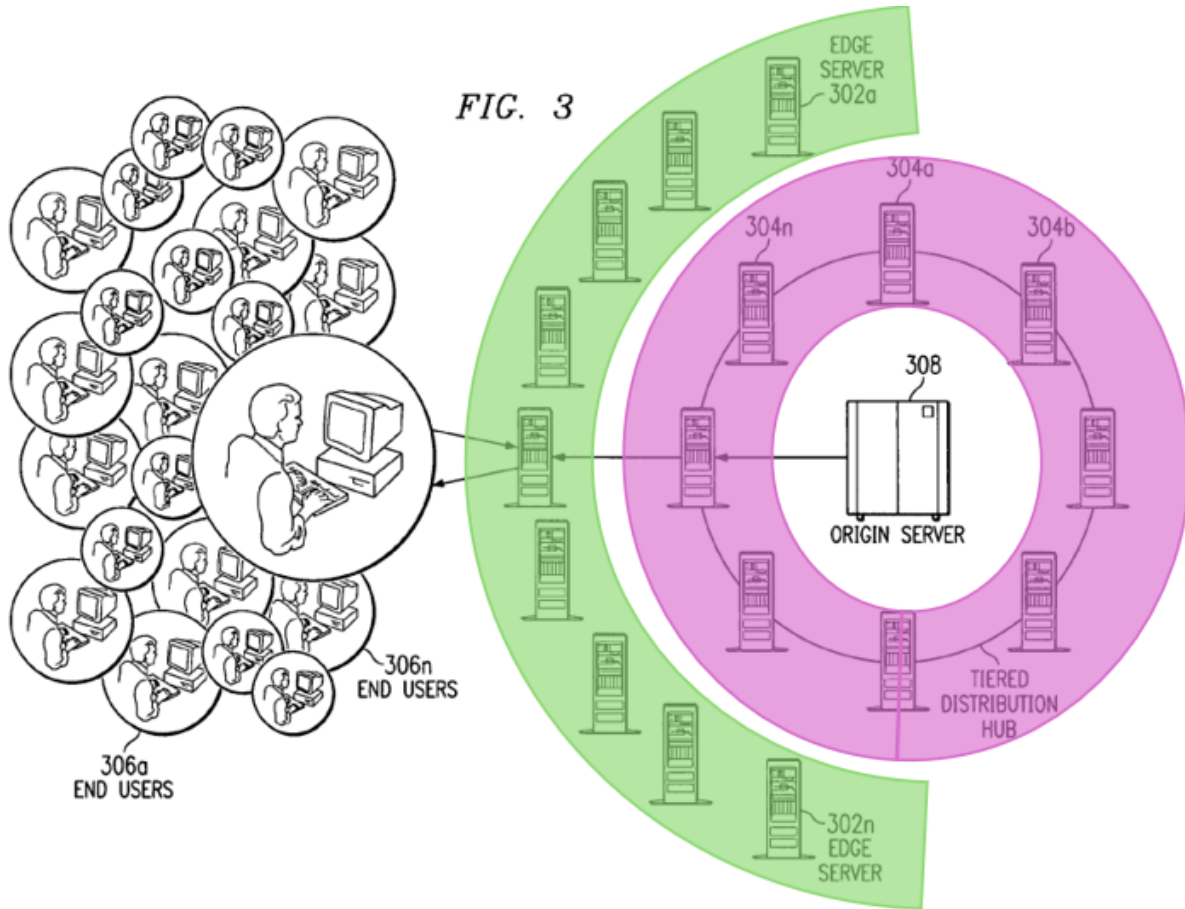
14. Claim 8: “The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.”

316. The combination of Dilley, Pai, and Wang teaches this limitation because, as explained above, *supra* §VII.B.13, a POSITA would have understood that the popularity service can be associated with any or all of the servers of the CDN, including the first server of the first tier of servers. Thus, in my opinion, a POSITA would have found it obvious for the popularity service to be associated with the first server of the first tier of servers.

