

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.

Case No. IPR2024-01181
U.S. Patent No. 10,749,700

**DECLARATION OF HARRY BIMS, Ph.D.
REGARDING U.S. PATENT NO. 10,749,700**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	SUMMARY OF OPINIONS	1
III.	BACKGROUND AND QUALIFICATIONS	2
	A. Background.....	2
	B. Compensation	7
IV.	MATERIALS CONSIDERED	8
V.	LEGAL STANDARDS.....	8
	A. Level of Ordinary Skill.....	9
	B. Claim Construction.....	10
	C. Anticipation under 35 U.S.C. § 102	11
	D. Obviousness under 35 U.S.C. § 103	11
	E. Secondary Considerations of Non-Obviousness	14
VI.	BACKGROUND AND TECHNOLOGY OF THE '700 PATENT.....	16
	A. Overview of the '700 Patent	16
	B. Claims of the '700 patent.....	17
VII.	ANALYSIS.....	17
	A. Ground 1: The Combination of <i>Rao</i> in View of <i>Scahill</i> Renders Claims 1, 2, 4, 5, 7-11, 13, and 15-17 Obvious	18
	1. Overview of the <i>Rao</i> Reference	18
	2. Overview of the <i>Scahill</i> Reference.....	25
	3. Introduction to Rationale to Modify <i>Rao</i> in view of <i>Scahill</i>	28

4.	Independent Claim 1.....	36
a.	1[pre]: “A wireless end-user device comprising:”	36
b.	1[a]: “at least one wireless modem to provide network access for the wireless end-user device;”	36
c.	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;”	37
d.	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”	41
e.	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.”	45
5.	Claim 2: “The wireless end-user device of claim 1, the activity classifier dynamically associating network access policy controls with each of the plurality of executable applications using different policies that apply to blocked activities, network access controlled activities, and network access non-controlled activities.”	47
6.	Claim 4: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity	

	based on a user's current interaction with that application.".....	49
7.	Claim 5: "The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on that application currently providing a foreground service."	51
8.	Claim 7: "The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network capacity demand as part of a request for an execution time."	51
9.	Claim 8: "The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network request as part of a request for an execution time."	53
10.	Claim 9: "The wireless end-user device of claim 8, wherein the network request specifies whether the network request specifies one of a roaming network or a non-roaming network as the connected network for the network request."	54
11.	Claim 10: "The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a scheduled time request as part of a request for an execution time."	57
12.	Claim 11: "The wireless end-user device of claim 1, further comprising a user interface allowing a user to specify one or more of the network access policy controls to be associated with a given one of the plurality of executable applications."	59
13.	Claim 13: "The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a device usage state."	59

14.	Claim 15: “The wireless end-user device of claim 1, wherein the launch manager receives a list identifying executable applications that have a launch control policy in effect.”	60
15.	Claim 16: “The wireless end-user device of claim 1, wherein the launch manager is integrated with an operating system broadcast intent manager.”	62
16.	Claim 17: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager deprioritizes one or more processes associated with a given one of the executable applications.”	64
B.	Ground 2: Claim 3 Is Obvious Over <i>Rao</i> in view of <i>Scahill</i> and <i>Oestvall</i>	65
1.	Overview of the <i>Oestvall</i> Reference.....	65
2.	Rationale for Modifying <i>Rao</i> in view of <i>Scahill</i> and <i>Oestvall</i>	66
3.	Claim 3: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access controlled activity or a blocked activity based on a lack of user interaction with that application for a period of time.”	68
C.	Ground 3: Claims 6, 14, 18, and 19 Are Obvious Over <i>Rao</i> in view of <i>Scahill</i> and <i>Montemurro</i>	69
1.	Overview of <i>Montemurro</i>	69
2.	Rationale for Modifying <i>Rao</i> in view of <i>Scahill</i> and <i>Montemurro</i>	71
3.	Claim 6: “The wireless end-user device of claim 1, wherein according to at least one of the network access policy controls, the launch manager delays an execution time for one or more of the applications until a	

	connection is made through the at least one wireless modem to an alternative network.”	74
4.	Claim 14: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a change in a current network to which the at least one wireless modem is providing network access.”	76
5.	Claim 18: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager controls the frequency at which a given one of the executable applications is allowed network access.”	77
6.	Claim 19: “The wireless end-user device of claim 1, wherein, based on an amount of network data communicated to or from a given one of the executable applications, the activity classifier dynamically changes at least one network access policy control associated with that given executable application.”	79
D.	Ground 4: Claims 12 and 20 Are Obvious Over <i>Rao</i> in view of <i>Scahill</i> and <i>Araujo</i>	81
1.	Overview of the <i>Araujo</i> Reference.....	81
2.	Rationale for Modifying <i>Rao</i> in view of <i>Scahill</i> and <i>Araujo</i>	83
3.	Claim 12: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a power state of the device.”	88
4.	Claim 20: “The wireless end-user device of claim 1, wherein the launch manager schedules corresponding	

execution times for applications to correspond with
power state changes of the at least one wireless modem.” 90

VIII. DECLARATION 92

I, Harry Bims, Ph.D., declare as follows:

I. INTRODUCTION

1. I have been asked to submit this declaration on behalf of Lenovo (United States) Inc. and Motorola Mobility LLC (“Petitioners”) in connection with a petition for *inter partes* review of U.S. Patent No. 10,749,700 (“the ’700 patent”). I have been told that the petition is being submitted to the Patent Trial and Appeal Board of the United States Patent and Trademark Office by the Petitioners.

2. Specifically, I have been retained as a technical expert by the Petitioners to study and provide my opinions on the technology claimed in, and the patentability or unpatentability of, claims 1-20 of the ’700 patent (“the Challenged Claims”). For purposes of this declaration, I was not asked to provide any opinions that are not expressed herein.

II. SUMMARY OF OPINIONS

3. This declaration is directed to the Challenged Claims of the ’700 patent, and sets forth the opinions I have formed, the conclusions I have reached, and the bases for each.

4. In forming my opinions, counsel for Petitioners told me to assume that the effective filing date of the ’700 patent is January 28, 2009.

5. Based on my experience, knowledge of the art as of the effective filing date of the ’700 patent, analysis of prior art references, and the understanding

of the claim terms in view of the specification that a person of ordinary skill in the art would have as of the effective filing date, it is my opinion that the Challenged Claims of the '700 patent are unpatentable as obvious over the prior art references discussed below.

III. BACKGROUND AND QUALIFICATIONS

A. Background

6. My educational background, career history, publications, and other relevant qualifications provided here are only a summary. My full curriculum vitae is attached to this declaration as Exhibit 1003.

7. Based on my education, research, and work experience, I am knowledgeable about the subject matter of the '700 patent and the related prior art.

8. I have worked extensively in the field of digital communications. I have studied telecommunications and systems engineering since approximately 1981. Further, I have over 30 years of industry experience in computer network design, including the design of hardware and software for computer communications in a wireless context. During this period, I have designed and implemented various products involving cellular communication technology.

9. I received a B.S. in Computer and Systems Engineering from Rensselaer Polytechnic Institute in 1985. In 1988, I received a M.S. in Electrical Engineering from Stanford University. In 1993, I received a Ph.D. in Electrical

Engineering, also from Stanford University. As a graduate student at Stanford University, I studied the principles of digital communications theory, including data modulation and demodulation, error checking and correction algorithms, and the architecture and design of semiconductor circuits used for digital communications. My Ph.D. thesis at Stanford addressed the application of trellis coding and precoding to a digital modulation system, and was titled “Trellis Coding for Multi-Level, Partial Response Continuous Phase Modulation with Precoding.”

10. After receiving my Ph.D. in 1993, I worked for Glenayre Technologies - Wireless Access Group, where I focused on hardware and software architecture and design, including inventing, designing, and building a patented computer system for real-time testing of two-way pagers and co-developing a wireless application protocol that included a CRC error checking algorithm. From 1999 to 2001, I was the Director of Engineering for Symmetry Communications, where I was responsible for managing the software development for 2.5G network equipment (the precursor to 4G network equipment) that handles communications to and from personal cellular telephones. The equipment products we developed (a “Serving GPRS Support Node” (or “SGSN”) and “Gateway GPRS Support Node” or (“GGSN”)) routed and managed cellular phone communications by relaying packets of voice, data, and video between the base station tower and the Internet,

regulating the network bandwidth allocated to different classes of cellular phone applications, and transferring control of cellular phone communications from one base station to the next as cellular phones move throughout a geographic area. I also held management responsibility for the Firmware, Hardware, Performance, and Systems Engineering Groups.

11. In January of 1999, I launched a technology consulting company, Protocomm Systems Inc., which focuses on the development of advanced wireless communications protocols and related software implementations for wireless product companies.

12. From 1999 to 2001, I was also the Director of Software Architecture for Symmetry Communications System. In this position, I was responsible for the software architecture for core SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node) products for the GPRS (Generic Packet Radio Services) market. I also held management responsibility for the Firmware, Hardware, Performance, and Systems Engineering Groups.

