

UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF TEXAS
TYLER DIVISION

BEDROCK COMPUTER TECHNOLOGIES	§	
LLC,	§	
PLAINTIFF	§	
	§	
VS.	§	CIVIL ACTION NO. 6:09-cv-269-LED-JDL
	§	
SOFTLAYER TECHNOLOGIES, INC., et al.,	§	
	§	
DEFENDANTS	§	

**YAHOO!’S RENEWED MOTION FOR JUDGMENT AS A MATTER OF LAW
REGARDING INVALIDITY**

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
LEGAL STANDARD IN THIS CASE	3
ARGUMENT.....	6
I. CLAIM 1 OF THE '120 PATENT IS INVALID.....	6
A. '95 Linux Code Anticipates Claim 1 of the '120 Patent.....	6
1. Kuznetsov Confirmed that the '120 Patent is Invalid	8
2. The '95 Linux Code Discloses Each Element of Claim 1	8
a. '95 Linux Code Discloses the First Element of Claim 1	9
b. '95 Linux Code Discloses the Second Element of Claim 1	10
c. '95 Linux Code Discloses the Third Element of Claim 1	11
d. '95 Linux Code Discloses the Fourth Element of Claim 1	11
3. Bedrock's Conclusory Response to the '95 Linux Code Lacks Merit.....	12
B. The NRL Code Renders Claim 1 Obvious	14
1. The NRL Code Discloses the First Element of Claim 1	17
2. The NRL Code Discloses the Second Element of Claim 1	17
3. The NRL Code Discloses the Third Element of Claim 1	18
4. The NRL Code Discloses the Fourth Element of Claim 1.....	19
C. The '95 Linux Code and NRL Code Were Not Considered by the PTO	20
D. The '495 Patent Invalidates Claim 1.....	20
II. CLAIM 2 OF THE '120 PATENT IS INVALID.....	25
A. Bedrock Does Not Contest Yahoo!'s Invalidating Testimony	26
B. The Apple Patent and/or the '663 Patent Combined with the Claim 1 Prior Art Render Claim 2 Obvious	27

1.	'95 Linux Code in Combination with Apple Patent Renders Claim 2 Obvious	27
2.	NRL Code in Combination with Apple Patent or '663 Patent Renders Claim 2 Obvious	28
3.	The '495 Patent in Combination with External Chaining Prior Art and Apple Patent or the '663 Patent Render Claim 2 Obvious	28
III.	SECONDARY CONSIDERATIONS CANNOT SAVE THE '120 PATENT	29
	CONCLUSION.....	30

TABLE OF AUTHORITIES

Cases	Page
<i>AdvanceMe Inc. v. RapidPay</i> , 509 F. Supp. 2d 593 (E.D. Tex. 2007).....	4, 14, 30
<i>Black v. Ce Soir Lingerie Co.</i> , 2008 WL 3852722 (E.D. Tex. 2008).....	23
<i>Blackboard, Inc. v. Desire 2 Learn, Inc.</i> , 574 F.3d 1371 (Fed. Cir. 2009).....	4
<i>Burlington Truck Lines, Inc. v. United States</i> , 371 U.S. 156 (1962).....	5
<i>Citizens to Pres. Overton Park v. Volpe</i> , 401 U.S. 402 (1971).....	5
<i>Davis v. McDowell</i> , 596 F.3d 1355 (Fed. Cir. 2010).....	12, 17, 21, 25
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966).....	15, 29
<i>i4i Ltd. P'ship. v. Microsoft Corp.</i> , 598 F.3d 831 (Fed. Cir. 2010).....	4
<i>Iovate Health Scis., Inc. v. Bio-Engineered Supplements and Nutrition, Inc.</i> , 586 F.3d 1376 (Fed. Cir. 2009).....	4, 6
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	14, 15, 23
<i>Lear, Inc. v. Adkins</i> , 395 U.S. 653 (1969).....	5
<i>Mahurkar v. C. R. Bard, Inc.</i> , 79 F.3d 1572 (Fed. Cir. 1996).....	4
<i>Motor Vehicle Mfrs. Ass'n v. State Farm Mut. Auto. Ins. Co.</i> , 463 U.S. 29, 52 (1983).....	4, 5
<i>Orion IP LLC v. Hyundai Motor Am.</i> , 605 F.3d 967 (Fed. Cir. 2010).....	4

<i>Paid Search Engine Tools, LLC v. Yahoo! Inc.</i> , 746 F. Supp. 2d 808 (E.D. Tex. 2010).....	5
<i>Schering Corp. v. Geneva Pharms., Inc.</i> , 339 F.3d 1373 (Fed. Cir. 2003).....	6
<i>Standard Oil Co. v. Am. Cyanamid</i> , 774 F.2d 448 (Fed. Cir. 1985).....	23
<i>Tokai Corp. v. Easton Enters., Inc.</i> , --- F.3d ---, 2011 WL 308370 (Fed. Cir. Jan. 31, 2011).....	29, 30

Statutes

35 U.S.C. § 102(a)	5, 30
35 U.S.C. § 102(b)	5, 30
35 U.S.C. § 102(g)(2)	5, 30
35 U.S.C. § 103(a)	5, 14, 30

Rules

Fed. R. Civ. P. 50(a)	4
Fed. R. Civ. P. 50(b)	1, 4, 5
Patent Rule 98.....	20, 22

INTRODUCTION

At trial, the overwhelming evidence established that each element in claims 1 and 2 of U.S. Patent No. 5,893,120 ('120 patent) was well-known in the art long before the alleged invention. Because the jury did not reach the counterclaim of invalidity, this Court is now the first to consider and decide all evidence of invalidity – including all material prior art and all pertinent information not presented to the PTO. Here, the numerous prior art references not only teach the claims of the '120 patent, they also show that the claimed invention would be anticipated and obvious to one of ordinary skill in the art. Indeed, the inventor admitted he did not invent the techniques of the '120 patent; instead, his invention was simply a combination of known elements. As such, Yahoo! renews its Motion for Judgment as a Matter of Law on Invalidity pursuant to Rule 50(b).

Specifically, the prior art presented at trial included: (1) Linux source code versions 1.3.51, 1.3.52, and/or 2.0.1 from 1995-1996¹ ('95 Linux Code) (Ex. 1); (2) the Naval Research Labs key management computer source code (NRL code) (Ex. 2); (3) U.S. Patent No. 5,121,495 to Nemes ('495 patent) (Ex. 3, Ex. 4); (4) U.S. Patent 6,119,214 to Apple (Apple patent) (Ex. 5); (5) U.S. Patent No. 4,996,663 to Nemes ('663 patent) (Ex. 6); (6) "The Art of Computer Programming," Volume 3, "Searching and Sorting," by D. E. Knuth (Knuth) (Ex. 7); (7) Robert L. Kruse, Data Structures and Program Design (Kruse) (Ex. 8); and (8) Daniel F. Stubbs and Neil W. Webre, Data Structures with Abstract Data Types and Pascal (Stubbs) (Ex. 9). Importantly, neither the '95 Linux Code nor the NRL code were before the PTO examiner in the original patent prosecution or the subsequent re-examination.² As such, this Court owes no deference to

¹ These Linux versions were referred to collectively or individually throughout trial as '95 Linux Code, Kuznetsov Code, Old Linux Code, or Prior Art Linux Code.

² On April 28, 2011, the day after Yahoo! advised the jury that Bedrock never had submitted the NRL code to the PTO, Bedrock finally submitted a copy to the PTO in the second re-examination proceeding.

the PTO in this proceeding.

Through Yahoo!'s fact witness, expert witnesses, third-party witnesses, and numerous exhibits, the evidence overwhelmingly invalidates claims 1 and 2 of the '120 patent. In particular:

- David Filo (Filo), Yahoo!'s co-founder and an expert in computer programming, testified and explained how the '95 Linux Code and NRL code invalidated the '120 patent. (T.T. 4/29/11 a.m. at 128-141).³
- Joel Williams (Williams), Yahoo!'s invalidity expert with over 38 years of experience designing computer systems and 12 years of Linux experience, testified in detail as to why each of the prior art references invalidated each element of claims 1 and 2 of the '120 patent. (T.T. 5/9/11 at 186-206).
- Alexey Kuznetsov (Kuznetsov), author of the *route.c* source code in the '95 Linux Code, testified that his source code did what was claimed in the '120 patent and invalidated the '120 patent. (T.T. 4/29/11 a.m. at 12-15; 4/29/11 p.m. at 146-152).
- Dan McDonald (McDonald), author of the NRL code, testified that it would have been a combination of familiar forms and techniques and obvious to one of ordinary skill in the art to combine the two portions of *key.c* in the NRL code that contained a hash table using external chaining and on the fly garbage collection using linked lists. (T.T. 5/9/11 at 141:2-142:24).
- Nicholas Godici (Godici), a former commissioner of Patents at the United States Patent and Trademark Office ("PTO") testified at trial that there was no evidence that the PTO considered the '95 Linux Code or NRL code during the re-examination of the '120 patent (T.T. 5/9/11 at 158-168).
- Nemes, inventor of the '120 patent, admitted at trial that he did not invent: (1) linked lists with on-the-fly garbage collection; (2) external chaining; (3) hash tables; (4) databases; (5) linked lists; (6) pointers; (7) deletion of expired records; (8) garbage collection; (9) collision resolution. These are the crucial pieces to the asserted claims of the '120 patent. Nemes' admissions on these points further evidence the testimony of Yahoo!'s witnesses that what Nemes claimed to invent was already well-known to those skilled in the art and, in fact, had been done already. (T.T. 4/27/11 p.m. at 50-54).