15. Claim 9: “The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.”

317. The combination of Dilley, Pai, and Wang discloses this limitation.

318. Dilley discloses this limitation for at least the reasons provided for limitation 1(c). For instance, Dilley discloses that the parent server is a second server, which is located in a second tier of servers, distinct from the first tier of servers (i.e., the tier encompassing the edge servers).



EX-1003, Fig. 3 (annotated).

319. Indeed, Dilley employs a tiered distribution system, including, for example, edge servers in one tier, parent servers in a second tier, and an origin server at a third tier. Thus, the combination of Dilley, Pai, and Wang discloses this limitation. Combining Dilley with Pai and Wang as described above would not alter these teachings.

16. Claim 10

- a. 10(a) “The method as recited in claim 3 further comprising: receiving, at the first server of the first tier of servers of the content delivery network, an additional request for the resource available from the content delivery network;”**

320. The combination of Dilley, Pai, and Wang discloses this limitation. because Dilley’s system is meant to be repeated. Indeed, Dilley and Pai seek to identify popular content in order to serve that content from the edge, improving delivery speeds and insulating the origin servers. EX-1003, 6:4-10; EX-1004, ¶[0017]. Thus, a POSITA would have understood that the combined system would receive repeated requests for popular content and would go through the same process for each additional request. EX-1003, 8:5-18.

321. For instance, for an initial request, Dilley identifies whether the edge server has the content cached and whether the content is fresh. EX-1003, 8:3-11. If it is not, it will request the item from another server. EX-1003, 8:11-18. When the first server receives this content item, it updates its cache, caching the content at the first server for future requests. *Id.* Thus, when an additional request is received, Dilley’s first server will have the object in the cache and it will be fresh, thus being able to quickly and reliably serve the item to the end-user. EX-1003, Fig. 6.

322. Combining Dilley, Pai, and Wang as described above would not alter these teachings, and the combined system would accordingly disclose this limitation.

b. 10(b) “determining that the resource is cached at the first server of the first tier of servers; and”

323. The combination of Dilley, Pai, and Wang discloses this limitation because one of the initial steps of Dilley’s process is to determine whether the resource is cached at the first server. EX-1003, 8:3-11, Fig. 6 (at 604). As Dilley explains, if the content is cached and fresh, the first server will serve the content to the end-user. *Id.* It is only if the system determines that the content is not cached or is not fresh that it will request the content from another server of the system. EX-1003, 8:11-18. After updating its cache, *see supra* §VII.B.9.a., the resource would be available at the first server (within the first tier of servers), and Dilley would accordingly determine that the resource was cached there.

324. Combining Dilley, Pai, and Wang as described above would not alter these teachings, and the combination would accordingly disclose this limitation.

c. 10(c) “providing the cached resource from the first server of the first tier of servers in response to the additional request for the resource.”

325. The combination of Dilley, Pai, and Wang discloses this limitation.

326. As Dilley explains, once the item is cached at the first server (i.e., the edge server) it will be available for additional requests, such that when Dilley’s system receives a request from an end-user, it will check if the requested content is in the cache and is fresh. EX-1003, 8:3-11, 8:15-18. If the content had recently been cached at the edge server from a recent request, *see supra* §VII.B.9.a., then it would

be cached and fresh and would be served to the end-user from the edge. *Id.*, Fig. 6 (at 604).