13. In 2001, I developed a business plan for building network infrastructure for 802.11 enterprise networks, and then later that year founded AirFlow Networks, Inc., where I invented and received over eleven patents on its core technology, which was based on the 802.11 wireless local area network specification.

14. I have been a member of the Institute of Electrical and Electronics Engineers (“IEEE”) for more than 28 years. The IEEE is an international organization of more than 400,000 engineers, and is the world’s largest technical professional society, with locations in more than 160 countries. I am currently a Senior Member of the IEEE, and serve in a variety of leadership roles, including as a reviewer and voter on draft amendments to technical standards such as 802.11 (Wi-Fi), and 802.3 (Ethernet) prior to publication by their 802 Committee. The IEEE 802 Committee is the standards development organization responsible for publishing protocol standards for Ethernet, Wi-Fi, and a variety of other wired and wireless protocols that are in commercial use globally.

15. I am currently an expert consultant for Protocomm Systems, LLC and Bims Laboratories, LLC, both of which I founded. I perform technical research in wireless technology standards, such as UMTS, LTE/4G, 5G, Wi-Fi, Bluetooth, and other network communication protocols. I am named as an inventor of twenty-three telecommunications related patents. One of those patents is U.S. Patent No. 8,935,580 (“’580 patent”) which issued January 13, 2015 and is currently assigned to Apple Inc. The application for the ’580 patent was filed with the USPTO on June 18, 2013, with priority to an earlier application filing date of July 2, 2008, which became U.S. Patent 8,468,426. The ’580 patent, titled “Multimedia-Aware Quality-of-Service and Error Correction Provisioning,” discloses a method of

encoding a stream of data packets in which the packets are routed to separate “QoS processing modules,” each possessing their own QoS profile, where the packets are processed according to multiple factors, such as priority, traffic rate, latency, tolerated jitter, transmitter scheduling, media type, etc. Subsequent to this processing, the packets are combined and transmitted.

16. The services I provide include consulting in telecommunication standards setting, protocol simulation and implementation, technology assessments, engineering lab testing, and product analysis. Through Bims Laboratories, I have participated in technical discussions for several years in the IEEE 802 Committee to develop new technology that allows LTE and 5G networks to coordinate their network activities with Wi-Fi networks.

17. I have also served in several leadership capacities with the IEEE 802.16 Working Group (2009-present) and the IEEE 802.15 Working Group where I currently serve as technical editor for a wide-area cellular standard that is a competitor to LTE and 5G, and I am currently a voting member in the IEEE 802.11 Working Group. Within the IEEE 802.15 Working Group, I am the technical editor for IEEE 802.16t, an amendment to the WiMAX standard that supports long-range wireless networks for the railroad industry. Furthermore, since 2003, I have served as a wireless technology expert in patent infringement cases. I am actively involved in liaison conversations regarding the interoperation of IEEE 802.11-

based wireless networks and 3GPP networks, including 3G, LTE, and 5G networks.

18. I am an author of 9 technical publications and a named inventor of 23 telecommunication-related patents, including U.S. Patent No. 8,995,996, which describes how wireless devices that interfere with each other can coordinate their communications in an optimal manner. The entire list of those patents and publications can be found in my full CV.

19. I have been active in professional engineering associations and research bodies. From 1985 to 1991, I was the AT&T Bell Labs Cooperative Research Fellow. In 2000, I served as a Chair of IEEE Engineering Management Society, Silicon Valley Chapter. From 2007 to 2010, I served as a vice chair and secretary of IEEE 802.16h License-Exempt Task Group and have been a senior member of IEEE since 2012.

B. Compensation

20. I am being compensated for services provided in this matter at \$850 per hour. My compensation is not conditioned on the conclusions I reach as a result of my analysis or on the outcome of this matter, and in no way affects the substance of my statements in this declaration.

21. I do not have any financial interest in any Petitioner, or any of its subsidiaries. I also do not have any financial interest in Patent Owner, Headwater

Research LLC. Further, I do not have any financial interest in the '700 patent, and I have not had any contact with the named inventors of the '700 patent: Gregory G. Raleigh, James Lavine, and Alireza Raissinia.

IV. MATERIALS CONSIDERED

22. In forming my opinions, I read and considered the '700 patent and its prosecution history, the exhibits listed in the Exhibit List filed with the petition for *inter partes* review of the '700 patent, as well as any other materials referenced herein.

V. LEGAL STANDARDS

23. In forming my opinions and considering the subject matter of the '700 patent and its claims in light of the prior art, I am relying on certain legal principles that counsel in this case has explained to me. I understand that the claims define the invention. I also understand that an unpatentability analysis is a two-step process. First, the claims of the patent are construed to determine their meaning and scope. Second, once the claims have been construed, the content of the prior art is compared to the construed claims. I have been told the following legal principles apply to analysis of patentability based on 35 U.S.C. §§ 102 and 103. I also have been told that, in an *inter partes* review proceeding, a patent claim may be deemed unpatentable if it is shown by a preponderance of the evidence that the claim was anticipated by a prior art patent or publication under

§ 102 and/or rendered obvious by one or more prior art patents or publications under § 103.

A. Level of Ordinary Skill

24. I understand a person of ordinary skill in the art is determined by looking at (1) type of problems encountered in the art; (2) prior art solutions to those problems; (3) rapidity with which innovations are made; (4) sophistication of the technology; and (5) educational level of active workers in the field.

25. Therefore, based on my experience and accounting for these factors, I believe a person of ordinary skill in the art would have had, as of January 28, 2009 (the date I have been told to assume is the effective filing date of the '700 patent), (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. More education, e.g., a doctorate in an academic area emphasizing electrical engineering, computer engineering, computer science, or a related technical field, would compensate for fewer years of work experience. I meet this definition of one of ordinary skill in the art now and also at the earliest possible priority date of the '700 patent.

26. I am also familiar with the knowledge of the person of ordinary skill in the art as of the priority date of the '700 patent. I am able to opine on how the

person of ordinary skill in the art would have understood the disclosure and claims of the '700 patent, the disclosures of the prior art, the motivation to combine the prior art, and what combinations would have been obvious to one of ordinary skill in the art. In the following, I also refer to a person of ordinary skill in the art using the short-form acronym "POSITA."

B. Claim Construction

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

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

29. I also understand that in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have

been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

C. Anticipation under 35 U.S.C. § 102

30. I understand that a patent claim is invalid as anticipated if a single piece of prior art teaches every element of the claims, viewed from the perspective of a person of ordinary skill in the art.

31. I have also been told that an anticipatory reference does not have to recite word for word what is in the anticipated claim. Rather, anticipation can occur when a claimed limitation is “inherent” in the relevant reference. I have been told that if the prior art necessarily functions in accordance with, or includes, the claim limitations, it can anticipate even though the reference does not recite a limitation word for word.

D. Obviousness under 35 U.S.C. § 103

32. I understand that a patent claim is invalid as obvious if the claimed invention would have been obvious to a person of ordinary skill in the art at the time the claimed invention was made. This means that even if all of the elements of the claim cannot be found in a single prior art reference that would anticipate the claim, a person of ordinary skill in the field who knew about all the prior art would have come up with the claimed invention. I understand that in an obviousness determination, the person of ordinary skill in the art is presumed to have

knowledge of all material prior art. I understand that whether a claim is obvious is based upon the determination of several factual issues.

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

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

35. I understand that a patent claim composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. In evaluating whether such a claim would have been obvious, one may consider whether a reason has been identified that would have prompted a person of ordinary skill in the art to combine the elements or concepts from the prior art in the same way as in the claimed invention. There is no single way to define the line between true inventiveness on the one hand (which is patentable) and the application of common sense and ordinary skill to solve a problem on the other hand (which is not patentable). For example, market forces or

other design incentives may be what produced a change, rather than true inventiveness.

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

37. I understand that one must be careful not to determine obviousness using the benefit of hindsight; many true inventions might seem obvious after the fact. I understand that the determination should be based on the position of a

person of ordinary skill in the field at the time the claimed invention was made and should not consider what is known today or what is learned from the teaching of the patent.

E. Secondary Considerations of Non-Obviousness

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

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

40. I have also been informed that the patentee must establish a nexus between any secondary consideration factors and the claimed invention. Where a secondary consideration factor results from something other than what is both

claimed and novel in the patent, there is no nexus to the merits of the claimed invention.

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

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

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

VI. BACKGROUND AND TECHNOLOGY OF THE '700 PATENT

A. Overview of the '700 Patent

44. According to the '700 patent, wireless network resources of many wireless networks are stretched thin. “The increasing popularity of various smart phone devices, net book devices, [and] tablet computing devices,” is leading to a “network capacity crunch.” '700 patent, 3:33-41. Network resources, e.g., base stations, are inundated with requests for network access, leading to “flooding” or “overloading” of base stations. '700 patent, 6:23-7:5. When this happens, base stations become “overwhelmed” by the requests and may begin “fall[ing] behind” in servicing the requests. '700 patent, 6:61-7:5.

