

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LENOVO (UNITED STATES) INC.
and MOTOROLA MOBILITY LLC,
Petitioners,

v.

HEADWATER RESEARCH LLC,
Patent Owner.

IPR2024-01181
Patent 10,749,700

**DECLARATION OF ERIK DE LA IGLESIA IN SUPPORT OF PATENT
OWNER'S RESPONSE TO THE PETITION**

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

Executed on June 3, 2025, at Mountain View, California.



Erik de la Iglesia

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Background and Qualifications	2
III.	Legal Principles	6
A.	Claim Construction	6
B.	Burden of Proof	6
C.	Anticipation	6
D.	Obviousness	7
IV.	Person of Ordinary Skill in the Art.....	8
V.	Overview of the '700 Patent	9
VI.	Claim Construction	12
VII.	Overview of the Prior Art	12
A.	Rao	12
B.	Scahill	16
VIII.	Dr. Bims' claim mapping does not satisfy claim 1's requirements.....	18
IX.	Dr. Bims additionally fails to motivate and provide a reasonable expectation of success for his high-level proposal of incorporating Scahill's application scheduler functionality into Rao's packet filter.	22

I, Erik de la Iglesia, hereby declare as follows:

I. Introduction

1. I am over the age of eighteen (18) years and otherwise competent to make this declaration.

2. I have been retained as an expert witness on behalf of Patent Owner for the above-captioned *inter partes* review (“IPR”). I understand that the petition for *inter partes* review involves U.S. Patent No. 70,749,700 (“the ’700 patent”).

3. I make this declaration based on my personal knowledge, educational background and training, consideration of the materials I discuss herein, and my expert opinions.

4. My work on this matter is being billed at my normal hourly consulting rate of \$750, with reimbursement for actual expenses. My compensation is not related to the outcome of any proceeding involving the ’700 patent. I have no personal interest in the outcome of the case.

5. In preparing this Declaration, I have reviewed and considered the ’700 patent, the ’700 patent’s prosecution history, the Petition and prior art references submitted with the Petition, the declaration submitted by Dr. Bims in this proceeding, Dr. Bims’ deposition transcript, and each document cited in my declaration.

II. Background and Qualifications

6. My qualifications for forming the opinions given in this declaration are summarized here and are addressed more fully in my curriculum vitae, which is submitted as Exhibit 2002.

7. I hold a Bachelor of Science degree in Electrical Engineering from the University of Florida, and a Master of Science degree in Electrical Engineering from Stanford University. While at Stanford, I was an NSF Graduate Research Fellow. My graduate focus included VLSI CMOS architecture, high-speed circuit design, computer networks and protocols, and artificial intelligence. I have worked in several computer and electrical engineering related fields for over 25 years resulting in 68 issued U.S. patents that have been cited over 2850 times in USPTO patent applications. Since 2003, I have served in a number of chief architect and founder roles of companies that have been acquired or gone public. My experiences include the design or implementation of networks and network-based communication systems similar to the technology of the '700 patent, including systems for controlling network usage.

8. Between December 1997 and November 1999, I was a circuit design engineer at Intel Corporation within the Mobile and Handheld Products Group. My responsibilities included working on several aspects of mobile processors and chipsets including clock and power distribution, dynamic clock frequency

selection, and the circuit design, device simulation, and validation of such features. Until March of 2000, I worked on similar functionality within the Microprocessor Products Group for the Itanium 3 and 4 products.

9. My work in power distribution included thermal modeling and predictive control of silicon devices and their packaging such as multi-tiered cooling systems for embedded, mobile, and desktop environments. Such systems included multiple tiers of thermal sensors, active and passive thermal control systems, and the feedback and analytic firmware and software control. My models were part of the development of the Thermal Design Power (TDP) metrics used by OEM and system designers to plan thermal control solutions for consumer and industrial products.

10. Between March 2000 and September 2003, I worked as a logic designer and architect for WebStacks and, through acquisition, Extreme Networks. WebStacks built a fully hardware-based TCP and HTTP processing stack for network processing functions including load balancing, proxy processing, and content rewriting. I was responsible for the architecture and design of the HTTP processing engine including header extraction and processing.

11. Between August 2003 and August 2007, I worked as Founder, Chief Architect and Director of Engineering for Reconnex, later acquired by McAfee. Reconnex built a gigabit line-rate network security analyzer capable of

classification, heuristic analysis, and policy enforcement on arbitrary protocols and content types. The Reconnex product captured and analyzed content transmitted over the internet using HTTP and other protocols, including storage protocols. Design of the Reconnex system was based on a modified and customized version of the Linux kernel which included specifically designed driver source code and interfacing between the application and kernel domains. Reconnex technology was employed in the audit and compliance verticals based on its ability to quickly analyze and classify the content of transmissions and interact with protocols and communications crossing corporate and departmental boundaries.

12. Between August 2007 and September 2008, I founded Strangways, a company focused on addressing messaging security for webmail and internet messaging protocols. Products and technology developed by Strangways were sold to SendMail Inc., Iron Port (Cisco), and other companies. The Strangways product utilized a Squid caching server and ICAP content rewrite server to translate HTTP to SMTP.

13. Between September 2008 and April 2014, I worked as Chief Architect for Gridiron Systems and, through acquisition, Senior Director of Technology for Violin Memory. I held the Senior Director position with Violin Memory both before and after the company's IPO. My responsibilities included architecture and

design of flash media controllers, adaptive caching design, and machine learning for application storage behavior analytics.