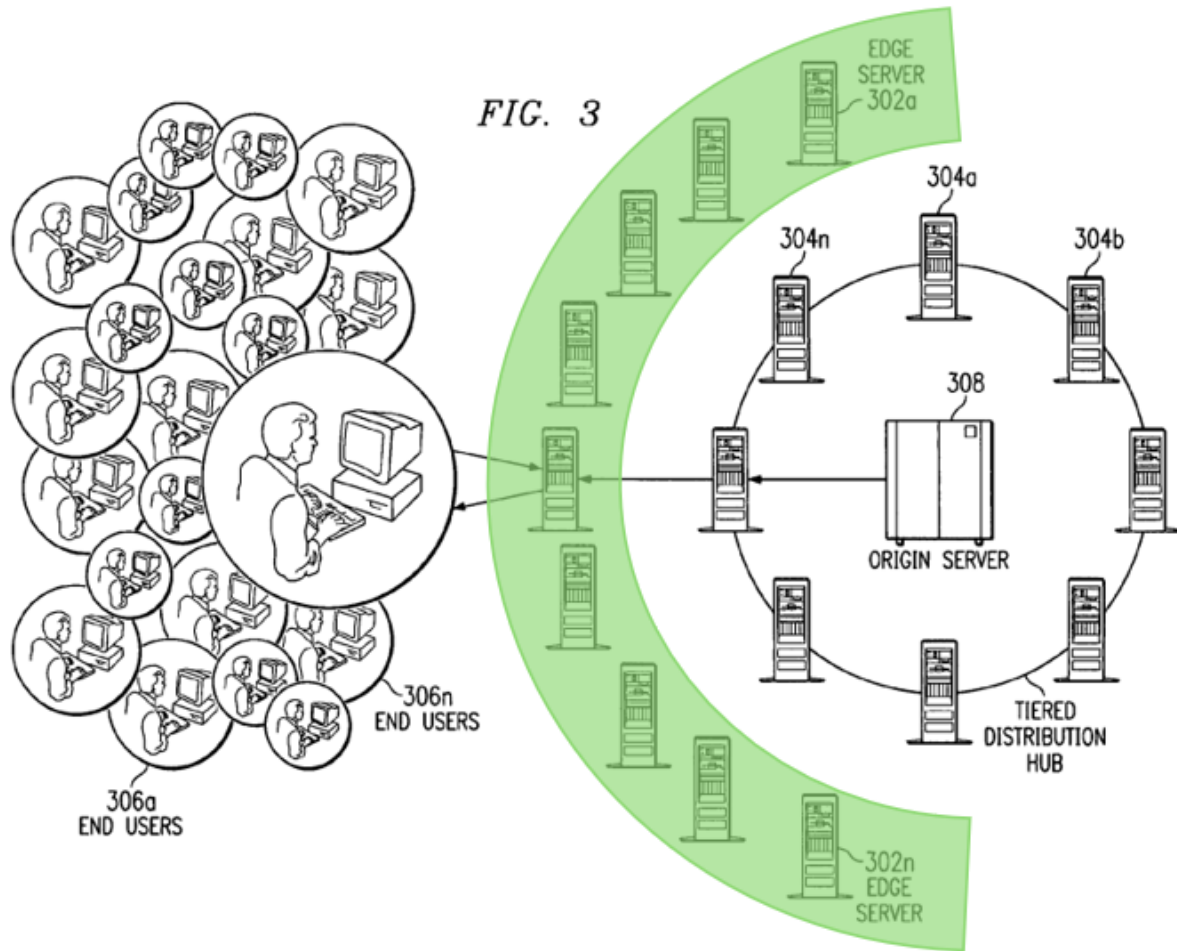
17. Claim 11

a. 11(preamble) “A content delivery network comprising:”

327. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(preamble). *Supra* §VII.B.7.a.

b. 11(a) “a first tier of servers comprising a first plurality of servers, an edge server of the first tier of servers receiving a request from a requesting device in communication with the first server for a resource available from the content delivery network;”

328. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(a). *Supra* §VII.B.7.b. Specifically, Dilley discloses that the first tier of servers comprises a first plurality of servers:

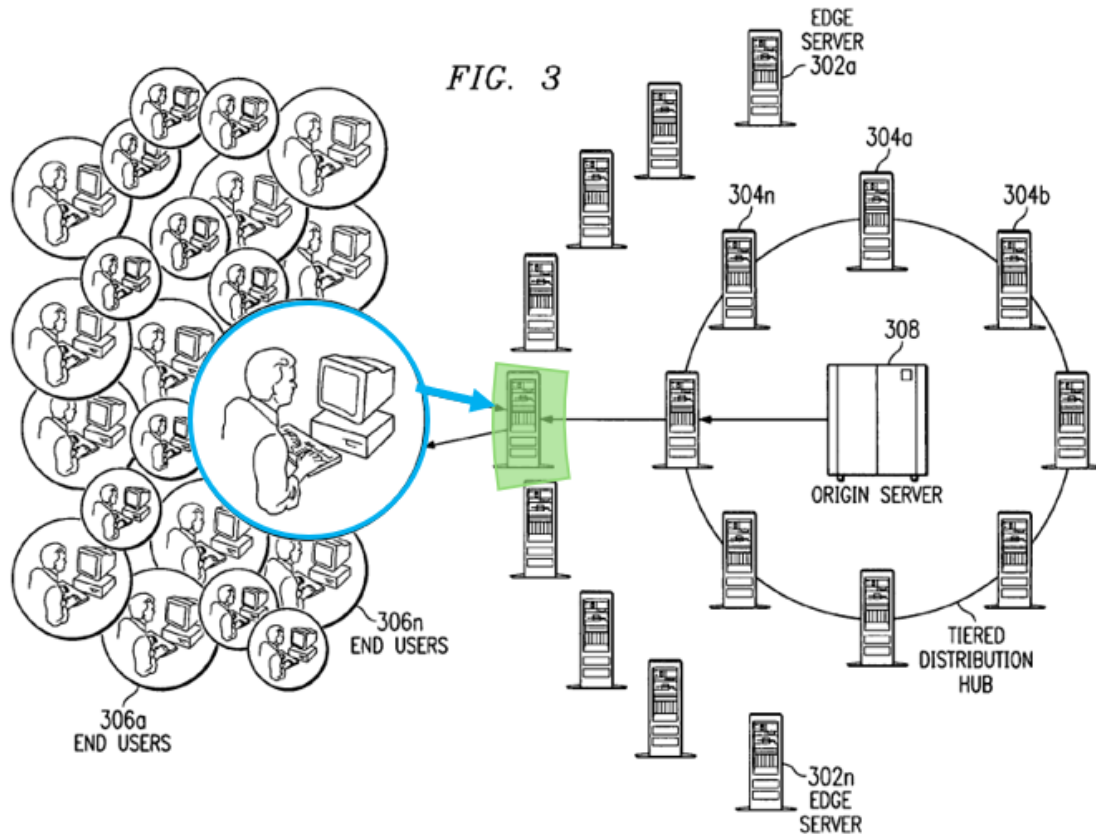


EX-1003, Fig. 3 (annotated).

329. Indeed, Dilley explains that it implements a “tiered distribution” system that uses “two or more tiers: a set of edge servers and a hierarchical set of ‘parent’ or hub regions that service the edge servers.” EX-1003, 4:63-5:1. Dilley discloses that “the CDN service provider operates *a plurality of CDN edge servers 302a-n.*” EX-1003, 5:1-4 (emphasis added), Fig. 3.

330. As further explained above, the first server is an edge server of the first tier of servers, and it is in communication with the requesting device because it is

the edge server that “receives end-user requests for origin server content and serves this content.” EX-1003, 7:50-53.