45. According to the '700 patent, a device's network access requests may vary in several ways. First, they may vary in their priority. Typically, each request is associated with a particular application. '700 patent, 4:8-24. Because the services provided by these applications vary, the priority of requests may vary.

46. For example, a service that is “interactive,” such as an application that is interacting with a user, may be higher priority because it requires “reasonably low response time” for a request, while a service occurring in the background may be “lowest priority.” '700 patent, 16:50-60. For other services, it may make sense to hold off sending requests until a “scheduled transmission time.” '700 patent, 88:36-44.

47. To avoid “overloading” the base stations, the patent explains, the device intercepts requests by applications to launch or access the network. ’700 patent, 6:23-7:5, 11:24-45, 66:38-54. The device then handles requests differently based on priority, likelihood to lead to repeated requests, or other classifications, as set forth in a “control policy.” ’700 patent, 11:55-58, 12:48-57. For example, the device associates policy controls with particular applications. ’700 patent, claim 1. Then, when an application issues a request for a time to execute an activity, the device intercepts the request and schedules the time for execution of that activity, according to the policy controls. ’700 patent, claim 1.

B. Claims of the ’700 patent

48. I understand that the IPR Petition challenges claims 1-20 of the ’700 patent. I further understand that among those claims, claim 1 is an independent claim.

VII. ANALYSIS

49. Based on my personal experience at the relevant time and my review of the prior art, it is my opinion that: (1) claims 1, 2, 4, 5, 7-11, 13, and 15-17 would have been obvious to a POSITA as of the effective filing date of the ’700 patent based on a combination of *Rao* and *Scahill*; (2) claim 3 would have been obvious to a POSITA as of the effective filing date of the ’700 patent based on a combination of *Rao* in view of *Scahill* and *Oestvall*; (3) claims 6, 14, 18, and 19

would have been obvious to a POSITA as of the effective filing date of the '700 patent based on a combination of *Rao* in view of *Scahill* and *Montemurro*; and (4) claims 12 and 20 would have been obvious to a POSITA as of the effective filing date of the '700 patent based on a combination of *Rao* in view of *Scahill* and *Araujo*. In my opinion, the Challenged Claims recite combinations of well-known prior art elements that perform their known functions to produce predictable results.

A. Ground 1: The Combination of *Rao* in View of *Scahill* Renders Claims 1, 2, 4, 5, 7-11, 13, and 15-17 Obvious

1. Overview of the *Rao* Reference

50. Like the '700 patent, *Rao* “generally relates to optimizing peer-to-peer network communications, and in particular, to techniques for application-aware prioritizations of network communications on a client.” *Rao*, [0002]. In my opinion, because *Rao* is directed to the same field of endeavor as the '700 patent (application management in devices that can run multiple applications) and same problem as the '700 patent (optimizing use of network communications on an application-by-application basis), *Rao* is analogous art to the '700 patent.

51. As *Rao* explains, in prior art computing devices (also referred to as “client[s]”), “[c]ommunications over a network . . . are typically processed in the order generated by the activity of a user and applications of the client.” *Rao*,

[0003]. According to *Rao*, though, this is not “always desirable.” As one example, “a network packet generated or received for an application running in the background may be processed ahead of a network packet generated or received for the application running in the foreground.” *Rao*, [0003]. In another example, a voice-over-IP (VoIP) telephone call may experience increased latency and reduced quality because the packets of another application were processed ahead of the packets of the VoIP telephone call. *Rao*, [0003].

52. Accordingly, *Rao* proposes a prioritization scheme that “provide[s] application-aware, client-specific prioritization of packet traffic.” *Rao*, [0003]. To facilitate the prioritization scheme, *Rao*’s computing device includes a remote access client 120 that interacts with applications 338 on the computing device and the computing device’s network stack 310, as illustrated in Figure 1B and Figure 5A below. *Rao*, [0098], [0101], [0180], [0184]. As *Rao* explains, the remote access client 120 “manages and controls network communication prioritizations on the client.” *Rao*, [0080].

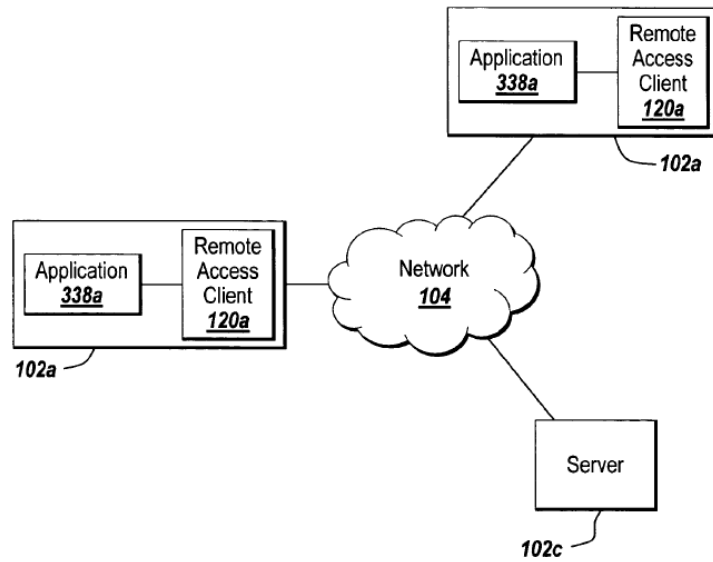


Fig. 1B

Rao, Fig. 1B.

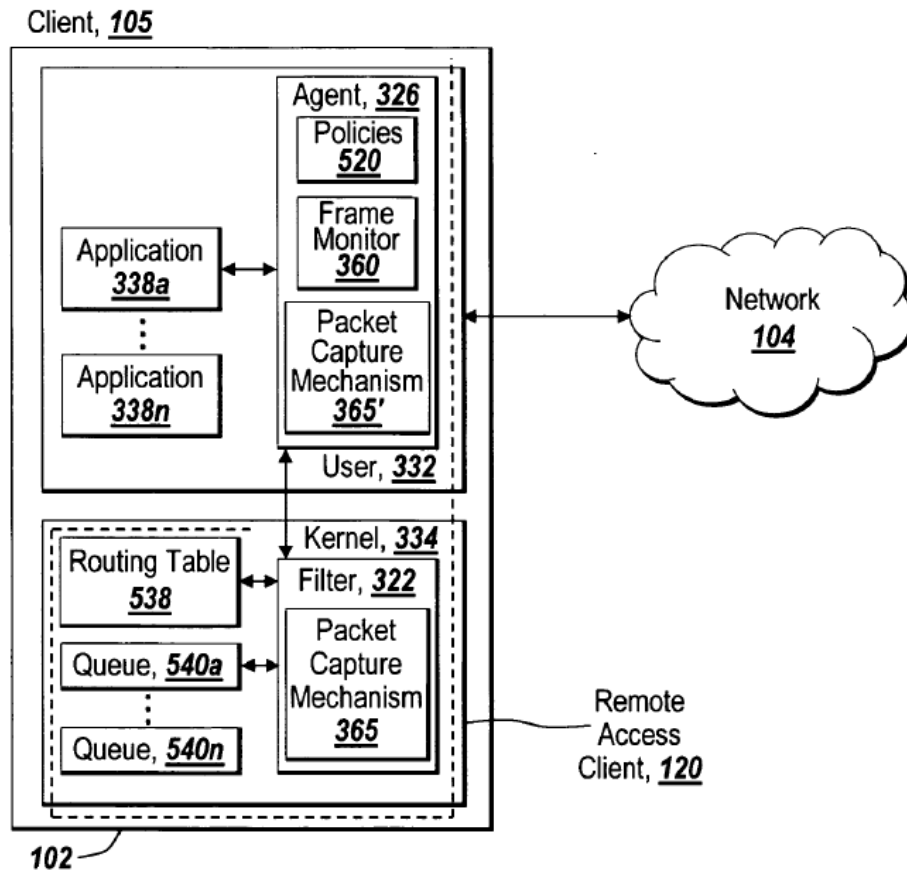


Fig. 5A

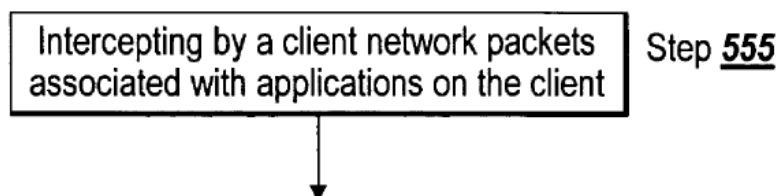
Rao, Fig. 5A.

53. As shown, Rao's remote access client 120 includes an agent 326 that stores policies 520. These policies, Rao explains, "specify[] client-side prioritization of network communications related to applications" at the computing device. Rao, [0182]. As one example, the policies may "define prioritization based on whether an application is running in the foreground or the background of the

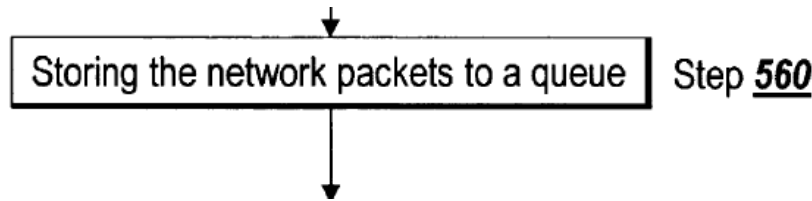
client 105.” *Rao*, [0182]. *Rao* envisions a “multitude of ways to define client-side application priorities,” including conditionally and hierarchically. *Rao*, [0182].