Noticeably absent from the NIRC, which issued on May 24, 2011, was any evidence of consideration of the NRL code or the '95 Linux Code—the exact same problem that Bedrock faced in the trial of this case.

³ Throughout this brief, the trial transcripts shall be referred to as "T.T." followed by the date of the trial transcript, the reference of a.m. or p.m. to reference the morning or afternoon transcript, and the specific page and line numbers of the testimony referenced.

Against this strong, voluminous evidence, Bedrock only provided conclusory argument from its expert, Dr. Mark Jones (Jones), limited to merely seven pages in the entire trial transcript.⁴ Jones summarily addressed only two of the prior art references—and notably ignored the other prior art references (the Apple patent, Knuth, Kruse, Stubbs, the '663 patent and the '495 patent and related application) that Yahoo! exhaustively covered through numerous witnesses at trial as invalidating the asserted claims. In particular, Jones:

- ignored the specific lines of source code from the '95 Linux Code and NRL code that Yahoo!'s witnesses, Williams and Filo, explained at trial invalidated claims 1 and 2 of the '120 patent (T.T. 5/9/11 at 283-288);
- did not disagree with Williams' testimony that combining two particular functions in the *key.c* file of the NRL code would meet all the elements of the '120 patent and would be a fairly easy thing to do (T.T. 5/9/11 at 288-289);
- did not challenge Williams' detailed analysis that the '495 patent, '663 patent, the Apple patent, Knuth, Kruse, or Stubbs references in combination would render the '120 patent obvious and, therefore, invalid (T.T. 5/9/11 at 289-290); and
- did not challenge Yahoo!'s evidence and references that Claim 2 was rendered obvious; nor did Jones present any evidence whatsoever rebutting the testimony and documents that Claim 2 of the '120 patent was invalid. (T.T. 5/9/11 at 287-288).

In short, Bedrock failed to present sufficient evidence at trial to refute Yahoo!'s overwhelming prior art references and other invalidity evidence. As such, a reasonable jury could have arrived at no other verdict than to invalidate the asserted claims of the '120 patent. Consequently, this Court should grant judgment as a matter of law invalidating claims 1 and 2 of the '120 patent.

LEGAL STANDARD IN THIS CASE

Judgment as a matter of law (JMOL) is appropriate when the court finds that a “reasonable jury would not have a legally sufficient evidentiary basis to find for the party on that

⁴ Although Bedrock's principals can read and program in C computer language, none had reviewed the prior art before trial.

issue.” *Orion IP, LLC v. Hyundai Motor Am.*, 605 F.3d 967, 973 (Fed. Cir. 2010); *i4i Ltd. P’ship. v. Microsoft Corp.*, 598 F.3d 831, 841 (Fed. Cir. 2010); Fed. R. Civ. P. 50(a). The Federal Circuit has frequently granted or affirmed JMOLs based on invalidity in patent cases. *See, e.g., Orion IP LLC*, 605 F.3d 967 (granting JMOL on invalidity); *Blackboard, Inc. v. Desire 2 Learn, Inc.*, 574 F.3d 1371, 1379-80 (Fed. Cir. 2009) (granting JMOL on invalidity); *Mahurkar v. C. R. Bard, Inc.*, 79 F.3d 1572 (Fed. Cir. 1996) (affirming JMOL of anticipation). *See also, Iovate Health Scis., Inc. v. Bio-Engineered Supplements and Nutrition, Inc.*, 586 F.3d 1376, 1380 (Fed. Cir. 2009) (affirming summary judgment of invalidity based on anticipation); *AdvanceMe Inc. v. RapidPay*, 509 F. Supp. 2d 593, 627 (E.D. Tex. 2007) (Davis, J), *affirmed, per curiam* 2008 U.S. App. LEXIS 10618 (Fed. Cir. 2008) (granting summary judgment on anticipation and obviousness).

Here, rather than applying a higher standard, the parties jointly stipulated for this Court to apply the preponderance of the evidence standard in deciding whether to invalidate the ’120 patent. Dkt. No. 636. Yet, the voluminous testimony and evidence that Yahoo! presented more than satisfy this standard or even the higher clear and convincing standard.

Importantly, because the jury did not return a finding on Yahoo!’s counterclaim of invalidity, and as Yahoo! preserved these issues through a timely Rule 50(a) motion, this decision—whether a reasonable jury would have a legally sufficient evidentiary basis to find the accused claims invalid—is proper and ripe for this Court. Fed. R. Civ. P. 50(b). In making that decision, this Court owes no deference to the PTO’s decision to confirm the accused claims in re-examination because the crucial pieces of prior art—the ’95 Linux Code and the NRL code—were not considered by the PTO. Judicial deference to an administrative action is appropriate only where the administrative action was the “product of reasoned decisionmaking.” *Motor Vehicle Mfrs. Ass’n v. State Farm Mut. Auto. Ins. Co.*, 463 U.S. 29, 52 (1983) (interpreting §

706(2)(A) of the APA). Reasoned decision-making requires, at a minimum, that “the agency must examine the relevant data and articulate a satisfactory explanation for this action, including a ‘rational connection between the facts found and the choice made.’” *Id.* at 43 (quoting *Burlington Truck Lines, Inc. v. United States*, 371 U.S. 156, 168 (1962)). In re-examining the ’120 patent, the PTO could not have engaged in any “reasoned decisionmaking” because there is no evidence that shows the ’95 Linux Code and the NRL code were considered by the PTO when assessing validity. Consequently, this Court should not defer to the PTO on this decision. *See Paid Search Engine Tools, LLC v. Yahoo! Inc.*, 746 F. Supp. 2d 808 (E.D. Tex. 2010) (Folsom, CJ) (granting summary judgment of invalidity based on prior art reference not considered by PTO).

Moreover, this Court should not rely on the PTO’s factual determinations because the agency did not have all relevant information before it, and, therefore, its fact-finding procedures were inadequate in this case. *See Citizens to Pres. Overton Park v. Volpe*, 401 U.S. 402, 415 (1971) (Supreme Court precedent authorizes *de novo* determination of factual issues “when the [PTO] action is adjudicatory in nature and the agency’s fact-finding procedures are inadequate”). As the Supreme Court has noted, the PTO is generally required to act “without the aid of the arguments which could be advanced by the parties interested in proving patent invalidity.” *Lear, Inc. v. Adkins*, 395 U.S. 653, 670 (1969). Here, this Court is not limited to the PTO record and, instead, should consider all the evidence—including Yahoo!’s witness, expert witnesses, third party witnesses, documentary evidence and the crucial missing prior art—and make a *de novo* determination as to validity of the accused claims of the ’120 patent.

For all the reasons detailed below and as evidenced at trial, Yahoo! is entitled to JMOL that claims 1 and 2 of the ’120 patent are invalid under 35 U.S.C. §§ 102(a), (b), (g)(2), and 103(a).

ARGUMENT

I. CLAIM 1 OF THE '120 PATENT IS INVALID

Claim 1 of the '120 patent is invalid as anticipated by the '95 Linux Code and is rendered obvious by the NRL code and the '495 patent with additional references. Of notable import, Kuznetsov, the author of the '95 Linux Code confirmed that his 1995 code invalidated the '120 patent. Furthermore, both Filo and Williams testified in detail as to how and where each element of the '120 patent was found in the '95 Linux Code and the NRL code.⁵ In contrast, Bedrock presented no evidence to controvert this detailed line-by-line source code testimony.