14. The Gridiron product was a storage accelerator used in many analytic environments such as big data architectures with constant ingest and processing of financial, consumer and IoT data. In my role as Chief Architect, I often worked with customers to implement and accelerate their analytics environments and spoke on related topics at trade shows and industry user groups. During product development, I worked with the data science team at Shopzilla to improve database and storage performance for real-time shopping recommendation and user tracking for advertising.

15. My work at Webstacks and Reconnex related directly to packet processing including concepts such as priority and rules based processing relevant to the current proceeding. Approximately two dozen of my issued patents relate directly to these concepts.

16. Additional information regarding my employment history and prior expert testimony is included in Exhibit 2002. As reflected in Exhibit 2002, I have consulted on several matters involving intellectual property disputes including instances of alleged patent infringement requiring analysis of source code, reviewing and analyzing patents, writing expert reports, and testifying in court. I have testified at deposition and at trial in a number of these cases.

III. Legal Principles

A. Claim Construction

17. I understand that the first step in performing a validity analysis of the patent claims is to interpret the meaning and scope of the claims by construing the terms and phrases found in those claims. I understand that the appropriate construction of a claim term is its ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent and the prosecution history.

18. I understand that standard for claim construction in an *inter partes* review is the same standard as is applied in district court proceedings.

19. I understand that a determination of the meaning and scope of the claims is a matter of law. I have been informed that to determine the meaning of the claims, one should consider the intrinsic evidence, which includes the patent's claims, written description, and prosecution history.

B. Burden of Proof

20. I understand that in an *inter partes* review, the petitioner has the burden of proving unpatentability by a preponderance of the evidence.

C. Anticipation

21. I have been instructed by counsel and understand that a reference is anticipated if a single prior art reference discloses each and every claim element,

either explicitly or inherently, as arranged in the same way as in the claim. I understand that where even one claim element is not disclosed in a reference, a contention of anticipation fails.

22. I further understand that when a reference fails to explicitly disclose a claim element, that reference inherently discloses that element only if the reference must *necessarily* (or “inherently”) include the undisclosed claim element.

D. Obviousness

23. I have been instructed by counsel and understand that a combination of prior art references may render a claim obvious if, at the time of the invention, a person of ordinary skill in the art would have selected and combined those prior-art elements in the normal course of research and development to yield the claimed invention.

24. I understand that in an obviousness analysis, one should consider the *Graham* factors, including: the scope and content of the prior art; the differences between the claimed inventions and the prior art; and the level of ordinary skill in the art. I further understand the obviousness analysis is to be performed on a claim-by-claim basis. I understand that a person of ordinary skill in the art is a person of ordinary creativity, not an automaton.

25. I have been instructed by counsel and understand that obviousness requires more than a mere showing that the prior art includes separate references

covering each separate limitation in a claim under examination. I understand obviousness requires the additional showing that a person of ordinary skill at the time of the invention would have been motivated to combine those references in a manner that would include all limitations of the challenged claim, and, in making that combination, a person of ordinary skill in the art would have had a reasonable expectation of success.

26. I also understand that an obviousness analysis must be conducted with awareness of the distortion caused by hindsight bias and with caution of arguments reliant upon *ex post* reasoning. For instance, I understand that when considering obviousness, I should put myself in the position of a person of ordinary skill in the field at the time of the invention, rather than considering new information that is known today, but was not known before the priority date of the challenged patent.

IV. Person of Ordinary Skill in the Art

27. Dr. Bims states that a POSITA in the field of the '700 patent as of January 28, 2009 “would have (1) held at least a bachelor’s degree in computer science, electrical engineering, or a related field, and (2) three to five years of experience with networking, power consumption of networked computing devices, and/or wireless digital communications systems.” Ex. 1003 ¶25 Dr. Bims further states that “[m]ore education... would compensate for fewer years of work experience.” *Id.*

28. I note that Dr. Bims acknowledges that his proposed level of skill in the art can be satisfied by the stated bachelor's degree plus only three years of relevant experience. In my opinion, this the level of ordinary skill in the art, and inserting a range (i.e., "three to five years") introduces additional confusion, because obviously a person with three years' experience has a different level of skill than a person with five years' experience. For purposes of this declaration, I adopt Dr. Bims' proposed level of skill in the art to the extent it requires only three years of experience in addition to a bachelor's degree—in my opinion, a person having five years of experience in addition to a bachelor's degree would have slightly more than ordinary skill in the art. (My opinions would be the same regardless of which level of skill in the art is applied, but the level of skill in the art is at least somewhat relevant here given the high level of complexity of the modification Dr. Bims is proposing, as explained in further detail below.) I also note that the relevant experience could be either educational or work experience.

29. I have at least this level of skill in the art (even if five years of experience were required in addition to a bachelor's degree), and have had such level of skill since before the priority date of the '700 patent.

V. Overview of the '700 Patent

30. At a high level, the challenged claims of the '700 Patent (claims 1-20) are directed to a device that is capable of associating "network access policy

controls with... [a] plurality of executable applications,” wherein the applications “request an execution time according to the corresponding network access policy controls,” and wherein the device includes “a launch manager to schedule corresponding execution times for [the requesting] applications” according to the policy controls.