54. Further, as shown, *Rao*’s remote access client 120 includes a filter 322 that employs a packet capture mechanism 365 and communicates with queues 540a-540n. *Rao*, [0099], [0101], [0180]. As *Rao* explains, the packet capture mechanism 365 “may use any hooking application programming interface (API) to intercept, hook, or otherwise obtain . . . network traffic associated with [an] application.” *Rao*, [0110]. The queues, *Rao* discloses, are for “queuing and prioritizing [the] network communications” obtained by the packet capture mechanism 365. *Rao*, [0180].

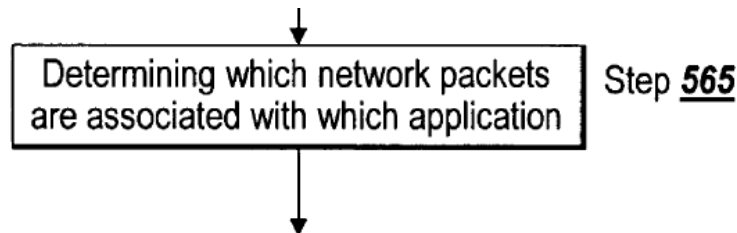
55. *Rao*’s Figure 5B illustrates one example of its prioritization technique. *Rao*, [0184]. The process includes five steps: steps 555, 560, 565, 570, and 575. *Rao*, Fig. 5B. Beginning with step 555, “client 105 intercepts,” *e.g.*, using the packet capture mechanism, “one or more network[] packets associated with an application 338a-338n on the client 105.”



Rao, [0184], [0186], Fig. 5B. Then, “[a]t step 560, the network packets intercepted at step 555 may be stored in a queue 540a-540n,” such as a “temporary queue.”

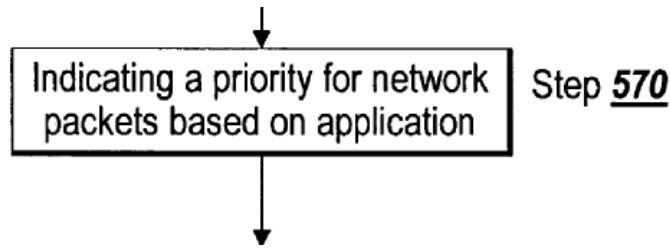


Rao, [0186], Fig. 5B.¹ “At step 565, the remote access client 120 . . . determines the association of network packets with applications 338a-338n in order to determine priorities and apply any priority based policies.”

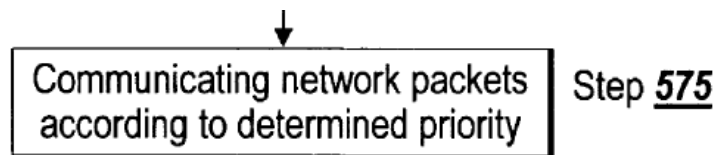


Rao, [0186], [0187], Fig. 5B. And, “[a]t step 570, the remote access client 120 . . . indicates a priority for intercepted . . . network packets based on the application 338a-338n associated with the packet at step 565.” *Rao*, [0189]. For example, the remote access client may “use[] the policies 520 to apply a priority to network packets of applications 338a-338n in accordance with the prioritization rules specified or indicated by the policies.”

¹ According to *Rao*, this step is optional. *Rao* discloses that “[i]n other embodiments, the network packet is stored to a determined priority queue 540a-540n or a queue 540a-540n associated with an application 338a-338n after the network packet is prioritized at steps 565 and/or step 570.” *Rao*, [0186].



Rao, [0189], Fig. 5B. Finally, at step 575, “the network packets are communicated from the queues 540a-540n according to the determined priorities . . . which, in turn, may be based or derived from the policies 520 of the client.” *Rao*, [0194]. To this end, *Rao* explains, the prioritized packets are “provide[d] . . . for communications via a network stack of the client.” *Rao*, [0038]. The network stack, *Rao* explains, “may comprise one or more network layers[,] . . . one or more protocols,” and/or “one or more network drivers . . . included as part of the operating system . . . or as part of any network interface cards or other network access components of the computing device.” *Rao*, [0100].



Rao, [0184], Fig. 5B. Thus, in *Rao*, “outbound network packets generated by an application 338a-338n on the client 105 are prioritized by the client 105 prior to transmission based on the type and/or priority of the application 338a-338n.” *Rao*, [0184].

2. Overview of the *Scahill* Reference

56. *Scahill* recognizes that “users are typically unaware of which network connections are available to them at any particular time or of the characteristics of an available network connection such as whether a network is congested or not.” *Scahill*, 3:23-25. *Scahill* further recognizes that “a user may not be aware of the optimum network connection characteristics a particular application hosted on a wireless communications device associated with a user might require to perform a particular task.” *Scahill*, 3:25-27. To address these problems, *Scahill* teaches that it would be useful “if the execution of the plurality of applications is somehow coordinated.” *Scahill*. 4:9-11. *Scahill* discloses one such solution as a method that “seeks to mitigate and/or obviate the limitations known in the art with the timing of data transmissions from mobile communications devices.” *Scahill*, 5:8-9. In particular, like the ’700 patent, *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. *Scahill*, Abstract. In my opinion, because *Scahill* is directed to the same field of endeavor as the ’700 patent (application management in devices that can run multiple applications) and same problem as the ’700 patent (optimizing use of network resources on an application-by-application basis), *Scahill* is analogous art to the ’700 patent.

57. According to *Scahill*, “[t]he connection requests generated by . . . [a] client application 12a may require network connections which vary in time and/or which vary depending on the amount of data and/or type of task to be performed.” *Scahill*, 19:9-11. Thus, when an application requests execution of a task, *Scahill*’s method entails (1) determining task completion conditions, including whether any network or device conditions are required; (2) retrieving stored data indicating network characteristics for an available network connection during a predetermined time period; (3) processing the task completion conditions to determine if the network characteristics retrieved for the predetermined period of time are sufficient to complete the task; and (4) in the event the network characteristics are sufficient, scheduling the task for execution. *Scahill*, Abstract.

58. *Scahill*’s Figure 1B illustrates a client application 12a for carrying out *Scahill*’s method. Client application 12a includes “a network aware scheduling software client library 14a.” *Scahill*, 18:15-16. This “client library 14a is arranged to allow a connection requesting component 11 of the application 12a to interface with an application scheduler 24 library component of the client library 14a.” *Scahill*, 18:16-18.