Besides the '95 Linux Code and the NRL code, the '495 patent combined with other prior art references that use external chaining for collision resolution invalidate the '120 patent. Only this Court—not the PTO—has before it the evidence that Bell Communications knew that it was readily apparent to one of ordinary skill in the art in 1988 to use other collision resolutions, such as the external chaining found in the '120 patent, with the '495 invention. For all these reasons specified below, claim 1 is invalid.

A. '95 Linux Code Anticipates Claim 1 of the '120 Patent

Here, the '95 Linux Code discloses every element of claim 1. “A patent is invalid for anticipation if a single prior art reference discloses each and every limitation of the claimed invention.” *Schering Corp. v. Geneva Pharms., Inc.*, 339 F.3d 1373, 1377 (Fed. Cir. 2003); *Iovate Health Scis., Inc. v. Bio-Engineered Supplements and Nutrition, Inc.*, 586 F.3d 1376, 1380 (Fed. Cir. 2009).

Kuznetsov, the author of the '95 Linux *route.c* code, confirmed that Linux versions

⁵ Neither the '95 Linux Code nor the NRL code were before or considered by the PTO in the original '120 patent prosecution. Further, neither piece of prior art was listed as considered by the PTO in its decision to issue the NIRC during the re-examinations in January 2011 or May 2011; nor were they included in the '120 patent Re-examination Certificate. (Ex. 19, Ex. 20, T.T. 4/27/11 p.m. at 144:2-6; 144:19-24).

1.3.51, 1.3.52 and 2.0.1⁶ were publicly available on the internet on or about December 27, 1995, December 29, 1995, and July 3, 1996, respectively. (T.T. 4/29/11 p.m. at 150:10-20; *see* Ex. 1). Indeed, Yahoo! had the '95 Linux Code in late 1995 or early 1996—almost two years prior to the '120 patent application. (T.T. 4/29/11 a.m. at 86:1-4, 19-87:1).

Filo, Kuznetsov, and Williams independently concluded that the '95 Linux Code invalidated the '120 patent:

- Filo concluded that the '95 Linux Code has “on-the-fly garbage collection with a hashing table and external chaining” as claimed by claims 1 and 2 and pin-pointed the specific lines of '95 Linux Code that meet the requirements of, and thus invalidate, claim 1 of the '120 patent. (T.T. 4/29/11 a.m. at 129:8-11; 129-132; 133:2-4).
- Kuznetsov, author of the '95 Linux Code, testified that his code “directly invalidates the ['120] patent.” (T.T. 4/29/11 p.m. at 152:22-23). This is consistent with Kuznetsov’s initial reaction in 2009 in which he stated that the '120 patent is “obviously invalid,” (Ex. 10), and his email in December 2010, “is not this ['95 Linux Code] enough to invalidate the patent.” (Ex. 11; Ex. 12).
- Williams put Nemes’ invention into perspective—by Nemes’ own admission that he did not invent hashing techniques using external chaining or on-the-fly garbage collection using linked lists—Nemes’ invention of combining these two things was not new or non-obvious, and, in fact, was done before. (T.T. 5/9/11 at 185:1-25). Williams further explained element by element why the '95 Linux Code invalidated claims 1 and 2 of the '120 patent. (*Id.* at 186:5-8; 187-198, 204-05).

Bedrock does not dispute that the '95 Linux Code discloses each element of claim 1; instead, it merely contends, without support, that the '95 Linux Code only has stand-alone garbage collection⁷ instead of on-the-fly garbage collection -- a position resoundingly rejected by the Code’s author, the source code, Filo, and Williams.

⁶The code in the three versions of the '95 Linux Code that perform the same on-the-fly garbage collection as the '120 patent are located in different lines of code due to other changes made to the Linux code in the various versions. For convenience, the source code lines that will be referenced in this motion will be Linux v.2.0.1, which comports with the trial testimony of Filo and Williams.

⁷ Yahoo! does not contest that the '95 Linux Code has multiple types of garbage collection – including stand-alone and on-the-fly garbage collection. The type of garbage collection at issue in the '95 Linux *route.c* source code is unquestionably on-the-fly. (T.T. 4/29/11 a.m. at 129:8-11).

1. Kuznetsov Confirmed that the '120 Patent is Invalid

Kuznetsov, a Russian computer programmer and researcher in physics for the Institute for Nuclear Research who wrote the '95 Linux *route.c* code, immediately recognized the '120 patent was invalid. (Ex. 10). When first confronted with the '120 patent in October 2009, Kuznetsov reviewed the patent and his prior '95 Linux Code and responded:

This is ridiculous. The patent is obviously invalid, hashing cannot be covered by a patent, it is described in all the textbooks since Computer Science emerged. :-)

Even formally the patent is filed on Jan 2, 1997. The code in net/ipv4/route.c was written in 1995 (see linux kernel source archives) and got to its final stable version in linux-2.0, which is dated by Jul 1996.

(Ex. 10). The next year, after being reminded of his '95 Linux Code, Kuznetsov again confirmed the invalidity of the '120 patent:

I did not even look so far behind. :-) Is not this [his '95 Linux *route.c* source code] enough to invalidate the patent?

(Ex. 11, 12). Furthermore, Kuznetsov confirmed at trial that the '120 patent was invalid, (T.T. 4/29/11 a.m. at 12:16-18), the '120 patent collided with his work done in 1995, (T.T. 4/29/11 a.m. at 13:10-11), and that lots of references dating back to 1985 described the techniques in the '120 patent. (T.T. 4/29/11 a.m. at 14:20-21). Although Bedrock repeatedly questioned him regarding infringement, Kuznetsov was firm in his response: "I told him that I have code dated by 1995, which directly invalidates the patent." (T.T. 4/29/11 p.m. at 152:21-23). Bedrock does not, because it cannot, contest Kuznetsov's testimony, statements, or emails regarding the invalidity of the '120 patent.

2. The '95 Linux Code Discloses Each Element of Claim 1

Furthermore, as Williams and Filo explained, each element of claim 1 is taught in the '95 Linux Code. Bedrock does not contest the second element and only half-heartedly attempts to rebuff the other three. Against the overwhelming testimony and evidence to the contrary,

Bedrock's weak attempt fails.

a. '95 Linux Code Discloses the First Element of Claim 1

Claim 1 recites a "linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring." (Ex. 13 at 13:26-28). The '95 Linux Code specifically discloses this element, including the claimed linked list, (T.T. 5/9/11 at 186-198; Ex. 1), and that at least some of the records stored in the hash table of route table entries in the linked list "automatically expire." Williams explained why and where in the '95 Linux Code this element was met:

Q. Okay. So focusing in on that first element, which is the linked list with automatically expired records, let's look at the Linux prior art code. Show how it has that first element of the claim.

A. Okay. You can see over on the left side is the actual structure of a record. That's the Post-it note. And at Line 67 is the link. That's the link to the next Post-it note. That's the little arrow to the next Post-it note.

And the code on the right-hand side is actually using that. And Line 1345 is where we access the hash table. It uses a hash value to access the hash table, and that gets to the head of the linked list. That's the access.

And then it steps through the linked list at Line 1365 and Line 1372. Line 1382 is where it's going to the next record.

Q. And does the linked list have automatically expired records?

A. Yes, sir, it does.

Q. Let's see, where is that in the code?

A. Well, this, again, that same code, and I've highlighted a comment at Line 1362. That's a note to the programmer, and it says clean up duplicate and aged-off entries. So the aged off-entries are the expiring records. And the code that actually does that that detects those is on Line 1369.

You can see this expression that says the `rt_cache_timeout`. That's detecting the expired record as it's stepping through the records.

(T.T. 5/9/11 at 188:19-189:23; see Ex. 1, file *route.h*, structure *rtable*, l. 65,67, file *route.c*, function *ip_rt_check_expire*. at 968, *rt_cache_add*, at 1369-80). Similarly, Filo also confirmed

that the '95 Linux Code meets this requirement of claim 1:

And the idea is you start with the first record and you iterate through. While we're walking through the list here, if you look at Line 1369, we will see that there is a check to see if -- here we see this `Cache_TIMEOUT`, and basically what this is doing is checking to see if this particular record in the linked list has expired.

We identify that it has expired, and immediately in Lines -- well, in the Lines 1372 through 1378, that is where it's removing the expired record. And so this is, again, all within the same access of the linked list while we're walking it. It has identified and removed the expired record.

(T.T. 4/29/11 a.m. at 130:18-131:5). As such, the first element is met.

b. '95 Linux Code Discloses the Second Element of Claim 1

Claim 1 also requires a "record search means utilizing a search key to access the linked list." (Ex. 13 at 13:29-30). The '95 Linux Code discloses this second element (means plus function) as well—a search key, a form of destination address that is used as an input to a hash function to access the linked list. (Ex. 1, file *route.c*, function *rt_cache_add*, at 1345; file *route.c*, function *rt_cache_add*, at 1365-83; T.T. 5/9/11 at 183:6-11, 187:25-188:18, 190:1-18). As Williams explained at trial:

Q. Let's go to the second element, which is that records search means. Let's see where the Linux prior art discloses this element. Can you describe that for us?