31. Claim 1 is the only independent claim of the ’700 patent, and it recites:

1[pre]. A wireless end-user device comprising:

1[a] at least one wireless modem to provide network access for the wireless end-user device;

1[b] an activity classifier to, for each of a plurality of executable applications stored on the wireless end-user device, dynamically associate one or more corresponding network access policy controls with each of the plurality of executable applications;

1[c] an Application Programming Interface (API) accessible to each of the plurality of executable applications, the API callable by each of the executable applications to request an execution time according to the corresponding network access policy controls for that application; and

1[d] a launch manager to schedule corresponding execution times for applications requesting an execution time through the API, the launch manager selecting corresponding execution times, for the applications requesting execution

time, according to the respective network access policy controls for each such application.

Ex. 1001, claim 1.

32. As shown in claim 1 above, the application requests execution time by calling an API, and the requested execution time is scheduled according to the policy controls.

33. Relevant to my analysis in this declaration, a POSITA would understand (and the '700 patent makes clear) that there is a distinction between the requested and scheduled “execution of an application” on the one hand, and communication generated as a result of execution on the other hand. *See, e.g.*, Ex. 1001 at 75:4-7 (distinguishing between “an application executing on the device” and communications sent or received from the executing application); *id.* at 102:64-103:2 (“The application can be referred to as ‘running’ on the device or as being ‘executed’ by the device in accordance with known uses of those terms.”); *id.* at Claims 1, 11, 15, 17, 18, and 19 (referring to “executable applications”); *id.* at 35:42-51 (“In some embodiments, *at or after execution of an application that is determined to require network service usage...*”).¹ I also note that, consistent with the teachings of the '700 patent, a POSITA would understand that applications

¹ All emphases within this declaration are added unless I have specified otherwise.

generally utilize “application messaging layer requests (e.g., socket open/send requests)” (Ex. 1001, 93:2-5) which may ultimately result in packet generation, but that the “application level/layer” is distinct from the “packet level/layer.” *See id.* at 10:16-19.

VI. Claim Construction

34. For purposes of this declaration, I apply the plain and ordinary meaning of claim terms to a POSITA in light of the intrinsic record.

VII. Overview of the Prior Art

A. Rao

35. Rao relates to “prioritization of network communications.” *See* Ex. 1005 at Title. Rao prioritizes network communications by having a packet filter 322 “determine what action to take against packets,” including dropping those packets or placing them into a queue for transmission based on priority:

The filter 322 may use a filter table for filtering packets. The filtering table is used to determine what action to take against packets, such as packets intercepted by the packet capture mechanism 365. The filter 322 can inspect the contents of the packets, such as routing information, to determine the action to take based on the filtering table. In some embodiments, the filter 322 may drop or accept the network packets depending on their content. In other embodiments, the filter 322 may route the network packets

to the agent 326 based on the packets content and/or the filtering table. The filter table may also be used to ensure that unwanted packets are discarded.

Ex. 1005, [0102]; *id.* at [0191] (“Based on the indicated priorities for applications 338 a-338 n, in some embodiments, *the filter 322 may place, arrange, or coordinate network packets into queues 540a-540n* in support of the priorities.”); *see also* Ex. 1003 ¶¶50-555 (the Petition’s overview of Rao, which explains that Rao’s prioritization policies are implemented by queuing network packets according to their priority).

36. Rao’s packet filter is shown, for example, in Rao’s FIG. 1C., which indicates that Rao’s filter 322 exists outside of application 338 (which is an application that requests transmission of packets which are ultimately filtered and prioritized by filter 322).