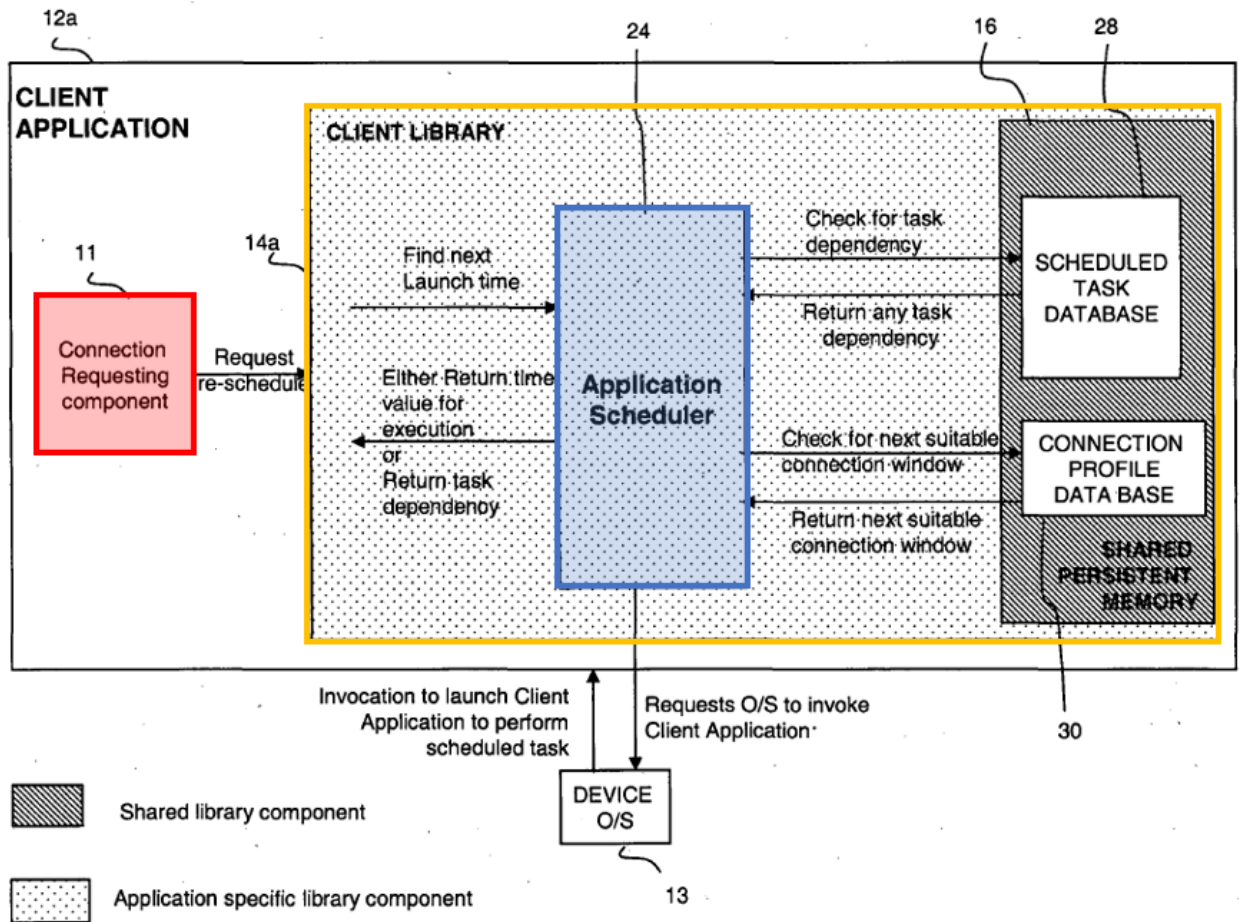


FIGURE 1B

Scahill, Fig. 1B (annotated).

59. The application's connection requesting component 11 may request a network connection so as to execute an application task. *E.g.*, *Scahill*, 19:9-11. The application scheduler schedules the task by assigning the task an "execution time" and storing it in the scheduled task database 28. *Scahill*, 19:18-35, 36:15-17. When the time arrives for a data transmission to occur, the client library 14a generates instructions for the device's operating system to invoke the client application 12a

to perform the relevant task. *Scahill*, 20:5-8. The operating system launches the client application 12a when all necessary conditions are satisfied, such as at a specific time requested by the application. *Scahill*, 20:9-10. *Scahill* discloses that the request may specify “task completion conditions” for the task, including a “start delay” (also called a “timing interval Δt ”) and a “duration . . . of the required connection.” *Scahill*, 24:19-23, 26:17-20.

3. Introduction to Rationale to Modify *Rao* in view of *Scahill*

60. In my opinion, it would have been obvious to a POSITA to improve *Rao*’s remote access architecture to include *Scahill*’s method of coordinating network communications.

61. *Rao* discloses a method of “application-aware prioritization of client-side network communications.” *Rao*, Abstract. Under *Rao*’s disclosure, as explained above, the remote access client uses policies to prioritize network communications from certain applications over others, such as packets from foreground applications prioritized ahead of packets from background applications that are relatively and consequentially de-prioritized. In this manner, *Rao*’s remote access client ensures network communications from certain applications are preferentially transmitted relative to network communications from other applications.

62. *Rao* also discloses that the remote access client “considers other network factors in determining what network packets to communicate from what queue.” *Rao*, [0195]. As an example, *Rao* discloses that “remote access client 120 may receive an indication of network congestion,” and as a result, “may throttle back or not communicate [certain] network packets.” *Rao*, [0195]. Thus, according to *Rao*, “client 105 controls and manages the prioritization of network communications of the client 105 on an application 338a-338n basis and in accordance with any policies 520 for the client 105 and in further view of any network statistics and other factors occurring on the network.” *Rao*, [0195].

63. In my opinion, a POSITA would have understood that in order to manage the prioritization of network communications on an application-by-application basis, in accordance with client policies, the client would have to dynamically coordinate the network communications of its applications. However, *Rao* provides limited detail on how to dynamically coordinate communications in view of network congestion or to resolve conflicts between applications that may arise even after *Rao*’s prioritization method is applied, such as starvation in an application that has a lower priority than another for an extended period of time (e.g., network packets from a background application may never transmit if the foreground application’s needs are always prioritized).

64. As a result, a POSITA implementing *Rao*'s teachings would have looked to similar references, including *Scahill* since it provides this detail when it teaches such a coordination method that is complementary to *Rao*'s prioritization method. *Scahill*, 4:15-18.

65. For example, *Scahill* "seeks to mitigate and/or obviate the limitations known in the art with the timing of data transmissions from mobile communications devices." *Scahill*, 5:8-9. *Scahill* recognized that, in prior art architectures where a user manually schedules when applications transmit their packets to perform synchronization, the scheduling of network communications "may not coincide with appropriate network conditions to support the type of data transmission required to complete the scheduled task." *Scahill*, 2:14-17. The result is that "[e]ach attempt at transmission [with prior art systems] needlessly wastes the resources of the device and if a data transmission is aborted as the network conditions alter in the course of the transmission, network resources have also been wasted." *Scahill*, 2:17-20.

66. Thus, according to *Scahill*, "it is useful if the task execution can be timed to coincide with appropriate network conditions to prevent having any necessity to perform such a delay operation." *Scahill*, 2:24-25. *Scahill* thus "seek[s] to preserve device resources so that they are used only when there is a likelihood of a connection having characteristics which will allow an application to

successfully complete one or more scheduled tasks in a particular connection window.” *Scahill*, 13:31-34. To this end, *Scahill*’s application scheduler dynamically schedules application tasks at times when appropriate network conditions (or other device conditions) are likely to be present. *Scahill*, 19:20-25.

67. *Scahill* further discloses that its method is beneficial for other reasons. First, *Scahill* explains that “[b]y coordinating a plurality of applications in their execution each task can be performed by one of the plurality of applications under suitable network conditions in a way which reduces or removes the likelihood of any conflict with other applications trying to also establish a connection over the same wireless communications link.” *Scahill*, 4:11-15. According to *Scahill*, “[e]ven when a wireless communications device can support more than one network connection it is useful if conflicts can be resolved to ensure applications . . . are coordinated as otherwise conflicts could still occur between applications.” *Scahill*, 4:15-18.

68. Second, *Scahill*’s method reduces the financial costs associated with certain data transmissions. For example, “under certain [network] conditions, e.g., if located in a roaming data network, the user may be on a tariff which would mean that each aborted data transmission still incurred a cost for the user.” *Scahill*, 2:20-22. *Scahill*’s method reduces the occurrences of these roaming costs by allowing

each application to specify a particular network for execution and scheduling the application's task at an execution time when the specified network is available.

69. Third, *Scahill*'s method optimizes a mobile device's power use. According to *Scahill*, prior art transmission methods unnecessarily drained battery resources, including where (1) a device must "continually scan for available networks and trigger applications when a particular network is available"; (2) "there may be long periods of time (many hours or even days) between the user being in range of a suitable network"; and (3) a user is only temporarily in range of a suitable network such that "[i]f the user passes through too quickly and the opportunity for use is limited, then an application has used up device resources such as electrical battery power attempting to use a network connection to complete a task which must then be aborted if the connection prematurely fails." *Scahill*, 4:20-30. *Scahill*'s method optimizes power consumption in these scenarios by scheduling tasks in a particular connection window. *Scahill*, 4:20-30. In view of these benefits, a POSITA would have been motivated to modify *Rao*'s architecture to further include *Scahill*'s scheduling method to extend the battery life of the device. In addition to improving *Rao*'s architecture, *Scahill*'s method aligns with *Rao*'s intent of optimizing the data transmission process.

70. In my opinion, a POSITA would have understood that *Scahill*'s application scheduling method is complementary to *Rao*'s application

prioritization method. For example, *Rao* discloses that its remote access client may use characteristics of an application, such as whether that application is running in the foreground or background, to determine the priority of a given network packet. *Rao*, [0189]. However, a POSITA would have understood that *Rao*'s disclosure does not prevent conflicts between applications, such as when a lower priority application is unable to transmit for an extended period of time, because the scheduling of tasks in prioritized applications is not coordinated to prevent conflicts when transmitting packets. Indeed, *Rao* itself envisions that its remote access client may use characteristics of the network, such as congestion, to dynamically adjust which network packets communicate from which priority queue. *Rao*, [0195]. A POSITA would have understood that *Scahill* further builds on *Rao*'s disclosure, by disclosing how to schedule the transmission of network communications so as to reduce conflicts (i.e., improve coordination among applications with potentially different priorities) in view of network conditions. Thus, rather than just prioritizing certain applications over others in queues, *Rao/Scahill*'s architecture would also schedule transmissions of prioritized applications' network communications to reduce conflicts and optimize network conditions. As *Scahill* recognized, such scheduling can also provide the further benefits of minimizing transmission conflicts, reducing financial costs, and optimizing a device's power consumption.