A. Yeah. There's a search means, again, using a hash value to access the head of the linked list. And at Line 1365, we start stepping through the linked list looking for a duplicate record.

(T.T. 5/9/11 at 189-190). Filo also confirmed that the '95 Linux Code discloses this element.

(T.T. 4/29/11 a.m. at 130:9-11).⁸ Bedrock does not contest this element; therefore, the '95 Linux Code meets this element.

⁸ Further, to calculate the hash value, *rt_redirect_1* calls the function *ip_rt_hash_code*, which is defined in *route.h* at 116-120. The resulting value is passed into the *rt_cache_add* function and is used to access the appropriate linked list in the hash table *ip_rt_hash_table*. (Ex. 1, file *route.c*, function *rt_cache_add*, at 1345). That linked list is then traversed in search of a record matching the search key (the destination address) and, if found, deletes it from the linked list. (Ex. 1, file *route.c*, function *rt_cache_add*, at 1365-83).

c. '95 Linux Code Discloses the Third Element of Claim 1

The third element of claim 1 recites “the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.” (Ex. 13 at 13:31-34). As required by this Court’s claim construction, the ‘95 Linux Code discloses that during the access to the linked list of route table entries to search for a duplicate entry to delete, it both identifies and removes route table entries that have expired. (T.T. 5/9/11 at 183:6-11, 187:25-188:18, 191:15-21). Williams explained, “[a]nd at Line 1365, we start stepping through the linked list looking for a duplicate record. This is in Line 1370; 1370 is where we detect the duplicate record. And 1372 is where the duplicate record would get removed.” (T.T. 5/9/11 at 190:6-11; *see* Ex. 1, file *route.h*, structure *rtable*, at 65, file *route.c*, function *rt_cache_add* at 1365-83).

In particular, the ‘95 Linux Code identifies an expired record on lines 1369-1370 based on whether the reference count is zero and the lifetime of the record has passed. After an expired record is identified, it is then removed from the linked list by adjusting the pointers in the linked list to bypass the record. Filo described, “it’s 1372, which is what changes the pointer and skips over and is effectively taking that record out of the list.” (T.T. 4/29/11 a.m. at 132:8-10; *see* Ex. 1, file *route.c*, function *rt_cache_add* [at 1372]). As such, the third element is met because the ‘95 Linux Code discloses the identical function and structure of this Court’s Claim Construction Order. (T.T. 5/9/11 at 183:6-11, 187:25-188:18; 191:15-21).

d. '95 Linux Code Discloses the Fourth Element of Claim 1

Finally, claim 1 recites “means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.” (Ex. 13 at 13:35-38). Williams specifically testified that this element is met by the ‘95 Linux Code:

Q. All right. Let's go to the last element of Claim 1, which is the accessing and removing where the Court has interpreted. It has to be inserting, retrieving, or deleting.

Again, faithful to the claim construction of the Court, does the Linux prior art disclose this element?

A. Yes, sir, it does. You can see that at Line 1345. Again, it's accessing that hash table to get the linked list. And at Line 1356 and Line 1357 is where we are actually inserting the record. That's inserting the record. Then we continue on that same access to search for the duplicate record and also the expired record to on-the-fly garbage collection.

Q. Again, so this is the same access we have described to the last element, correct?

A. Yes, sir.

Q. Okay, so this comment helps to explain – there is a comment on here that helps explain what is happening?

A. Yes. Again, it is Line 1362. It's clean up the duplicate records and the aged-off record, and that is the heart of the '120 patent.

(T.T. 5/9/11 at 191:22-192:19). As Williams testified with respect to this final element of claim 1, the '95 Linux Code discloses the identical function and the equivalent structure of this Court's Claim Construction Order. (T.T. 5/9/11 at 183:6-11, 187:25-188:18, 193:13-25).

3. Bedrock's Conclusory Response to the '95 Linux Code Lacks Merit

Lacking the ability to take on the code to rebut the testimony of Williams and Filo and the ability to address Kuznetsov's testimony and e-mails, Bedrock resorted to the tactic of diverting attention from the '95 Linux Code and referring to "old locks" and stand-alone garbage collection. Bedrock's conclusory statements, without more, are a ruse and are insufficient to survive JMOL. *See Davis v. McDowell*, 596 F.3d 1355, 1364 (Fed. Cir. 2010).

Furthermore, these statements are incorrect. Bedrock's statement that "clis" and "stis" are equivalent to "old locks" that allegedly define access is without merit, is not described in the '120 patent, is not included in this Court's claim construction, and was rejected by the jury in

this case.⁹ The clis and stis do not indicate that more than one access is taking place, nor are they mentioned anywhere in the '120 patent. (T.T. 5/9/11 at 193:4-9). Nemes admitted that the '120 patent did not include any description of “lock”, nor did it define “access” through or with “locks.” (T.T. 4/27/11 p.m. at 55:18-56:7). As Williams explained, the clis and stis are like “hanging out or (sic) do not disturb sign that just tells other software not to mess with this software while it’s doing this steps that it does. It doesn’t have anything to do with accesses.” (T.T. 5/9/11 at 192:22-193:2).

Although a “linked list” in the ‘95 Linux Code is the same as a “linked list” in the '120 patent, Bedrock attempts, through the clis and stis, to narrowly define each record in the linked list in the ‘95 Linux Code as a separate “access”.¹² Only through this failed, tortured construction of the locks defining “access” so that each “record” is accessed, rather than the entire linked list being accessed, can Bedrock pretend that the ‘95 Linux Code is stand-alone garbage collection rather than on-the-fly garbage collection—a position strongly rejected by the ‘95 Linux Code itself, as well as its author, Filo and Williams. (Ex. 1, file *route.c*, function *rt_cache_add*, at 1361-83). Indeed, after walking the jury through the ‘95 Linux Code, Filo concluded:

Q. Okay. So does [the ‘95 Linux Code] -- Defendant's Exhibit No. 48 describe on-the-fly garbage collection with external chaining in a linked list?

A. Yes, ma'am, it does.

Q. Does it also have the automatic removal of expired records?

⁹ At trial, Bedrock argued that all source code lines between “spin-lock” and “spin-unlock” in the Linux Candidate Code constituted one “access” of the linked list. (T.T. 4/28/11 p.m. at 62:3-63:19). The jury soundly rejected this argument when it found Yahoo! did not infringe claims 1 and 2. (Dkt. No. 834). Bedrock’s attempt to distinguish the ‘95 Linux Code by interjecting its failed “lock” argument into its invalidity analysis is misplaced because the jury found that “locks” in the code do not determine the access. Moreover, the concepts of locks—whether spinlocks or clis/stis—are not included in the '120 patent. (T.T. 4/27/11 p.m. at 55:23-56:7).

A. Yes, ma'am. I talked -- just talked about that. It identifies the records and removes them while it's walking the list.

(T.T. 4/29/11 a.m. at 132:11-19).

Notably, Nemes, who also can read and program in C - the computer language in which the '95 Linux Code was written, never reviewed the '95 Linux Code prior to applying for the '120 patent or filing this suit. (T.T. 4/27/11 p.m. 141:11-19).¹⁰ Nevertheless, Nemes admits that if the '95 Linux Code predates the '120 patent and was structurally and functionally the same as the '120 patent, then the '120 patent would be invalid. (T.T. 4/27/11 p.m. 141:20-25). Indeed, considering this admission and the overwhelming evidence regarding the '95 Linux Code, no reasonable jury could find that claim 1 was not invalid. As such, for this reason alone, this Court should invalidate claim 1 of the '120 patent.

B. The NRL Code Renders Claim 1 Obvious

The elements contained in two portions of one file of the NRL code,¹¹ when combined, invalidate the '120 patent. A patent is invalid as obvious “if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” 35 U.S.C. § 103(a). “The combination of familiar elements with known methods is obvious when it provides no functionality except for yielding predictable results.” *AdvanceMe Inc.*, 509 F. Supp. 2d at 610 ; *see KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-416 (2007). Such is the case here. This Court has previously invalidated patents for obviousness when “one of skill in the art could easily view the prior art and make the common sense leap” to the asserted claims. *AdvanceMe*, 509 F. Supp. 2d at 625. In particular,

¹⁰ Moreover, Nemes had never reviewed any version of Linux, all of which were publicly available on the internet.