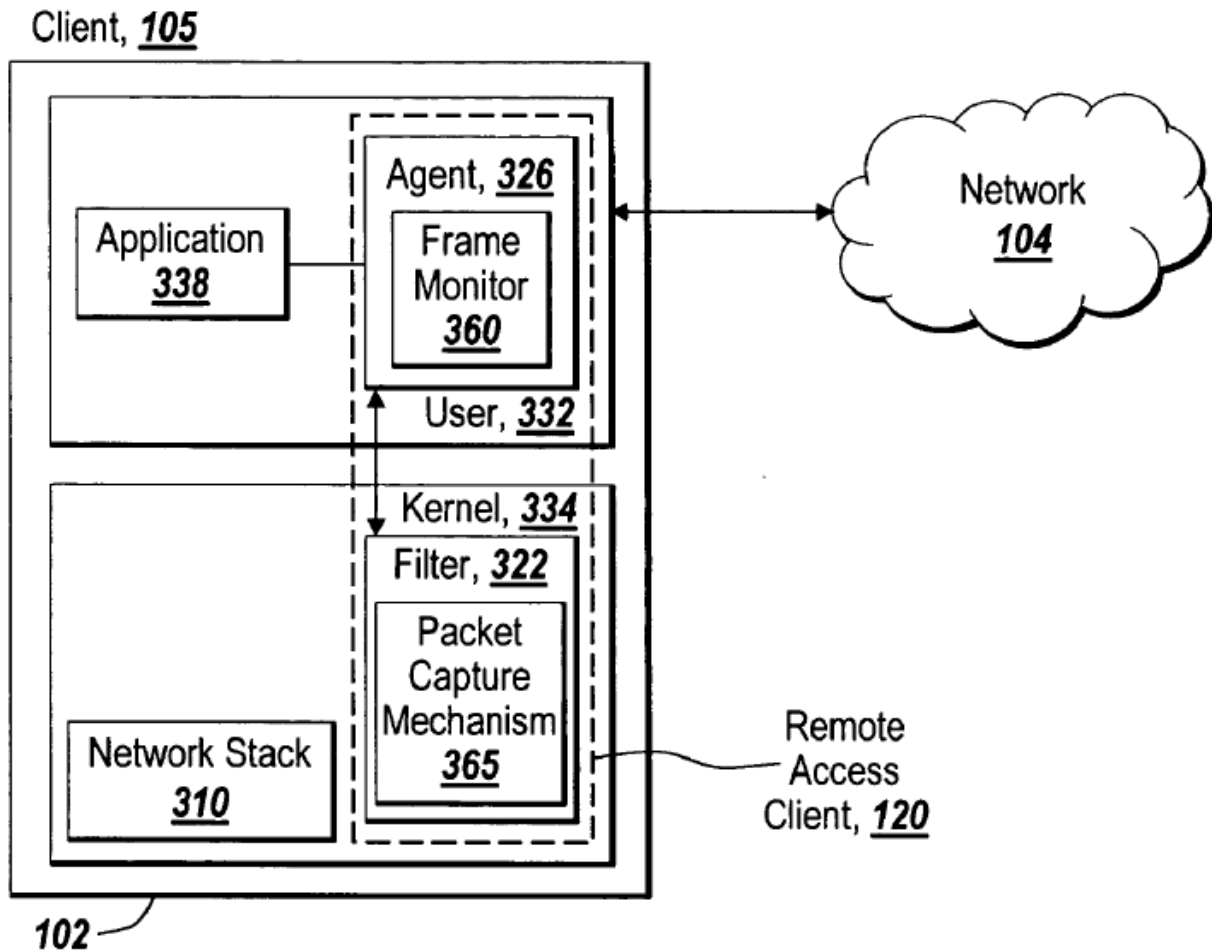


Fig. 1C

Ex. 1005, FIG. 1C.

37. Rao also discloses, in Figure 5B (reproduced below) “a flow diagram depicting one embodiment of the steps performed to provide client-side application-aware prioritization.” Ex. 1005, [0071].

550

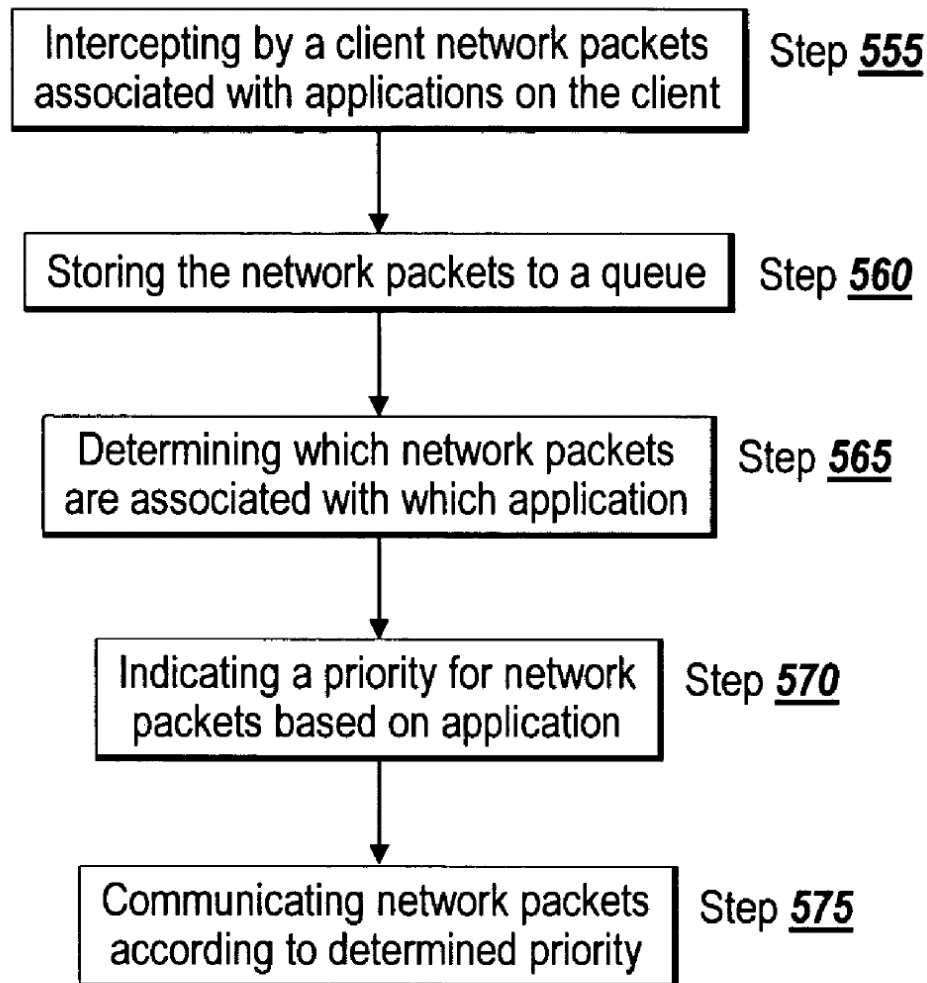


Fig. 5B

Ex. 1005, FIG. 5B.

38. I note that Dr. Bims highlights FIG. 5B as an example of Rao's prioritization scheme, and acknowledges that this figure ultimately prioritizes

network packets through a queueing system, and transmits prioritized packets accordingly. Ex. 1003 ¶¶54-55.