71. In my opinion, a POSITA would have had a reasonable expectation of success in modifying *Rao*'s architecture in view of *Scahill*'s method. *Rao* allows a great deal of flexibility with respect to how network communications from applications would be queued and communicated. Regarding queueing, for instance, *Rao* allows that "[t]he queues 540a-540n may comprise any type and/or form of suitable means and/or mechanisms for storing and/or arranging network packets," and discloses that an ordinary artisan "will recognize and appreciate that the network packets may be arranged in various priority queues 540a-540n, such as high, medium, low, or by any other granularity, and be placed or arranged in the queue in any suitable manner." *Rao*, [0180], [0191]. Regarding communicating, *Rao* discloses the prioritized packets are "provide[d] . . . for communications via a network stack of the client," which "may comprise one or more network layers . . . one or more protocols," and/or "one or more network drivers . . . included as part of the operating system . . . or as part of any network interface cards or other network access components of the computing device." *Rao*, [0038], [0100]. A POSITA would have appreciated that this flexible disclosure of queueing and communicating in *Rao* would accommodate queueing and communicating based on *Scahill*'s scheduling method (e.g., assigning an execution time to an application's task).

72. In my opinion, a POSITA would have understood how to implement *Scahill*'s scheduling method in *Rao*'s architecture. Under *Rao*'s architecture, the remote access client prioritizes network communications of the client 105 on an application 338a-338n basis, in accordance with the policies 520, and communicates the network communications in view of any network statistics or other network factors. *Rao*, [0195]. *Scahill* discloses that its scheduling method is implemented through a scheduling software client library (and related components). *Scahill*, 18:15-22. In *Rao*'s modified architecture, device applications would likewise include or otherwise interface with this scheduling software client library (and related components) to schedule transmissions. The scheduling library (also referred to as an application scheduler) would assign execution times for transmitting the applications' network communications in accordance with the policies 520, taking into consideration likely network conditions as *Scahill* describes. The result is that *Scahill*'s application scheduler and related components are integrated into *Rao*'s architecture and work together to prioritize network communications into queues and assign execution times using *Rao*'s policies, taking into consideration *Scahill*'s scheduling requests and other network factors. In my opinion, a POSITA would have understood how to modify *Rao*'s architecture to accommodate *Scahill*'s scheduling functionality.

4. Independent Claim 1

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

73. In my opinion, the combination of *Rao* and *Scahill* renders obvious the preamble. *Rao* discloses “multiple computing devices 102a-102c” (also referred to as “clients 105a-105c”) that connect to a network using one or more network connections. *Rao*, [0086]. “[C]lient 105 can be any type and/or form of computing device 102 that can run one or more applications 338,” such as a “web browser, web-based client, client-server application, [or] a thin-client computing client.” *Rao*, [0088]. For example, the device may also be a “mobile client,” such as a “notebook, personal digital assistant (PDA), a smart phone,” or “telecommunication device.” *Rao*, [0198]; *see also Rao*, [0130]. The device “can connect to one or more networks . . . through a variety of connections including . . . wireless connections.” *Rao*, [0095], [0198]. *Rao*’s device is an “end-user device” because *Rao* discloses a “user” of the device. *Rao*, [0183], [0199].

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

74. In my opinion, the combination of *Rao* and *Scahill* renders obvious 1[a]. Device 102 includes network interface 118, which enables connectivity to a “Local Area Network (LAN), Wide Area Network (WAN) or the Internet through a variety of connections,” such as “standard telephone lines, LAN or WAN links,” “broadband connections,” “wireless connections,” or “some combination of any or

all of the above.” *Rao*, [0095], [0125]. Network interface 118 includes “a built-in network adapter, network interface card, PCMCIA network card, card bus network adapter, USB network adapter, modem, or any other device suitable for interfacing the computing device 102 to any type of network capable of communication.” *Rao*, [0125].

- c. **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;”**

75. In my opinion, the combination of *Rao* and *Scahill* renders obvious 1[b]. *Rao* discloses the claimed “plurality of executable applications stored on the wireless end-user device” because each client 105 includes “one or more applications 338a-338n.” *Rao*, [0087]. *Rao* explains that “applications 338a-338n provide one or more real-time data communications, such as VoIP,” or “may provide email, collaboration, online meeting, and/or desktop sharing related services or functionality.” *Rao*, [0179]; *see also Rao*, [0088] (describing breadth of applications).

76. *Rao* also discloses that “[e]ach client 105a-105n includes a remote access client 120a-120c.” *Rao*, [0087]. Figure 5A (below) shows a system having a remote access client for routing network packets from a client 105 to a network 104 (via network interface 118):

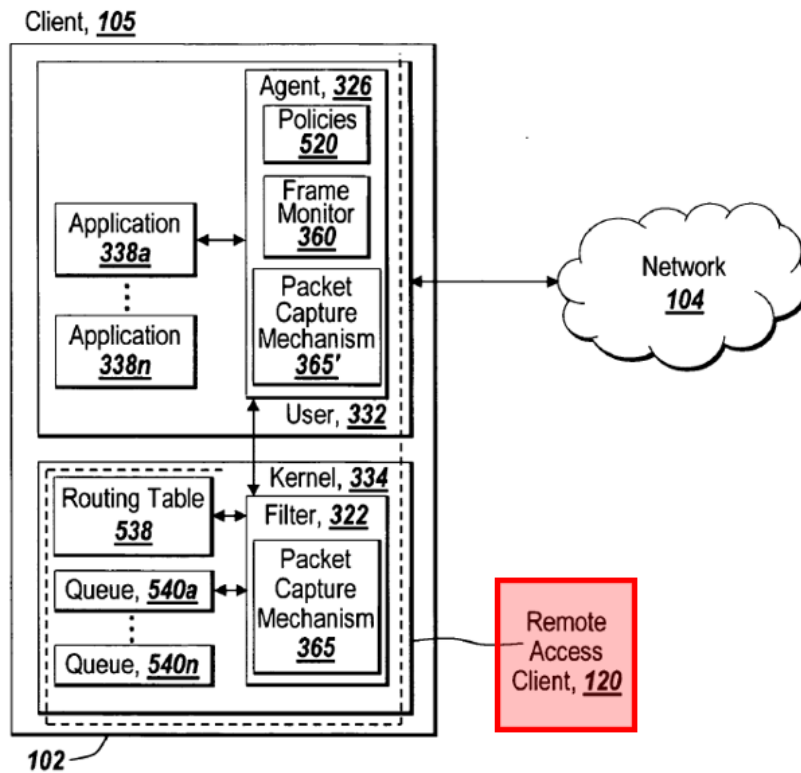


Fig. 5A

(*Rao*, Fig. 5A (annotated).) “[T]he remote access client 120 may have one or more policies 520 for specifying client-side prioritization of network communications related to applications 338a-338n.” *Rao*, [0182]. These policies 520 are the claimed “network access policy controls.” According to *Rao*, “[t]hese policies 520 may be specified by any suitable means and/or mechanisms,” including “by the name of the application 338a-338n and/or the type of application 338a-338n,” “in accordance to the type of one or more protocols used by the applications 338a-338n,” “the size of the payload of the network packet,” “based on whether an application is running in

the foreground or the background,” “destination network address,” “and/or destination port number.” *Rao*, [0182]. *Rao* explains that an ordinary artisan “will recognize and appreciate the multitude of ways to define client-side application priorities.” *Rao*, [0182].

77. *Rao*’s remote access client 120, via “agent 326,” operates as the claimed “activity classifier” to “dynamically associate one or more corresponding network access policy controls with each of the plurality of executable applications.” Under step 565 of *Rao*’s process (depicted in Figure 5B), remote access client 120 “determines the association of network packets with applications 338a-338n in order to determine priorities and apply any priority based policies 520.” *Rao*, [0187]. *Rao* further explains that “client 105, such as via agent 326, can associate network traffic with applications 338a-338n by any suitable means and/or mechanisms.” *Rao*, [0187]; *see also Rao*, [0187] (listing the multiple ways agent 326 can identify a network packet as generated from an application 338a-338n).

78. In my opinion, a POSITA would have understood *Rao*’s generation or receiving of network packets corresponds to “activit[ies]” executed by the applications. For instance, *Rao* describes the “a network packet generated or received for an application running in the background may be processed ahead of a network packet generated or received for the application running in the

foreground.” *Rao*, [0003]. The VoIP application’s network packets thus correspond to an activity (i.e., a VoIP telephone call) of the VoIP application. This is consistent with other disclosures in the prior art. For example, in describing its invention, *Scahill* discloses that it “assumes that there is a one-to-one mapping between a task and a client application.” *Scahill*, 17:6-10.

79. Remote access client 120 associates policies 520 “dynamically” because, for example, the policy currently associated with the application may depend on present characteristics of an application, including whether the application “is running in the foreground” (and in active use by the user) or “is running in . . . the background” (and not in active use by the user). *Rao*, [0188]. Similarly, “remote access client 120 may determine any other characteristics or statistics of the application 338a-338n such as size, memory usage, total execution time, and/or frequency of use.” *Rao*, [0188]. These characteristics reflect an application’s current state and may change over time, requiring dynamic association by remote access client 120.