¹¹ The NRL code was publicly available on the internet by late 1995. (Ex. 14; Ex. 2; T.T. 5/9/11 at 139:14-16).

this Court should consider four factors: (1) scope and content of the prior art; (2) differences between the claimed invention and the prior art; (3) level of ordinary skill in the relevant art; and, if applicable, (4) objective or secondary considerations tending to prove nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 18 (1966). Each of these factors leave no other result than the '120 patent being invalid for obviousness.

“[W]hen a patent simply arranges old elements with each performing the same function it had been known to perform and yields no more than one would expect from such an arrangement, the combination is obvious.” *KSR Int’l Co.*, 550 U.S. at 417 (quotation omitted). In this case, Nemes merely combined known elements that were already performing the same function in the NRL code. Bedrock acknowledges that these elements are contained in the NRL code – it merely states that they are not combined. This misses the mark and is not sufficient to survive JMOL. For the reasons below, the NRL code invalidates claim 1.

The combination of two functions in the NRL code’s *key.c* file—*key_acquire* and *key_search*—renders claim 1 obvious. The *key_acquire* function performs on-the-fly garbage collection with a linked list but does not have external chaining. And the *key_search* function uses a hash table with external chaining. McDonald, a computer scientist and author of the NRL code, testified at trial that the reason why *key_acquire* did not use a hash table with external chaining was because it was not dealing with a large number of records. (T.T. 5/9/11 at 141:2-12). He further testified that, if there were a need to store a large number of records, “I would have used a hash table with linked list chaining” and that it would have been obvious to combine hashing or external chaining with the on-the-fly garbage collection of *key_acquire*. (T.T. 5/9/11 at 141:25-142:24).

But the Court has more than McDonald’s testimony. Both Filo and Williams were more

than well-skilled in the art¹² and testified that it would have been obvious to combine the *key_acquire* and *key_search* functions to invalidate claim 1 of the '120 patent. Williams walked through the lines of code in the *key.c* file and explained that “they’re side by side in the same software. So it would have been obvious to someone to combine those if they needed to, and you would get all the elements of the '120 patent; and that’s—combining that in that way is a perfectly reasonable thing to do.” (T.T. 5/9/11 at 198:9-23). Filo also testified that all elements of claim 1 were present and, when combined, would invalidate the patent:

Q. All right. So within the *key.c* of the NRL code, you had one part of the code that talked about a hash table with external chaining, and another part of the code—code that talked about on-the-fly garbage collection with a linked list; is that right?

A. Yes, ma'am.

Q. And combined these two references, these two lines, sections of lines that you've talked about from the NRL code, do they invalidate the '120 patent?

A. I believe they do.

(T.T. 4/29/11 a.m. 139:14-23).

Bedrock’s expert Jones does not dispute that all the elements of claim 1 are contained in the NRL’s *key.c* file—he merely acknowledges Filo’s and Williams’ testimony that they are in two separate functions of this one file. Instead, Bedrock’s only basis in the entire trial transcript for the NRL code not invalidating the '120 patent is Jones’ unexplained, conclusory statement that it would not be obvious to combine the two portions in the *key.c* file. This single statement rings hollow when Jones concedes that combining the *key_search* and *key_acquire* functions to meet the elements of claim 1 of the '120 patent would have been easy to accomplish. (T.T. 5/9/11 at 289:7-14). In the face of overwhelming testimony and documents to the contrary,

¹² At trial, the parties did not have a material factual dispute as to the level of ordinary skill in the art. (T.T. 5/10/11 at 39-40).

without any explanation, Bedrock's feigned denial is not sufficient to survive JMOL before this Court. *See Davis v. McDowell*, 596 F.3d 1355, 1364 (Fed. Cir. 2010) (an expert's conclusory statements without supporting facts are insufficient to survive summary judgment).

As shown below, each of the elements of claim 1 are present in the *key.c* file of the NRL code, which Bedrock does not contest.¹³

1. The NRL Code Discloses the First Element of Claim 1

The NRL code meets this element and uses a linked list of records called a *key_acquirelist* (*see* Ex. 13 at 13:26-28; Ex. 2 at 129). Filo and Williams both testified that the NRL code's *key_acquire* function uses a linked list to store and provide access to records in the memory of the system:

- Filo testified that the NRL Code discloses a linked list at line 1431-1459. (T.T. 4/29/11 a.m. at 135:24-136:15).
- Williams testified that there is a routine in the NRL code that “does on-the-fly garbage collection on a linked list.” (T.T. 5/9/11 198:17-19; *see* Ex. 2 at 1447-1448).

Further, the records in the linked list automatically expire because in each *key_acquirelist* record there is an expiration time and by default is set to expire in 15 seconds after the record is added to the list. As such, the records become obsolete and are therefore no longer needed in the storage system after their expiration time (Ex. 2, function *key_acquire*, at 116, 132, 1445-1457, 1557-59). Bedrock does not dispute this element, and it is met.

2. The NRL Code Discloses the Second Element of Claim 1

The *key.c* file of the NRL code contains both the *key_acquire* and *key_search* functions. The *key_acquire* function in combination with the *key_search* function discloses a record search

¹³ Further, Bedrock failed to present evidence or testimony to rebut that the NRL code in combination with other prior art references, such as Knuth, Kruse, or Stubbs, would render claim 1 obvious. (Ex. 2, Ex. 7, Ex. 8, Ex. 9).

means utilizing a search key to access the linked list. (T.T. 5/9/11 at 197-201; *see* Ex. 13 at 13:29-30). Specifically:

- The *key_acquire* function gets the head of the linked list (via *key_acquirelist*) and traversing the linked list in search of a record with a key that matches the search key. (Ex. 2, function *key_acquire*, at 1430-59; T.T. 5/9/11 at 200; T.T. 4/29/11 a.m. 139:14-141:14; 4/29/11 p.m. at 38:10-18).
- The *key_search* function performs a hash to get to the head of the externally chained linked list. (Ex. 2 at 675-682).

Filo and Williams both explained that these two functions in combination meet this means plus function requirement of claim 1. (Filo T.T. 4/29/11 a.m. 139:14-141:14; 4/29/11 p.m. at 38:10-18; Williams T.T. 5/9/11 at 183:6-11, 187:25-188:18, 197-201). Moreover, at the time of application of the '120 patent, hashing per collision resolution was a well-known technique taught in prior art references such as Knuth, Kruse, and Stubbs. Any of these prior art references in combination with the *key_acquire* function meet this claim limitation (*i.e.*, have the identical function and the identical or equivalent structure identified by the Court). (*See* Ex. 7, pp. 513-18; Ex. 8, pp. 206-08; Ex. 9, pp. 324-25). Bedrock does not contest this element, and it is met.

3. The NRL Code Discloses the Third Element of Claim 1

Specifically, expired records are identified and removed at lines 1431-1459 of the NRL code as Williams and Filo explained in detail during trial. (Ex. 15 at 188-194; Ex. 2 at 1431-59; Filo T.T. 4/29/11 a.m. at 134-42; Williams T.T. 5/9/11 at 199:4-9). This is made clear by the NRL developers' comment in the source code, stating, "[S]ince we're already looking at the list, we may as well delete expired entries as we scan through the list." (Ex. 2 at 1447-51; Ex. 2 at 1446-52). Indeed, this comment mirrors the same comment in the "Heart of the Technique" included in the pseudocode of the '120 patent: "HEART OF THE TECHNIQUE: Traverse entire list, deleting expired records as we search."