B. Scahill

39. Unlike Rao's filter 322 which influences network communication by placing network packets into queues, Scahill's application scheduler 24 assigns execution time to applications so that they can generate network communications in the first place. *See, e.g.*, Ex. 1003 ¶56. ("Scahill discloses a method of coordinating the execution time of a plurality of applications requiring a network connection that are all hosted on the same mobile device."). This is accomplished, not through a queueing system, but by a logic structure in which task pre-requisites and dependencies are evaluated before allowing any given task to proceed, as illustrated in Scahill's Figure 3 (reproduced below).

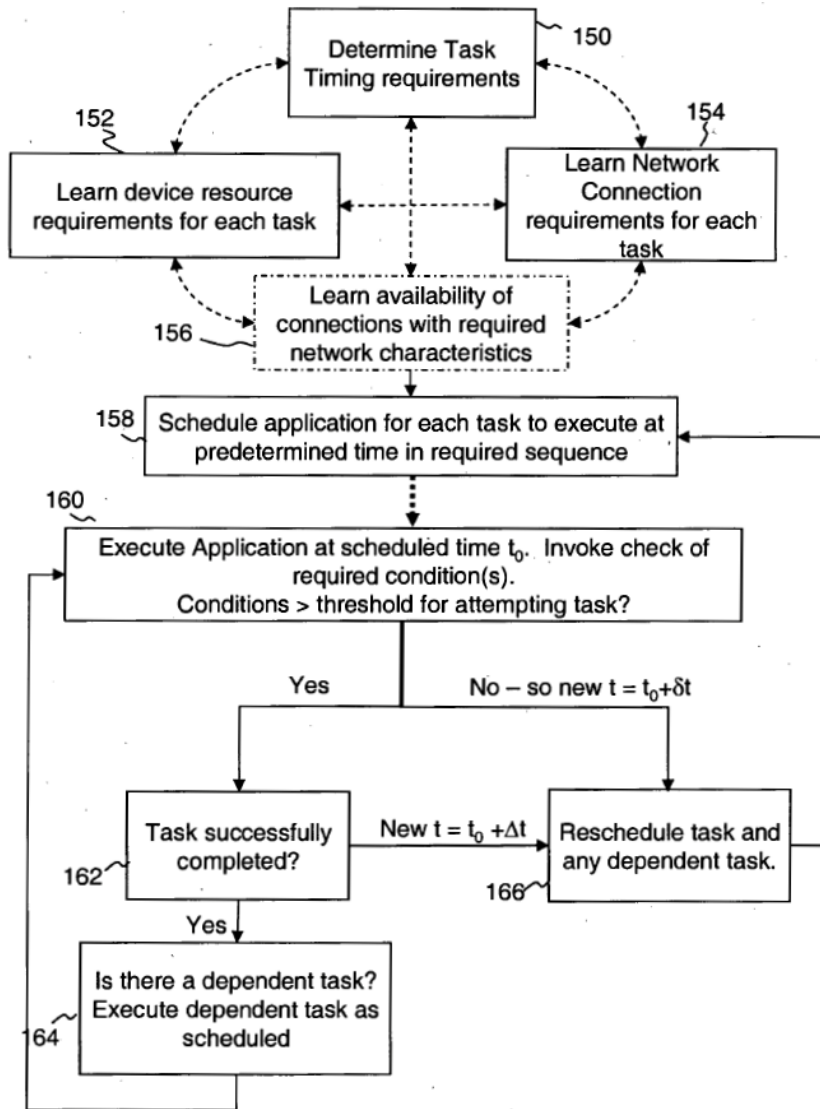


FIGURE 3

Ex. 1008, FIG. 3.

40. In my opinion, Scahill does not teach or suggest that its application scheduler would be used to schedule the timing of transmissions of any individual packet, as opposed to scheduling when a particular application task (which may ultimately generate packets) is executed. As I explain throughout this declaration,

application scheduling and making packet filtering decisions are significantly different endeavors.

41. Scahill further discloses that what it describes as “application scheduler 24” is actually a *plurality* of application schedulers, one for each application that requires scheduling. Ex. 1008, 17:14-17 (indicating that “[e]ach software client software library 14a,b comprises a plurality of library components, some specific to the respective client application 12a,12b); FIG. 1B (indicating that application scheduler 24 is one of these “[a]pplication specific library component[s]”); Ex. 2003 at 73:14-75:10 (Dr. Bims testifying that “application scheduler 24 is actually an application-specific application scheduler such as 24a corresponding to application 12a,” and “that would also be true for application 12b,” and “same thing for application 12c[,] [i]t would have an application scheduler 24c”).

VIII. Dr. Bims’ claim mapping does not satisfy claim 1’s requirements.

42. All challenged claims require a “launch manager to schedule corresponding execution times, for the applications requesting execution time” as recited by claim limitation 1[d]. Ex. 1001 at 107:47-52.

43. Dr. Bims is explicit that in his proposed combination, Rao’s packet filter 322 would be the “launch manager.” Ex. 1003 ¶86 (Dr. Bims: “Rao discloses that ‘filter 322’ (i.e., the claimed ‘launch manager’) arranges network

communications from the applications into a queue according to their respective policies.”). And although his theory is light on specifics, Dr. Bims also states that Rao’s filter 322 would be modified to incorporate the functionality of Scahill’s application schedulers 24. *See* Ex. 1003 ¶¶86-88 (not specifying exactly how Rao’s filter 322 would be modified); *see also* Ex. 1003 ¶¶60-72 (providing only ambiguous details stating at a high level that Scahill’s application scheduler functionality be implemented in Rao’s system, but not specifying precisely how this would be implemented).