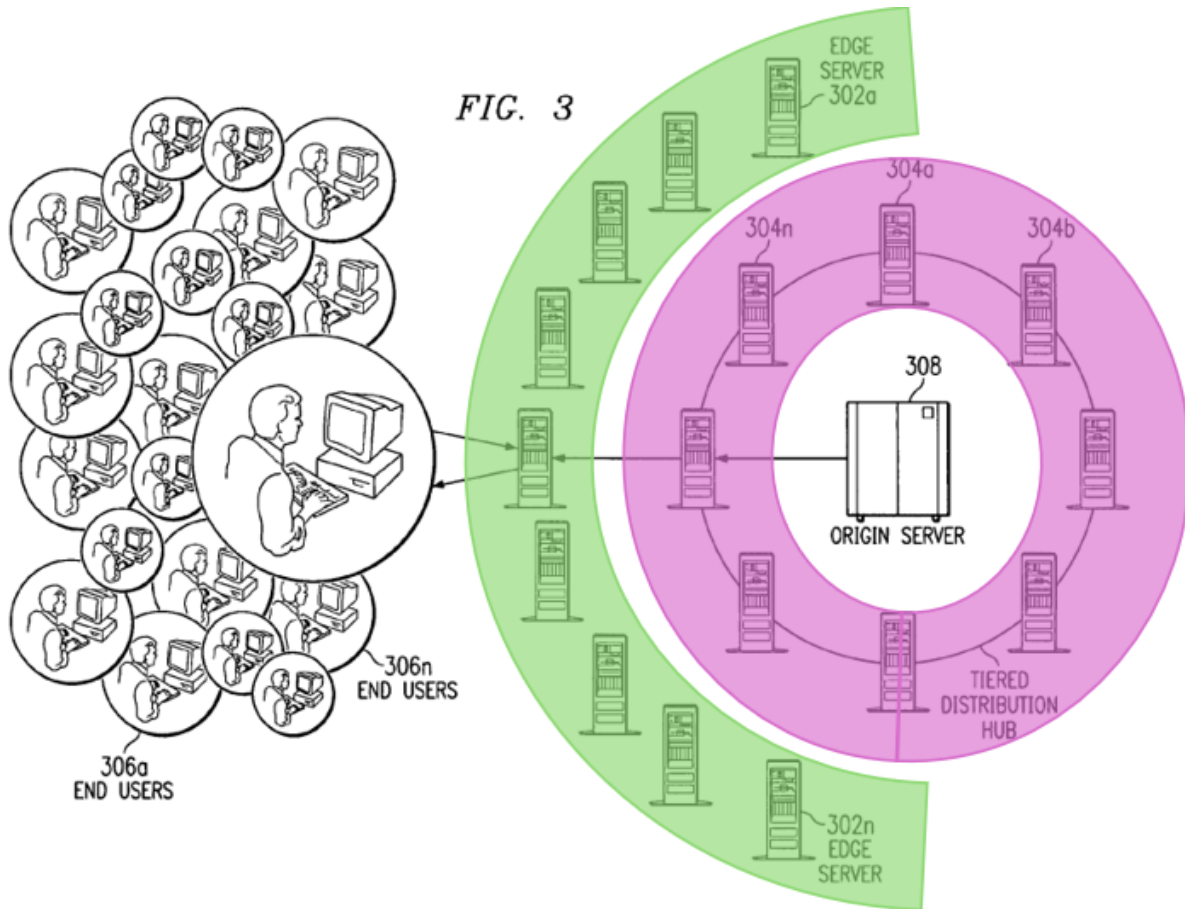
See, e.g., EX-1003, Fig. 3 (showing an edge server receiving the request from an end user).

331. Combining Dilley, Pai, and Wang as described above would not alter these teachings, and the combination would accordingly disclose this limitation.

c. 11b(i) “a second tier of servers comprising a second plurality of servers,”

332. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(c) and claim 9. *Supra* §§VII.B.7.d,

VII.B.15. More specifically, Dilley discloses that the second tier of servers comprises a second plurality of servers:



EX-1003, Fig. 3 (annotated).

333. As explained above, Dilley implements a “tiered distribution” system that uses “two or more tiers: a set of edge servers and a hierarchical set of ‘parent’ or hub regions that service the edge servers.” EX-1003, 4:63-5:1. Dilley further discloses that in the Figure 3 example, “the CDN service provider operates a plurality of CDN edge servers 302a-n and a set of tiered distribution hubs 304a-n.” EX-1003, 5:1-4. These tiered distribution hubs encompass a second plurality of

servers within a second tier of servers and are used to “buffer the origin server 308” and “ensure that the edge servers have the information requested by the end users.” EX-1003, 5:4-10.