The NRL Code

```
1431 for(ap = key_acquirelist->next; ap; ap = ap->next) {
1432   if (addrpart equal(dst, (struct sockaddr *)&(ap->target)) &&
1433       (etype == ap->type)) {
1434     DPRINTF(IDL_MAJOR_EVENT, ("acquire message previously sent!\n"));
1435     if (ap->expiretime < time.tv_sec) {
1436       EPRINTF(IDL_MAJOR_EVENT, ("acquire message has expired!\n"));
1437       ap->count = 0;
1438       break;
1439     }
1440     if (ap->count < maxkeyacquire) {
1441       DPRINTF(IDL_MAJOR_EVENT, ("max acquire messages not yet
1442         exceeded!\n"));
1443     }
1444     return(0);
1445   } else if (ap->expiretime < time.tv_sec) {
1446     /*
1447      * Since we're already looking at the list, we may as
1448      * well delete expired entries as we scan through the list.
1449      * This should really be done by a function like key_reaper()
1450      * but until we code key_reaper(), this is a quick and dirty
1451      * hack.
1452     */
1453     DPRINTF(IDL_MAJOR_EVENT, ("found an expired entry...deleting
1454       it!\n"));
1455     prevap->next = ap->next;
1456     KFree(ap);
1457     ap = prevap;
1458     prevap = ap;
1459   }
}
```

'120 Patent

```
Search Table Procedure
function search_table(record_key: record_key_type;
  var position: list_element_pointer;
  var previous_position: list_element_pointer;
  var index: 0 .. table_size - 1): boolean;
/* Search table for record_key and delete expired records in target list; if found, position is made to
point to located record and previous_position to its predecessor, and TRUE is returned; otherwise
FALSE is returned. index is set to table subscript that is mapped to by hash function in either
case. */
var p: list_element_pointer; /* Used for traversing chain. */
previous_p: list_element_pointer; /* Points to p's predecessor. */
begin
  index := hash(record_key); /* hash returns value in the range 0 .. table_size - 1. */
  p := table[index]; /* Initialization before loop. */
  previous_p := nil; /* Ditto */
  position := nil; /* Ditto */
  previous_position := nil; /* Ditto */
  while p # nil
    /* HEART OF THE TECHNIQUE: Inverse entire list, deleting
    /* expired records as we search. */
  begin
    if p^.record_contents is expired
      then remove(p, previous_p, index) /* ON-THE-FLY REMOVAL OF EXPIRED RECORD. */
    else begin
      if position = nil then if p^.record_contents.key = record_key
        /* If this is record wanted, */
      then begin position := p; previous_position := previous_p end;
        /* save its position. */
      previous_p := p; /* Advance to */
      p := p^.next /* next record. */
    end /* else begin */
  end;
  return (position # nil) /* Return TRUE if record located, otherwise FALSE. */
end /* search_table */
```

As such, the NRL code discloses the identical function and identical or equivalent structure for identifying and removing at least some of the expired records from the linked list that is required by the Court's claim construction order. (Ex. 14 at 1431-59; T.T. 5/9/11 at 183:6-11, 187:25-188:18, 197-201; Ex. 13 at 13:31-34). Bedrock neither disputed this nor presented any evidence in rebuttal.

4. The NRL Code Discloses the Fourth Element of Claim 1

The *key_acquire* function at lines 1431-59 retrieves a record matching the search key and, at the same time, removes expired records from the linked list as described above. (Ex. 2 at 1421-1459; T.T. 4/29/11 a.m. at 137-139; T.T. 5/9/11 at 198-201; *see* Ex. 13 at 13:35-38).

To the extent the *key_acquire* function does not disclose the structure for this limitation, the *key_acquire* function in combination with the *key_search* and *key_delete* functions discloses this element. Yahoo!'s expert Williams explained at trial that this would have been a fairly easy and obvious thing to do. (T.T. 5/9/11 p.m. at 199-201). The *key_search* function is called by the function *key_delete*. (Ex. 2 at 649, 675-682, 1014-16). And *key_delete* removes expired

records from the hash table by adjusting pointers to bypass the record. (Ex. 2 at 1020, 1031). As required by this Court's claim construction, this meets the identical function and the identical or equivalent structure for this final element of claim 1. (T.T. 5/9/11 at 183:6-11, 187:25-188:18, 197-201). As set forth above, Bedrock failed to rebut this evidence. Given the breadth of the evidence, a reasonable jury could only find, as should this Court, that claim 1 is invalidated by the NRL code.

C. The '95 Linux Code and NRL Code Were Not Considered by the PTO

Although Bedrock asserts that claims 1 and 2 were confirmed in re-examination,¹⁴ the testimony and evidence proved that Bedrock did not submit the crucial invalidating prior art—the '95 Linux Code or the NRL code—to the patent examiner. Indeed, Bedrock never identified the NRL code on an Information Disclosure Sheet (IDS) or submitted the NRL code to the PTO either in the original patent prosecution as admitted by Nemes, (T.T. 4/27/11 p.m. at 144:2-6; 144:19-24), or during the re-examination of the '120 patent. (T.T. 4/27/11 p.m. at 144:19-145:23; 5/9/11 160:11-161:4).¹⁵ Although Bedrock identified 87 prior art references in its submission of two IDS forms to the PTO, it notably failed to identify the NRL Code on an IDS or submit the NRL code to the PTO during re-examination. (*See* Ex. 16 (61 prior art references identified); Ex. 17 (26 prior art references identified)).

Furthermore, Godici, a former Commissioner of Patents, clarified that although the three versions of the '95 Linux Code were identified in the December 2010 IDS, (Ex. 17), that same IDS was not before the patent examiner until after the NIRC issued in January 2011. (T.T.

¹⁴ On May 24, 2011, the PTO issued a NIRC on the second re-examination. Notably, neither the second NIRC nor the documents referenced as considered in the second NIRC mention the '95 Linux Code or the NRL code. Moreover, those two material pieces of prior art are noticeably absent from Bedrock's response brief to the PTO.

¹⁵ Indeed, Bedrock's failure to identify known prior art references on an Information Disclosure Sheet during re-examination does not comply with Patent Rule 98 and did not discharge Bedrock's duty of disclosure. (T.T. 5/9/11 at 154:2-9, 162:5-12).

5/9/11 at 163:16-164:10; Ex. 18).¹⁶ Therefore, the '95 Linux Code would not have been before the examiner, and there was no evidence that it was considered in re-examination. (T.T. 5/9/11 at 196:2-9; 197:16-21). Most importantly, none of the three versions of the '95 Linux Code were listed among the 61 references considered by the PTO when it issued the NIRC in January 2011 (Ex. 19), nor were any versions of Linux source code or the NRL code included on the April 12, 2011, Re-examination Certificate. (Ex. 20; T.T. 5/9/11 at 166:17-167:5).

Bedrock did not present any evidence—document or testimony—that the examiner considered the NRL code or the '95 Linux Code during re-examination. Consequently, Bedrock's argument that the re-examination is evidence of validity falls flat, should be given no deference, and wholly fails to rebut Yahoo!'s evidence that the '95 Linux Code and NRL code render the '120 patent invalid.

D. The '495 Patent Invalidates Claim 1

The '495 patent, listing Nemes as the inventor, entitled "Methods and Apparatus for Information Retrieval Utilizing Hashing Techniques," was filed on October 31, 1989, and issued on June 9, 1992. (Ex. 3). At trial, Bedrock did not offer any evidence rebutting Williams' testimony that the '495 patent in obvious combination with the numerous references of external chaining renders the '120 patent invalid. Instead, Bedrock's expert, Jones, merely stated that the '495 patent differed from the '120 patent and had a "zigzag." This testimony is not on point and is insufficient to survive JMOL. *See Davis*, 596 F.3d at 1364.

The '495 patent is directed to the same problem as the '120 patent: using hashing techniques in an information storage and retrieval system and on-the-fly removal of automatically expiring records from the hash table. The only difference between the '120 patent

¹⁶ Moreover, the PTO record is unclear whether Bedrock submitted the actual '95 Linux Code to the patent examiner. (T.T. 5/9/11 at 161:17-162:4).

and the '495 patent is how a hash collision is resolved—the '120 patent uses external chaining, while the '495 patent uses linear probing.

But using either *external chaining or linear probing* as collision resolution mechanisms for on-the-fly garbage collection in 1989 would have been an insignificant change in the '495 patent. (T.T. 5/9/11 at 201:21-202:12). By the 1970s, it was widely known that using linear probing or external chaining resolved collisions and was taught in: the seminal Knuth treatise in 1973 (Ex. 7, pp. 513, 518), the Background of the '120 patent (Ex. 13 at 1:53-64), and the 1987 Kruse and 1985 Stubbs data structures texts referenced in the specification of the '120 patent (Ex. 8, pp. 202, 206, Ex. 9, pp. 324-25).^{17,18} Because collision resolution based on external chaining has advantages over linear probing when deleting records from a hash table, it would have been obvious to try to modify references teaching a method of deletion in hash tables with collision resolution based on linear probing to use collision resolution based on external chaining. It was well-known and easy to do. (T.T. 5/9/11 at 203:9-16).

In fact, in 1991, Nemes designed a system to shift between collision avoidance by linear probing and collision avoidance by external chaining, which is described in the Abstract of his own patent, U.S. Patent No. 5,287,499 ('499 patent). (Ex. 21 at p. 1; T.T. 5/9/11 at 9-21).