44. Other than blanket assertions that limitation 1[d] is satisfied or assertions that merely paraphrase the claim language with no detail as to how the combined system would satisfy that language (e.g., the first and last sentences of Dr. Bim’s discussion of limitation 1[d]), Dr. Bims’ only real explanation *why* 1[d] would be satisfied in his combination is an assertion that Rao’s filter 322 as modified would “assign[] execution times” to “**network communications**” in the proposed combination. Ex. 1003 ¶88 (“In Rao/Scahill’s architecture, filter 322 coordinates network communications into queues using Rao’s policies, and the prioritized network communications would be assigned execution times by filter 322, implemented to incorporate the scheduling functionality of Scahill’s application scheduler 24, taking into consideration Scahill’s scheduling requests and other network factors.”). As noted above, Dr. Bims does not explain *how*

network packets would be organized into queues by “incorporat[ing] the scheduling functionality of Scahill’s application scheduler 24.” I also note that Dr. Bims is clear that by “network communications” he is referring to “network packets (Ex. 1003 ¶88 (“Once queued, remote access client 120 communicates the network packets.”)), and this is consistent with Rao’s teachings which are focused on queuing and prioritizing network packets. I thus understand Dr. Bims to allege that “assign[ing] execution times” to *network packets* constitutes “selecting corresponding execution times for applications.” See Ex. 1003 ¶88.

45. I further note that Dr. Bims’ usage of the concept of assigning “execution time” to *packets* is highly unconventional, and a POSITA would not refer to execution time for packets in the context of systems such as Scahill and Rao. Packets are *transmitted* or *received*, not *executed*. Certainly, Rao’s filter 322 would not be accurately described as assigning execution times to network packets. To the extent Rao’s filter 322 assigns particular times to packets at all, it would be transmission times being assigned.

46. I also note that, separate from the semantics of whether or not packets are “executed” or not, there is a clear distinction between *packets* and *application execution*. This is because applications execute, and *as a* result of that execution, packets may be generated. For example, an application while executing may request that the operating system transmit data across a network, and as a result of

that application execution, the operating system might generate and transmit packets. Rao itself makes this clear, by teaching that packets are “*generated or received for an application.*” See, e.g., Ex. 1005, [0003] (discussing “network packets *for these applications*” and the order in which the applications are processed); *id.* at [0024] & [0027] (“[T]he present invention relates to a method for transmitting *packets from an application* using an unreliable transport protocol”). Thus, there is a clear distinction between *application execution* (which may *cause* packets to be generated), and the resulting *packets* (which are the potential *result of* application execution).

47. As I explain in more detail in the following section, packets are significantly different than application execution; not only are they distinct concepts, but most applications are not even aware of the packets that are generated as a result of communication requests.

48. Contrary to a POSITA’s understanding that packets are generated or received *for* or on behalf of applications, the Dr. Bims contends that Rao’s queue of “network packets” is a “way[] to queue applications.” See Ex. 1003 ¶87 (quoting Rao’s teaching of queuing network packets, then alleging that “Rao provides additional examples of the many potential ways to queue applications,” and then providing yet another discussion in Rao of “packets [being] placed or arranged in the queue”). In other words, the Dr. Bims contends that queueing of *packets*

constitutes a queuing of *applications*. I disagree with this contention, as it conflates queuing packets with queuing application execution, and indeed ignores the fundamental distinction between *packets* and *application execution* altogether.

49. A POSITA would readily understand that scheduling transmission of individual packets does not constitute “selecting corresponding execution times, for the applications requesting execution time” as claim 1 requires. As noted above, packets are potentially generated *as a result of* application execution, such that scheduling the transmission of already-generated packets does not constitute assigning an execution time for an application as the claims require. A POSITA would further understand that the existence of packets to queue *presumes* that some task has already executed to request data transmission that resulted in those packets being generated, such that queuing of the resulting packets *cannot* mean scheduling application execution (because the execution has already occurred).

IX. Dr. Bims additionally fails to motivate and provide a reasonable expectation of success for his high-level proposal of incorporating Scahill’s application scheduler functionality into Rao’s packet filter.

50. As I explain in the preceding section, Dr. Bims’ theory regarding limitation 1[d] conflates the concept of a *packet* with the concept of *application execution*, and he sets forth no explanation how Rao’s filter 322 as modified would actually satisfy the claim requirements.

51. I further note that even at a high level, Dr. Bims’ proposal to incorporate application scheduler functionality into a packet filter would not have been obvious, because a POSITA wouldn’t have been motivated to make the combination and wouldn’t have reasonably expected to succeed in doing so.

52. As an initial matter, a POSITA would recognize that it would not make sense to implement an application scheduler into a packet filter for several reasons. First, these two components operate at entirely different timescales—packets would have been transmitted on the order of hundreds of times per second using the technologies available in the relevant timeframe, whereas application scheduling would occur at human-observable timescales. *See, e.g.*, Ex. 1008, 4:2-6 (discussing “several data transmissions to be performed within a 24 hour period” that need to be scheduled). In other words, packet filters make timing decisions that would typically be on the order of *thousands of times faster* than the timing decisions that application schedulers make, such that the design and timing considerations and the feasible operations and considerations of these two different components are very different. For instance, a 50 millisecond processing delay in determining when or whether a packet should be transmitted would be disastrous for efficiency in any given system (performing only 20 such operations per second per operating core) , whereas a 50 millisecond delay in determining when an application should be executed would be essentially imperceptible in its impact on

system operation and user experience. In my opinion, this significant difference in timing requirements and operative scope between packet filters and application schedulers accounts, at least in part, for the difference in complexity between the scheduling operations of Scahill's application schedulers 24 (e.g., accounting for task dependencies as I discuss further below), and the manner in which Rao's packet filter 322 operates (such that it prioritizes packets by placing them into simple queue structures).