334. Combining Dilley, Pai, and Wang as described above would not alter these teachings, and the combination would accordingly disclose this limitation.

- d. **11(b(ii)) “a first server of the second plurality of servers receiving a request from the edge server for the content and, in response, transmitting a redirect instruction intended for the requesting device to the edge server to obtain the resource from a content server and transmitting an instruction to not cache the portion of the resource at the edge server when the portion of the resource is obtained from the content server; and”**

335. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitations 1(c)-1(e). *Supra* §§VII.B.7.d-f. Moreover, a POSITA would have also understood that the instruction to not cache the portion of the resource at the edge server could be transmitted from the first server of the second plurality of servers (i.e., the parent server), and that doing so would have been one of a finite number of design choices. Similar to the above discussion for limitation 1(d), having the parent server transmit the instruction to not cache would similarly be useful to free the origin server for other tasks.

- e. **11(c) “a popularity service tracking a popularity designation associated with the requested resource;”**

336. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(b). *Supra* §VII.B.7.c.

- f. **11(d) “wherein the edge server processes the redirect instruction to obtain the content from the content server and provides the obtained content to the requesting device.”**

337. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitations 1(d)-1(f). *Supra* §§VII.B.7.e-g.

- 18. **Claim 12: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”**

338. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for claim 2. *Supra* §VII.B.8.

19. Claim 13

- a. **13(a) “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular and”**

339. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 3(a). *Supra* §VII.B.9.a.

- b. 13(b) “the first server of the second plurality of servers transmits an instruction to cache the resource at the content server of the first tier of servers when the resource is obtained from the content server.”**

340. Similar to claim 3, this claim limitation appears to contradict the claim limitations from which it depends. For instance, claim 11 recites “transmitting an instruction *to not cache* the portion of the resource at the edge server when the portion of the resource is obtained from the content server.” *See* EX-1001, claim 1 (limitation 11(b(ii))) (emphasis added). Claim 13, which depends from claim 11, now recites “transmits an instruction *to cache* the resource at the content server of the first tier of servers when the resource is obtained from the content server.” *See* EX-1001, claim 13 (emphasis added). Thus, I understand that claim 13 is simultaneously requiring an instruction *to not cache* and an instruction *to cache* the same resource at the same server when the resource is obtained from the content server. However, for purposes of this proceeding, I assume this limitation can be reconciled with claim 11.

341. Accordingly, the combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 3(b). *Supra* §VII.B.9.b.

342. Moreover, a POSITA would have understood that the instruction to cache the portion of the resource at the edge server could be transmitted from the first server of the second plurality of servers (i.e., the parent server), and that doing so would have merely been one of a finite number of design choices. Similar to the

above discussion for limitation 1(d), having the parent server transmit the instruction would similarly be useful to free the origin (content) server for other tasks.

20. Claim 14: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a metro mid-tier server of the content delivery network.”

343. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for claim 4. *Supra* §VII.B.10.

21. Claim 15: “The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a peer server of the first tier of servers of the content delivery network.”

344. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for claim 5. *Supra* §VII.B.11.

22. Claim 16: “The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.”

345. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for claim 6. *Supra* §VII.B.12.

23. Claim 17: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the first server of the second plurality of servers of the content delivery network.”

346. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for claim 7. *Supra* §VII.B.13.

- 24. Claim 18: “The content delivery network as recited in claim 16 wherein the popularity service is associated with the content server of the first tier of servers of the content delivery network.”**

347. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for claim 8. *Supra* §VII.B.14.

25. Claim 19

- a. 19(preamble) “A method of content delivery in a content delivery network comprising:”**

348. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(preamble). *Supra* §VII.B.7.a.

- b. 19(a) “receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;”**

349. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(a). *Supra* §VII.B.7.b.

- c. 19(b) “accessing a popularity service associated with the first server of the first tier of the content delivery network to determine a popularity designation associated with the requested resource;”**

350. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(b) and claim 8. *Supra* §§VII.B.7.c, VII.B.14.