In addition to Nemes being aware of the ability to use these two collision resolution techniques in one system, the engineers and attorneys at Bell Communications Research Inc. ("Bellcore"), the assignee of the '495 patent, knew the '495 patent could easily be modified by one of ordinary skill in the art to apply to different collision resolution methods. They

¹⁷ Filo recalled that he learned the types of collision resolution, including linear probing, linked lists and external chaining in college in the mid '80s. (T.T. 4/29/11 a.m. at 74:10-21). Williams testified that external chaining and linear probing were well-known techniques and that textbooks taught the techniques side by side. (T.T. 5/9/11 at 203:2-18).

¹⁸ Although all of these references were before the PTO in the '120 patent prosecution, the chapters or specific page numbers of the references were not, nor was the import of the references described as required by Patent Rule 98.

specifically included that language in the original application for the '495 patent:

It is to be understood that the present invention will be described in connection with linear probing with open addressing only for convenience and because such a collision-resolution strategy is very commonly used. The techniques of the present invention can just as readily applied [sic] to such other forms of collision-resolution strategies by modifications readily apparent to those skilled in the art.

(Ex. 22 at Telcordia00000253). Nemes acknowledged that the Bell attorneys who wrote this paragraph were the brightest engineers and patent attorneys and made sure they captured his invention (T.T. 4/27/11 p.m. at 81:3-6, 14-20). Nevertheless, consistent with Bedrock and its principals' failure to disclose the material prior art to the PTO, this information—*that Bell knew that one of ordinary skill in the art could easily modify the '495 patent from one collision resolution to another*—was never before the PTO. Instead, Nemes admits that he struck out this paragraph, (T.T. 4/27/11 p.m. at 83:1-12), and did not submit this information to the PTO either in the patent prosecution or the re-examination. (T.T. 4/27/11 p.m. at 83:13-18; 84:23-85:1; 85:14-22).¹⁹

Although Nemes may now adopt a litigation-driven position and deny that he knew how to modify from linear probing to external chaining (a position belied by his '499 patent), Nemes' claimed knowledge is *not* the test. The relevant “question is not whether the combination was obvious to the patentee but whether the combination was obvious to a person with ordinary skill in the art.” *KSR Int'l Co.*, 550 U.S. at 420; *see Standard Oil Co. v. Am. Cyanamid*, 774 F.2d 448, 453-54 (Fed. Cir. 1985) (obviousness is based on a “hypothetical person having ordinary skill in the art”); *Black v. Ce Soir Lingerie Co.*, 2008 WL 3852722, at *3 (E.D. Tex. 2008) (“entirety of the obviousness inquiry should be expansive and flexible and account for common sense”). Bell

¹⁹ Indeed, Nemes never informed Bellcore (or Telcordia—Bell's successor) that he was applying for the '120 patent or that his idea of the '120 patent was the same as the '495 patent with a modified collision resolution of external chaining (T.T. 4/27/11 p.m. at 99:25-100:13).

Communications' admission by its own patent lawyers and engineers is clear evidence that this was well-known by one of ordinary skill in the art in 1988.

Moreover, Nemes admitted that he copied significant portions of the '495 patent into the specification, figures, and claims of the '120 patent. (T.T. 4/27/11 p.m. at 96:19-97:25; 98:1-99:24). Indeed, there is little fundamental difference between the '495 patent and the '120 patent beyond the former's use of collision resolution based on linear probing and the latter's use of external chaining. (T.T. 5/9/11 at 201-203).²⁰

As set forth below in the simple chart, each of the elements of claim 1 of the '120 patent is described in the '495 patent. The '495 patent in combination with references evidencing external chaining and linked lists render claim 1 obvious.

'495 Patent (1989)	'120 Patent (1997)
<u>Claim 1</u>	<u>Claim 1</u>
"An information storage and retrieval system using hashing techniques to provide rapid access to the records of said system and utilizing a linear probing technique to store records with the same hash address, at least some of said records automatically expiring, said system comprising"	"An information storage and retrieval system, the system comprising a linked list to provide access to records stored in a memory of the system, at least some of the records automatically expiring"
"a record search means utilizing a search key to access chain or records having the same hash address"	"a record search means utilizing a search key to access the linked list"
"said record search means including means for identifying and removing all expired ones of said records from said chain of records each time said chain is accessed"	"the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed"
"means, utilizing said record search means, for inserting retrieving and deleting records from said system and, at the same time, removing all expired ones of said records in the accessed chains of records"	"means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list."

²⁰ Applying the Court's Claim Construction Order with respect to each element, Williams testified that the '495 patent renders the '120 patent obvious and did so by comparing the claim language of the claims of the '495 patent to the '120 patent; the figures of the '495 patent to the figures of the '120 patent; and put in perspective that the techniques of external chaining and linear probing were well-known techniques at the time of the '120 patent and obvious to switch from one to another. (T.T. 5/9/11 at 183:6-11, 187:25-188:18, 201:17-203:24).

Bedrock does not dispute that these elements are met and it does not distinguish Bell’s statement. As such, no reasonable jury could find that claim 1 was not invalid; therefore, JMOL should be granted with respect to claim 1 of the ’120 patent.

II. CLAIM 2 OF THE ’120 PATENT IS INVALID

Claim 2 depends from claim 1 and further requires a “means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records”. (Ex. 13 at 13:40-42). Claim 2 is rendered obvious by combining any of the prior art references for claim 1 with the Apple patent or the ’663 patent, both of which disclose making a dynamic determination to remove records from a system. (Ex. 5; Ex. 6). Importantly, Bedrock neither disputed the disclosures of the Apple patent nor the ’663 patent nor did it present any evidence to rebut those references with any of the various combinations described above. Instead, Bedrock only contended that the ’95 Linux Code did not “dynamically determine” and therefore did not invalidate claim 2. Nevertheless, Bedrock did not contest that the ’95 Linux Code combined with the Apple or the ’663 patents renders claim 2 obvious—therefore, its statement has no weight. *See Davis*, 596 F.3d at 1364. As such, for any of the reasons described below, JMOL should be granted that claim 2 of the ’120 patent is invalid.

A. Bedrock Does Not Contest Yahoo!’s Invalidating Testimony

Applying the Court’s Claim Construction Order, Williams explained:

Q. What's claimed in Claim 2?

A. Claim 2 is about determining the number—maximum number of records to remove when you're going through the list, and you need to dynamically determine that.

* * * *

Q. [D]id you consider this reference [Apple patent] in forming your opinion that Claim 2 is invalid?

A. Yes, I did. This is an Apple patent. It does memory management of

some records, and they dynamically determine the number of records to remove as the system is running along.

Q. So did you form an opinion regarding the validity of Claim 2?

A. Yeah. When you take this patent and combine it with any of the other references we just talked about, it would invalidate Claim 2.

Q. And I think you got one more reference that shows this element of Claim 2, and that's another Nemes patent?

A. Yes, in this patent Dr. Nemes is comparing the system load. That's how busy the system is at the particular time and deciding whether to remove all of records or none of the records.

Q. So what is your opinion -- why is it your opinion that you can combine the Apple patent or the Nemes '663 patent with the prior art, whether that is the Linux or the '495 patent or the NRL code to make Claim 2 obvious?

A. Well, if somebody wanted to limit how much time was spent in the garbage collection, they would just take either of these references and combine them.

* * * *

Q. Does the Linux prior art, the NRL code, or the '495 patent in combination with prior art references -- we have to use combination -- such as the [Apple patent] or the '663 patent, invalidate Claim 2 of the '120 patent?

A. Yes, sir, it does.

(T.T. 5/9/11 at 204:2-6, 204:9-205:10, 205:25-206:4; *see id.* at 183:6-11, 187:25-188:18).

Bedrock did not dispute or offer any contradictory evidence.

B. The Apple Patent and/or the '663 Patent Combined with the Claim 1 Prior Art Render Claim 2 Obvious

The Apple patent describes a virtual memory manager that dynamically determines (or defines on-the-fly) the number of records to remove as the system runs. (Ex. 5 at 7:2-7, 14-37; 8:44-46; 9:11-14; T.T. 204:2-15).²¹ The Apple patent derives a formula for this calculation. (Ex. 5 at 8:29-32).

²¹ When a process or thread is created or deleted, or in response to some regularly occurring event, a number of entries in the page table are examined to identify and remove expired entries. (Ex. 5 at 7:2-7; 8:44-46; 9:11-14).

Moreover, the Apple patent discloses that “[a]ny other suitable approach can be employed to determine the number of entries to be examined during each step of the sweeping process,” and that this maximum number “might vary from one step to the next.” (Ex. 5 at 7:38-46). By placing a limit on the number of entries to examine on any particular sweep, the system disclosed in the Apple patent also determines a maximum number of entries that will be removed, because the system cannot remove more entries than it examines on any particular sweep. The ’663 patent also discloses a method for dynamically determining the number of records to remove. (Ex. 6).