80. In addition, *Scahill*’s application scheduler also operates to “dynamically associate one or more corresponding network access policy controls with each of the plurality of executable applications” by coordinating the scheduling of tasks (where each schedule is the result of a network access policy

control) for each of the plurality of executable applications to avoid conflicts between applications. *Scahill*, Abstract, 19:18-35, Fig. 1B.

- d. 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”**

81. In my opinion, the combination of *Rao* and *Scahill* renders obvious 1[c]. In *Rao/Scahill*’s architecture, “client library 14a is arranged to allow a connection requesting component 11 of the application 12a to interface with an application scheduler 24 library component of the client library 14a.” *Scahill*, 18:16-18. *Scahill* discloses that “[t]he connection requesting component 11 of the client application 12a is effectively any part of the client application 12a which generates a connection request to perform a task.” *Scahill*, 18:30-31; *see also* *Scahill*, 19:18-20 (discussing “a re-schedule request”). This connection requesting component of the client application thus requests an execution time for the application.

82. According to *Scahill*, “[w]hen a client application 12a wants to know its next launch time, the client library 14a of the client application 12a calls up, via an application programming interface [(“API”)], the client library application scheduling component 24.” *Scahill*, 43:3-5.

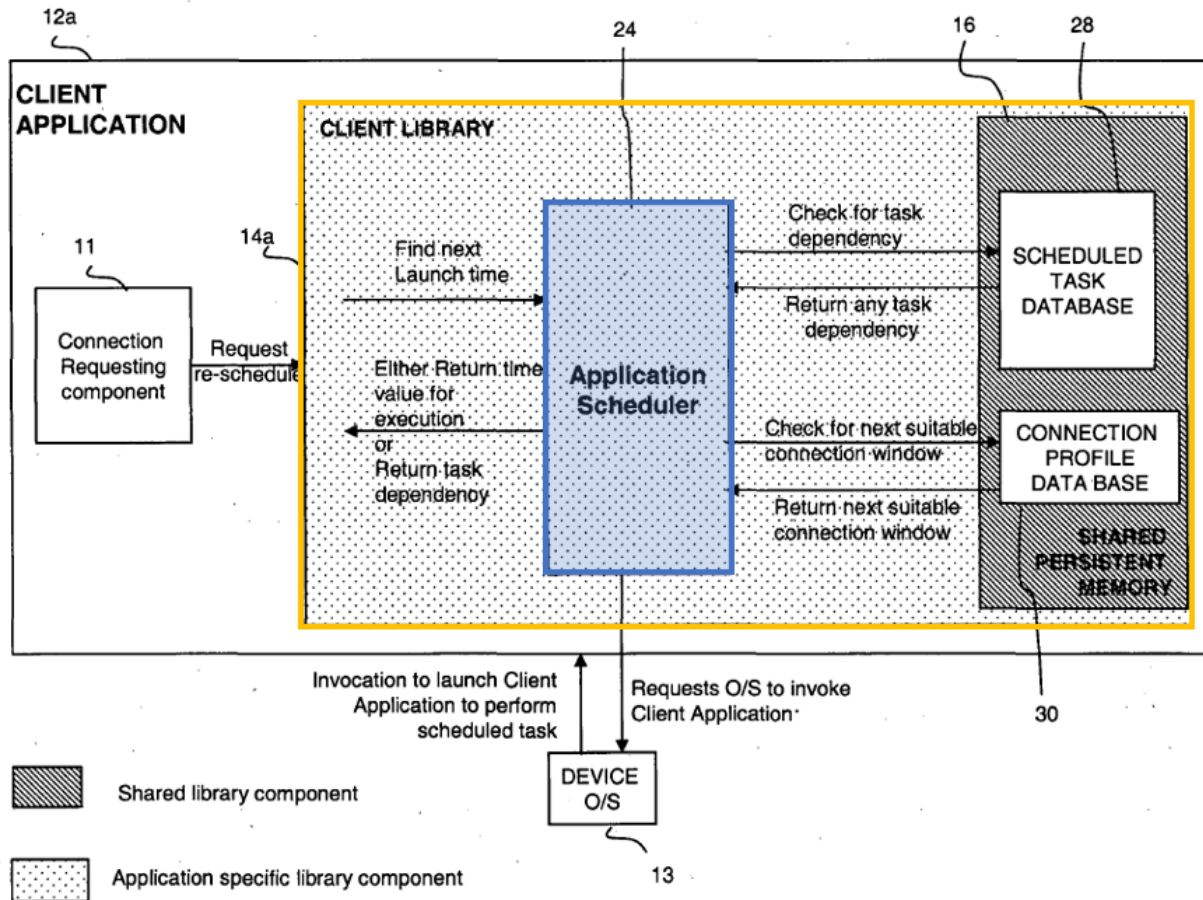


FIGURE 1B

Scahill, Fig. 1B (annotated). A POSITA would have understood that this API is the “Application Programming Interface (API) accessible to each of the plurality of executable applications,” as claimed. This API is accessed by each application to request an execution time because the API provides an interface to the client library application scheduling component 24 (application scheduler). *Scahill* teaches that client library 14a is “accessible by all client applications,” and the scheduling component API (within client library 14a) is thus callable by each of the executable applications to request an execution time. *Scahill*, 16:29-32. And according to

Scahill, “[t]his process (determining the next launch time) can be implemented using any conventional programming technique apparent to those of ordinary skill in the art as appropriate for the process.” *Scahill*, 43:5-7; *see also* 43:9-44:11.

83. An API is an Application Programming Interface and it is well understood by a POSITA. It is a specification that details the collection of functions, including their input parameters and output results, that allows an application program to execute a common set of software functions. For example, an operating system implements a common set of software functions for controlling various functions, including printing a file, storing a photo on a hard drive, turning on a microphone or speaker. Software functions such as these require input parameters and output results that have a particular format that is part of an API specification. Any application running on a computer can access its operating system functions through such an API.

84. In my opinion, a POSITA would have understood when *Scahill*’s client application issues a “request to perform a task,” it is requesting an execution time, because it entails a request for a scheduled time slot. *Scahill*, 18:30-31. In *Scahill*, the application’s request is made to a “scheduler” that schedules a “connection window” for the task, where “connection window ... refers to a period of time of predetermined duration” that can be scheduled at “an absolute time value.” *Scahill*, 19:25-26, 26:17-23. Moreover, in *Scahill*, “[t]here are a number of

application specific requirement(s),” also called “connection conditions,” “which the application scheduler library component 14a,b of each client application 12a,b ... must take into account when seeking to schedule performance of a task.”

Scahill, 19:3-7, 24:14-17. This can include a “timing interval Δt requested by the client application since the task was last performed.” *Scahill*, 24:19-23. For example, an application’s request could specify a “delay of 172800 secs and duration of 62 secs.” *Scahill*, 42:25-31. In each case, the API receives a request for a scheduled time slot, the “connection window,” as part of a request for an execution time from an application.

85. As I discussed in Section VII.A.3 (Motivation), under *Rao/Scahill*’s architecture, remote access client 120 manages the prioritization of network communications in accordance with at least policies 520 and *Scahill*’s scheduling requests. *Rao*’s architecture, as modified in view of *Scahill*, thus includes an API that is both accessible to each of the plurality of executable applications and callable by each of the executable applications to request an execution time according to the corresponding network access policy controls for that application.

- e. **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.”**

86. In my opinion, the combination of *Rao* and *Scahill* renders obvious 1[d]. *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. When an application issues a request for network communications from the launch manager through an API, the API will include as part of the input parameters of the request a desired execution time period for the execution of the network communications. The launch manager will handle this request in accordance with its policies and place the network communications in a queue with a scheduling time that has been selected and scheduled by the policies of the launch manager, taking into consideration the execution time requested as an input parameter in the request for network communications through an API.

87. According to *Rao*, “[i]n some embodiments, the agent 326 indicates the priority to the filter 322 for management of network packet queues 540a-540n to apply the indicated priorities.” *Rao*, [0190]. Based on the priorities indicated for applications 338a-338n, “filter 322 may place, arrange, or coordinate network packets into queues 540a-540n in support of the priorities.” *Rao*, [0191]. *Rao*

discloses that “filter 322 may move network packets from temporary queues 540a-540n or from memory or other storage to a queue 540a-540n associated with the application 338a-338n, a queue 540a-540n associated with a priority, or a queue 540a-540n associated with both an application and a priority.” *Rao*, [0191]. *Rao* provides additional examples of the many potential ways to queue applications. *Rao*, [0191]. According to *Rao*, “[o]ne ordinarily skilled in the art will recognize and appreciate that the network packets may . . . be placed or arranged in the queue in any suitable manner in practicing the operations of the present invention described herein.” *Rao*, [0191].

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. *See* Section VII.A.4.d (claim 1(c)). Once queued, remote access client 120 communicates the network packets. *Rao*, [0194], [0195]. The result is that filter 322 selects corresponding execution times for the applications requesting execution time in accordance with respective network access policy controls for each application.