53. Indeed, I note that the vast majority of instances (except the rare exception for applications that communicate by opening raw sockets, a concept not disclosed in either Rao or Scahill), applications are not even *aware* of what packets are generated on its behalf. Typical applications in systems such as Rao and Scahill simply request that data be transmitted (e.g., after requesting that a socket be opened, such as described at Ex. 1001 at 93:2-5), and lower-level system elements (such as the operating system) are responsible for generating, managing, and transmitting packets in order to effectuate the transmission of the data that the application has requested be transmitted. *See* Ex. 1001, 10:16-19 (distinguishing between the packet level and application level). Accordingly, applications in the context of Rao and Scahill would not be involved in packet scheduling, as the applications would not even be aware of the packets that would be transmitted.

54. In light of these significant differences between scheduling *packets* and scheduling *application execution*, it is my opinion that a POSITA would not be inclined to modify a packet filter to incorporate application scheduling functionality.

55. Dr. Bim's apparent motivation to incorporate *application scheduler* functionality into Rao's *packet filter* 322 appears to be based on a significant misunderstanding of what Rao's packet filter 322 actually does. As I previously noted, Dr. Bims states that Rao's filter 322 "queue[s] applications" (Ex. 1003 ¶87), even though he could not identify any disclosure in Rao of using its prioritization policies "to influence system operation other than by providing the result of prioritizing, ignoring, or not processing network packets." Ex. 2003 at 47:20-48:17.

56. Contrary to Dr. Bim's assertions, a POSITA would understand that Rao's filter 322 queues *packets* that may have been generated by applications, rather than actually queuing *applications* or application execution. And it is my opinion that there would be no motivation to incorporate application scheduler functionality into a packet filter as Dr. Bims proposes, given the significant differences between how these two distinct components operate.

57. The lack of evidence regarding motivation to make Dr. Bims' proposed combination is underscored by the fact that there is no evidence that

anyone has *ever* attempted to implement application scheduling functionality within a packet filter. I certainly am aware of no one having ever attempted to do so. It is also my opinion that this modification to Rao's filter 322 this would not be a predictable use of prior art elements according to their established functions, because application scheduling is far from an established function for a packet filter.

58. Further differences between Rao's filter 322 and Scahill's application schedulers 24 make Dr. Bim's proposed combination non-obvious. For example, Rao's filter 322 prioritizes packet transmission by placing packets into queues, which a POSITA would understand are linear data structures that place certain packets in front of others. In contrast, Scahill's application schedulers 24 do *not* use queues because they must account for dependencies between tasks.

59. In the context of Scahill, "a dependent task is a task whose execution by a client application 12a...c is dependent on the successful completion of another task by the same or a different client application 12a...c." Ex. 1008 at 20:18-20. An example dependency might be that the activity of "uploading an activities journal for various files" must be performed before another activity of "downloading all time-charged activities" can be performed. *See* Ex. 1008 at 21:25-28. Scahill teaches that "[i]n the event that a task has a task dependency, ...

the relevant application scheduler 24b,c will reschedule the execution time for [the] dependent task whenever the first task is rescheduled.” *Id.* at 20:20-23.

60. An illustration of the logic flow for Scahill’s application scheduling is shown in Scahill’s FIG. 3, reproduced below.