As Williams confirmed, it would have been obvious to one skilled in the art that this method and means of dynamically determining a maximum number could be combined with any system doing garbage collection. (T.T. 5/9/11 at 204:2-205:10). Furthermore, the motivation existed to combine this art. Because the removal of expired records can be an expensive process, placing a hard limit on the number of removals, or garbage collection, occurring at any particular time would have been desirable in any number of the prior art systems and methods described above. Applying this technique described in Apple or the ’663 patents to any of the prior art systems set forth below was well within the ability of one of ordinary skill in the art in January 1997 and renders claim 2 obvious. (T.T. 5/9/11 at 204:2-205:10; 205:25-206:7). Bedrock has presented no evidence or testimony to refute this combination.

1. ‘95 Linux Code in Combination with Apple Patent Renders Claim 2 Obvious

The ‘95 Linux Code contains a conditional statement that determines whether or not to delete an expired record based on a comparison of the record’s expiration time with the current time. (Ex. 1 at 1369-80). In response, Jones summarily states: “there is no dynamic determination in here of how many records to remove. It removes all the records it sees in this

loop.” (T.T. 5/9/11 at 287:25-288:2). Bedrock’s position is insufficient and cannot survive JMOL because it lacks merit and is directly at odds with its position that it took at trial that the Candidate’s code decision to delete one or no records constitutes a dynamic determination.

Furthermore, the ‘95 Linux Code in combination with the Apple patent renders claim 2 obvious. As described above, the Apple patent discloses a method of managing the physical memory allocated to processes in a computer system. (Ex. 5 at 7:38-46). A person of ordinary skill in the art would recognize that this simple ratio could be applied to dynamically setting the number of removals in the *ip_rt_hash_table* processing by, for example, computing the ratio of the number of entries in the *ip_rt_hash_table* to the number of active processes.

Bedrock does not rebut Yahoo!’s testimony that the ‘95 Linux Code in combination with the Apple patent invalidates claim 2. As such, claim 2 of the patent is invalid.

2. NRL Code in Combination with Apple Patent or ‘663 Patent Renders Claim 2 Obvious

As set forth above, the combination of the *key_acquire* and *key_search* portion of the *key.c* file in the NRL code render claim 1 obvious. (See T.T. 4/29/11 a.m. 139:14-23). Further, the Apple patent and/or the ‘663 patent specifically describe dynamic code determining the maximum number of records to remove. (Ex. 5; Ex. 6). As such, the NRL code in combination with the Apple patent or the ‘663 patent, as described above, renders claim 2 obvious. (T.T. 5/9/11 at 204:16-205:10; 205:25-206:7; see Ex. 2 at l. 1445-57; Ex. 5.) Bedrock does not refute this combination. For this reason also, claim 2 is invalid.

3. The ‘495 Patent in Combination with External Chaining Prior Art and Apple Patent or the ‘663 Patent Render Claim 2 Obvious

The ‘495 patent with the prior art reference of Knuth, Kruse or Stubbs which describes external chaining and linked lists as an alternative collision resolution in combination with the Apple patent, and/or the ‘663 patent renders claim 2 obvious. One of ordinary skill in the art

would have been motivated to combine the dynamic determination of any of these pieces of art with the '495 patent in order to limit the amount of cleanup done by the record search means. (T.T. 5/9/11 at 204:16-205:10; 205:25-206:7). Finally, as Bedrock also failed to refute these references, for this reason also, claim 2 of the '120 patent is rendered obvious. Consequently, no reasonable jury could find that claim 2 was not invalid; therefore, this Court should grant JMOL with respect to claim 2 of the '120 patent.

III. SECONDARY CONSIDERATIONS CANNOT SAVE THE '120 PATENT

The record is devoid of any evidence of secondary considerations, including commercial success, long-felt need, or the failure of others to make the invention, that may tend to prove the non-obviousness of the claimed inventions. *See Graham*, 383 U.S. at 17-18. Bedrock did not even pretend to address the issue. The commercial success of Linux operating systems as a whole are attributable to the contributions of the Linux community and to the numerous other advantages and facets of Linux contained in the millions of lines of source code and not the forty lines of the accused code. Indeed, Yahoo! did not choose the Linux operating system because of any speed or efficiency gained by the accused code. (T.T. 4/29/11 a.m. at 114:23-116:1).²² Nor did Yahoo! choose Linux because the accused code was known to prevent denial of service attacks. (T.T. 4/29/11 a.m. at 169:21-170:3). Instead, Yahoo! migrated to the Linux operating system because of its acquisition of other companies that used Linux, such as Inktomi and Overture in 2003, not because of the accused code. (T.T. 4/29/11 a.m. at 174:6-175:13; *see Ex. 23*).

Commercial success “may have relevancy to the overall obviousness determination, but a nexus must exist between the commercial success and the claimed invention.” *Tokai Corp. v.*

²² Yahoo!’s speed and efficiency are attributed to a number of other factors—none of which have to do with operating systems. (T.T. 4/29/11 a.m. at 93:14-98:3).

Easton Enters., Inc., --- F.3d ----, 2011 WL 308370, at *8 (Fed. Cir. Jan. 31, 2011) (internal quotations, citations omitted). Bedrock proffered none of its own commercial success, no evidence of any press, speeches or awards for the invention, no evidence of long-felt need, and no evidence that others failed to make the invention. Indeed, the evidence is to the contrary—both the ‘95 Linux Code and the NRL code were already in the marketplace two years prior to the ’120 patent application. Furthermore, there is no evidence from which one could reasonably infer a nexus between Yahoo!’s cost savings and use of the alleged patented invention. *See id.* at *9 (holding that sales data was not pertinent to the court’s obviousness determination when patent holder proffered no evidence to establish a nexus). Even assuming the existence of a nexus, secondary considerations cannot overcome the strong evidence of obviousness. *AdvanceMe*, 509 F. Supp. 2d at 625 (commercial success in practicing claimed methods could not overcome evidence of obviousness). “Granting a patent monopoly to this technological advance that would have occurred in the ordinary course without real innovation retards progress and deprives prior inventions of their value.” *Id.* Accordingly, a reasonable jury only could find—as should this Court—that claims 1 and 2 of the ’120 patent are obvious over the prior art.

CONCLUSION

Yahoo! respectfully requests that the Court grant its Renewed Motion for Judgment as a Matter of Law Regarding Invalidity, and enter judgment that claims 1 and 2 of the ’120 patent are invalid under 35 U.S.C. §§ 102(a), (b), (g)(2) and/or 103(a).

Respectfully submitted,

/s/ Jennifer H. Doan

Jennifer Doan
Texas Bar No. 08809050
jdoan@haltomdoan.com
J. Scott Andrews
Texas Bar No. 24064823
sandrews@haltomdoan.com
HALTOM & DOAN
Crown Executive Center, Suite 100
6500 Summerhill Rd.
Texarkana, Texas 75503
Tel: 903.255.1002
Fax: 903.255.0800

Fay E. Morisseau (Texas Bar No. 14460750)
fmorisseau@mwe.com
John C. Low (Texas Bar No. 24050960)
jlow@mwe.com
MCDERMOTT WILL & EMERY LLP
1000 Louisiana, Suite 1300
Houston, TX 77002
Tel: 713.653.1700
Fax: 713.739.7592

Yar R. Chaikovsky
ychaikovsky@mwe.com
John A. Lee
jlee@mwe.com
Bryan K. James
bjames@mwe.com
MCDERMOTT WILL & EMERY LLP
275 Middlefield Rd., Suite 100
Menlo Park, CA 94025
Tel: 650.815.7400
Fax: 650.815.7401

Christopher D. Bright
cbright@mwe.com
MCDERMOTT WILL & EMERY LLP
18191 Von Karman Ave, Ste. 500
Irvine, California 92612
Tel: 949.757.7178
Fax: 949.851.9348

Natalie A. Bennett
nbennett@mwe.com
MCDERMOTT WILL & EMERY LLP
227 West Monroe
Chicago, IL 60614
Tel: 312.984.7631
Fax: 312.984.7700

ATTORNEYS FOR YAHOO!, INC.

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing document was filed electronically in compliance with Local Rule CV-5(a). All other counsel of record not deemed to have consented to electronic service were served with a true and correct copy of the foregoing by certified mail, return receipt requested, on this the 6th day of June, 2011.

/s/ Jennifer H. Doan
Jennifer H. Doan

CERTIFICATE OF CONFERENCE

Counsel for Yahoo! and counsel for Plaintiff conferred, and Plaintiff opposes the relief requested in this motion.

/s/ Jennifer H. Doan
Jennifer H. Doan