- d. 19(c) “receiving an instruction to not cache the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network;”**

351. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(e). *Supra* §VII.B.7.f.

- e. 19(d) “requesting the resource from a second server of the content delivery network based at least on the popularity designation associated with the requested resource; and”**

352. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(c). *Supra* §VII.B.7.d.

353. Furthermore, Dilley explains that “the servers in the tiered distribution hubs [(i.e., the parent or origin servers)] will provide the edge servers with the requested content” when the “edge servers do not have the appropriate file when it is requested.” EX-1003, 5:6-10. Because the edge servers are used to prevent the origin servers from becoming “overwhelmed with requests for popular content,” a POSITA would have understood that content with a popularity designation, identifying that content as unpopular, would be served from the parent or origin servers, while popular content would be served from the edge. EX-1003, 5:10-21. Thus, based on an unpopular designation, Dilley would retrieve content from the second server.

354. Pai further dynamically manages and distributes content based on popularity values. EX-1004, ¶[0013]. As Pai explains, as content is consumed, its popularity value can change over time, moving from media source to media source. EX-1004, ¶¶[0020], [0026], [0029]. Pai explains that media with popularity values exceeding the highest threshold will be distributed and managed in one tier—media source one—while others would be distributed and managed in other tiers of media sources. EX-1004, ¶¶[0038]-[0045]. Thus, a POSITA would have understood that when Pai’s system receives a request for content that does not exceed the highest popularity threshold, the request would need to be forwarded to a second server (i.e., a server of media level two for mid-tier popular content or a server of media level three for unpopular content).

355. A POSITA would have thus implemented Pai disclosure of dynamically redistributing content based on its popularity into Dilly’s popularity service, thereby distributing the “flash-popular content” at the edge “closer to the requesting end-users,” EX-1003, 9:57-65, the mid-level popular items at the middle tier servers, and the least popular items at the origin furthest from the requesting end-users.

f. 19(e) “providing the obtained resource to the requesting device.”

356. The combination of Dilley, Pai, and Wang discloses this limitation for the same reasons provided for limitation 1(f). *Supra* §VII.B.7.g.

26. Claim 20: “The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.”

357. The combination of Dilley, Pai, and Wang discloses this limitation because Dilley discloses a “tiered infrastructure using two or more tiers: a set of edge servers and a hierarchical set of ‘parent’ or hub regions that service the edge servers.” EX-1003, 4:63-5:10. Thus, a POSITA would have understood that in a two-tiered system, the first tier would encompass edge servers, and the second tier would comprise the higher-level servers, performing the task of the parent and origin servers. Thus, the second server in the two-tiered system would likewise be content servers storing the content and “servic[ing] the edge servers.” EX-1003, 4:64-5:10. Indeed, Dilley explains that the edge servers “include[] a cache for storing content provider content,” and are thus content servers. EX-1003, 7:50-53. The parent and origin server also store content and are similarly content servers. Dilley explains that when an edge server does not have a content, it will reach out to a parent server or content server, which will serve the content from their cache, indicating that they are also content servers. EX-1003, 8:53-58. The two-tiered system would still achieve Dilley’s goals of quickly and reliably serving content while having the edge servers insulate the content server from repeated requests for popular content. EX-1003, 2:12-17, 2:26-31, 5:10-21.

VIII. SECONDARY CONSIDERATIONS

358. I have considered the issue of secondary considerations and am currently unaware of any evidence of secondary considerations that would support a finding of non-obviousness or otherwise impact my opinions herein.

IX. CONCLUSION

359. For the reasons set forth above, it is my opinion that claims 1-20 of the '573 patent are unpatentable in view of the prior art.

360. In signing this declaration, I understand that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

361. I declare that all statements made herein of my knowledge are true, that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Declaration of Todd C. Mowry, Ph.D.
U.S. Patent No. 10,924,573

Dated: April 30, 2025

By: Todd C. Mowry

Todd C. Mowry, Ph.D.