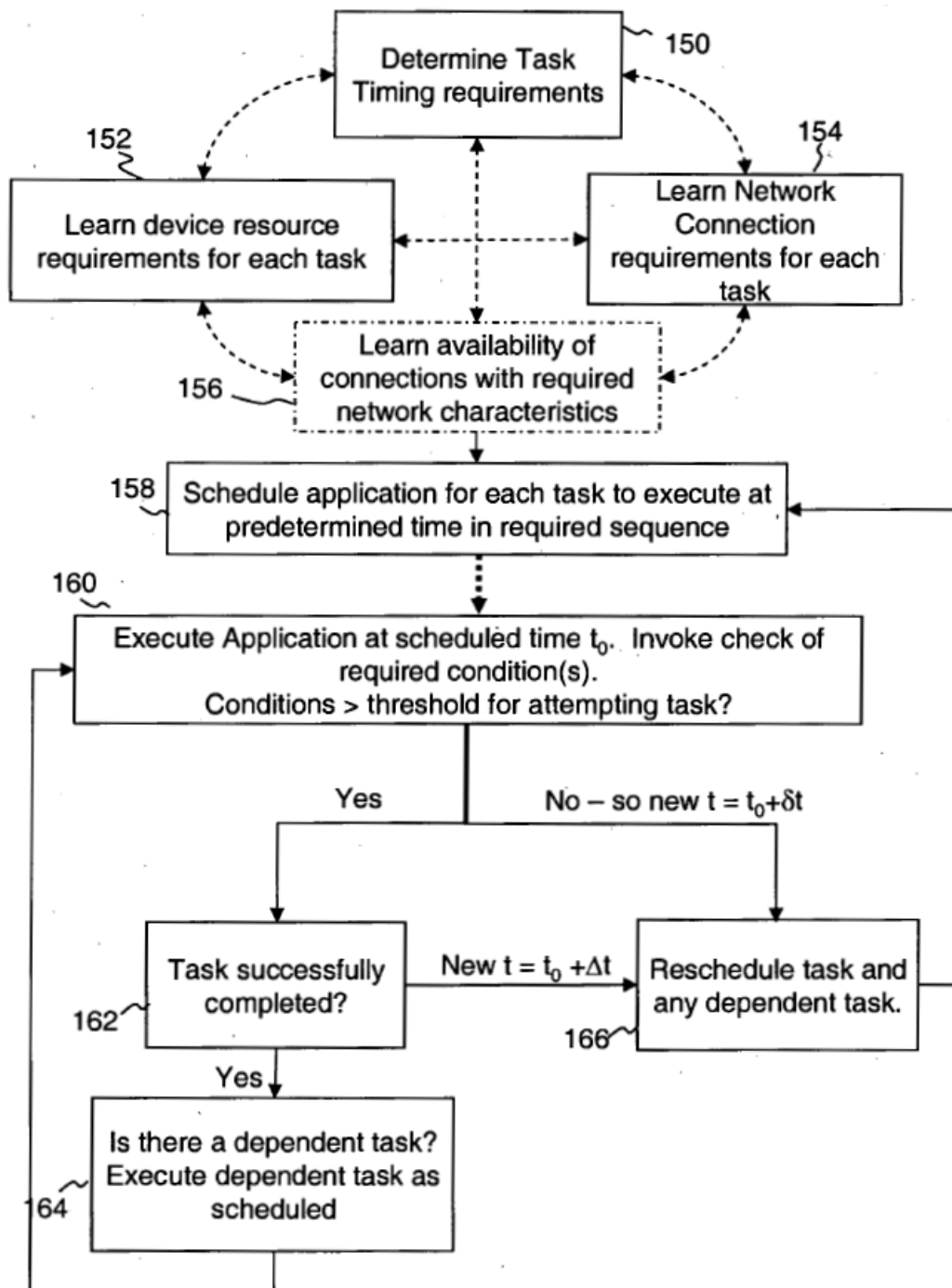


FIGURE 3

Ex. 1008, FIG. 3.

61. As shown in logic diagram above, even *after* an application's task has been scheduled for a particular time (in step 158), the system must "[i]nvoke [a] check of required condition(s)" such as whether the required dependencies have been satisfied. There is no indication that this complex logic would be (or could be) implemented in a queuing structure, much less by the type of queuing structure controlled by Rao's *packet* filter, and Dr. Bims has identified no modification to either filter 322 or application schedulers 24 to reconcile this discrepancy or explain why this modification would have been realizable (either conceptually or in practice) to a POSITA.

62. I also note that Dr. Bims' proposal of moving functionality from Scahill's application schedulers 24 into Rao's filter 322 is contrary to Scahill's teaching that each application is assigned its *own* application scheduler that is "specific to the respective client application 12a,12b." Ex. 1008, 17:14-17; Ex. 1008 at 32-34 (discussing how tasks "are scheduled... by the respective application schedulers 24a,b,c of each client application 12a...c"); Ex. 2003 at 73:14-75:10 (Dr. Bims testifying that "application scheduler 24 is actually an application-specific application scheduler such as 24a corresponding to application 12a," and "that would also be true for application 12b," and "same thing for application 12c[,] [i]t would have an application scheduler 24c").

63. Unlike in Scahill, where each application has its own application scheduler that operates within the application, Dr. Bims' proposal involves taking application scheduler functionality *outside* of the applications, and placing it within a single filter 322 which "is separate from the applications." Ex. 2003, 67:15-19 (Dr. Bims testifying that in his proposed implementation, "the filter resides in the remote access client, which is separate from the applications").

64. Dr. Bims, however, provides no motivation to modify Scahill's application schedulers (which Scahill teaches are specific to each application and operate within the applications themselves), such that there would only be a *single* component (filter 322), located *outside* of the applications, that incorporates that functionality. *See, e.g.*, Ex. 1003 ¶¶86-88 (not addressing this change, nor providing a motivation to make it). Nor does he provide any evidence that a POSITA would have expected to succeed in making this modification.

65. It is my opinion that a POSITA would have actually *avoided* making this proposed modification to Scahill's application schedulers due to the teachings of Scahill, because a POSITA would have recognized that "[t]here are a number of *application specific requirement(s)* which the application scheduler library component 14a,b of each client application 12a,b... must take into account" in performing its functionality. Ex. 1008, 24:14-17. Because requirements are specific to each individual application, a POSITA would understand it to be far more

sensible to leave application scheduling functionality within the application-specific scheduling components 24a, 24b, etc., rather than implementing them in a single component (such as filter 322) that is external to the relevant applications.

66. For each of the foregoing reasons, it is my opinion that a POSITA would not have been motivated to make the proposed modifications to Rao and Scahill, and even if they had somehow *wanted* to make those motivations they would not have expected to succeed given the complexity of Dr. Bims' proposed combination, and the significant differences between Rao's filter 322 and Scahill's application schedulers 24. Regardless of whether a POSITA would have had a bachelor's degree and three or five years of additional experience, these modifications would have been far beyond that level of skill in the art. In my opinion Dr. Bims' proposed modifications, to the extent possible to implement in a way that would have been viewed as satisfactory in terms of functionality and efficiency, would have required an extraordinary level of skill, even *assuming* that the implementing individual were for some reason motivated to mash two very different components together in the way Dr. Bim's alleges.