5. **Claim 2: “The wireless end-user device of claim 1, the activity classifier dynamically associating network access policy controls with each of the plurality of executable applications using different policies that apply to blocked activities, network access controlled activities, and network access non-controlled activities.”**

89. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 2. *Rao* discloses that remote access client can indicate an access activity type for an application based on whether the application is running in the background (i.e., “network access controlled activities”) or foreground (i.e., “network access non-controlled activities”). These understandings of the claim terms are consistent with their usage in the ’700 patent. ’700 patent, 64:13-21 (describing a “background services classification” as “protecting network capacity”); 108:1-4 (listing an application in the foreground as an example of “network access non-controlled activity”).

90. *Rao* discloses that “policies 520 define prioritization based on whether an application is running in the foreground or the background of the client 105.” *Rao*, [0182]. The remote access client’s “agent 326 [(i.e., the activity classifier)] uses the policies 520 to apply a priority to network packets of applications 338a-338n in accordance with the prioritization rules specified or indicated by the policies 520.” *Rao*, [0189]. Agent 326 may also “use characteristics of the

application 338a-338n, such as running in the foreground or background, to indicate priority for a network packet of the application 338a-338n.” *Rao*, [0189].

91. *Rao* also discloses that agent 326 also associates a policy for “blocked activities.” For example, *Rao* discloses that “[i]n some embodiments, one or more queues 540a-540n may be used for network packets intercepted but not prioritized because . . . policies 520 indicate to ignore or not process the network packet for prioritization.” *Rao*, [0193]. Because *Rao*’s agent 326 would indicate that a network packet should be ignored or not processed, the remote access client would decline to send network packets from the blocked application, which would eventually cause network packets to be deleted from its queue. By using agent 326 to associate a policy that requires ignoring or declining to process a network packet, *Rao* discloses an activity classifier that associates a policy based on blocked activities.

92. As discuss in Section VII.A.4.c (claim 1(b)), *Rao/Scahill*’s policies are associated “dynamically” because, for example, the policy may depend on an application’s present characteristics, such as whether the application is running in the foreground or is running in the background. *Rao*, [0188]. In my opinion, a POSITA would have understood that these characteristics (as well as the other disclosed characteristics) change over time, requiring dynamic association by remote access client 120. A user may frequently switch between foreground

applications, and the remote access client must be able to quickly adapt to prioritize the applications based on their changing status.

6. Claim 4: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on a user’s current interaction with that application.”

93. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 4. As *Rao* explains, “the remote access client 120 [via agent 326] may determine whether the application 338a-338n associated with the network packet is running in the foreground . . . of the client 105.” *Rao*, [0188]; *see also Rao*, [0189] (disclosing how “agent 326 may use characteristics of the application 338a-338n, such as running in the foreground or background”). As discussed in Section VII.A.5 (claim 2), an application running in the foreground is a “network access non-controlled activity.”

94. A POSITA would have understood that *Rao/Scahill* classifies an application as “running in the foreground” because of a user’s current interaction with that application. According to *Rao*, an “application is running in the foreground” where it is “in active use by the user.” *Rao*, [0003]. Similarly, *Scahill* describes “foreground applications” as “applications that the user is interacting with.” *Scahill*, 36:4-8. These disclosures are each consistent with other disclosures known before the priority date of the ’700 patent. *Freund*, 3:12-16 (explaining

“‘active’ use of the Internet” as “when user interaction with the PC causes the Internet access”); *Freund*, 10:17-43 (“A given application itself can be examined for determining whether it is ‘active’ by determining whether the application receives ‘focus’ and/or receives user input (e.g., mouse clicks or key strokes).”); *Oestvall*, [0006] (“A foreground application conversely does present a user interface with which a user can interact.”).

95. Alternatively, a POSITA would have been motivated and found it obvious to modify *Rao*’s architecture so that agent 326 determines that an application is running in the foreground “based on a user’s current interaction with that application.” As discussed above, user interaction was a widely recognized indication of whether an application was running in the foreground. *Freund* discloses the benefit of using user interaction to determine whether a given application is in the foreground is that “[a] given application itself can be examined for determining whether it is ‘active’ by determining whether the application receives ‘focus’ and/or receives user input (e.g., mouse clicks or key strokes).” *Freund*, 10:17-43. A POSITA would have understood that user input (i.e., user interaction) is an easily implementable indicator to determine whether an application is in the foreground and would have been within the skill of an ordinary artisan to implement.

7. **Claim 5: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on that application currently providing a foreground service.”**

96. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 5. As discussed for claim 4 in Section VII.A.6 (claim 4), *Rao/Scahill* discloses the activity classifier determining that a given application is a network access non-controlled activity by virtue of that application running in the foreground (that is, providing a foreground service). *Rao*, [0003]; *Scahill*, 36:4-8.

8. **Claim 7: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network capacity demand as part of a request for an execution time.”**

97. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 7. As discussed in Section VII.A.3 (Motivation), a POSITA would have been motivated to improve *Rao*’s architecture to include *Scahill*’s scheduling method. This method includes an application’s request to schedule an execution time via an API.

98. As explained in Section VII.A.4.d (claim 1(c)), *Scahill*, discloses that an application’s request specifies “application specific requirement(s)” (also called “connection conditions”) for the application, “which the application scheduler library component 14a,b of each client application 12a,b . . . must take into account

when seeking to schedule performance of a task.” *Scahill*, 19:3-7, 24:14-17. One such connection condition in *Scahill* is a “connection bandwidth exceeding a minimum value.” *Scahill*, 19:3-7; *see also* *Scahill*, 19:9-11.

99. For example, when requesting an execution time via an API, “client application 12a first provides a tuple of data,” including “the required network profile (NP).” *Scahill*, 29:13-16; *see also* Section VII.A.4.d (claim 1(c)) (discussing use of API). *Scahill* discloses that “the network profile for a particular task comprises one or more of the following fields: Network type, Min Uplink BW, Min Downlink BW, Max RTT, Max PacketLoss, and/or Max Jitter.” *Scahill*, 42:21-23. *Scahill* explains that “each field may be optionally specified by the client application (or the part of the client application generating the connection request).” *Scahill*, 42:25-27. *Scahill* discloses that, “in one embodiment, an application requiring to transfer 1 Mb over a min 128 kb/s downlink over a WLAN network could specify a network profile as follows: ‘WLAN, -1, 128000,-1,-1,-1,-1’ with delay of 172800 secs and duration of 62 secs.”² *Scahill*, 42:28-31.

100. In my opinion, *Scahill* therefore discloses that the scheduler receives, from an application, a connection bandwidth (“network capacity demand”) as part

² According to *Scahill*, “non-specified fields” can be given a default value, such as “-1.” *Scahill*, 42:27-28.

of its request for an execution time via an API. From this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*'s architecture such that *Rao/Scahill*'s application scheduler receives, from an application, a connection bandwidth as part of its request for an execution time via an API. As *Scahill* explains, a network connection attempt that does "not coincide with appropriate network conditions to support the type of data transmission required to complete the scheduled task . . . needlessly wastes the resources of the device and . . . network resources have also been wasted." *Scahill*, 2:14-20. By receiving the connection bandwidth in the application's request, *Rao/Scahill*'s application scheduler schedules an execution time that "coincide[s] with appropriate network conditions." *Scahill*, 2:24-25.

9. Claim 8: "The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network request as part of a request for an execution time."

101. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 8. As discussed for claim 7, when sending a request for an execution time via an API, "client application 12a first provides a tuple of data," including "the required network profile (NP)." *Scahill*, 29:13-16. *Scahill* discloses that, "the network profile for a particular task comprises one or more of the following fields: Network type, Min Uplink BW, Min Downlink BW, Max RTT, Max PacketLoss, and/or Max Jitter." *Scahill*, 42:21-23. As one example, "an application requiring to

transfer 1 Mb over a min 128 kb/s downlink over a WLAN network could specify a network profile as follows: ‘WLAN, -1, 128000,-1,-1,-1,-1’ with delay of 172800 secs and duration of 62 secs.” *Scahill*, 42:25-31. Thus, in this example, the API receives from an application a request for a WLAN network as part of a request for an execution time.

102. In my opinion, from this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*’s architecture such that *Rao/Scahill*’s application scheduler likewise receives, from an application, a network request as part of its request for an execution time via an API. By receiving the network request in the application’s request, *Scahill*’s scheduler can schedule an execution time that “coincide[s] with appropriate network conditions.” *Scahill*, 2:24-25.

10. Claim 9: “The wireless end-user device of claim 8, wherein the network request specifies whether the network request specifies one of a roaming network or a non-roaming network as the connected network for the network request.”

103. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 9. As discussed with respect to claim 8, applications in *Rao*’s architecture as modified by *Scahill* can request a “[n]etwork type” as part of the request for an execution time via an API. These network types include either a non-roaming network (e.g., a network associated with a home network operator) or a roaming

network (e.g., a partner of a home network operator that can provide roaming services).

104. *Scahill* discloses that its scheduling method improves on prior art scheduling systems that struggle to schedule transmissions under “appropriate network conditions to support the type of data transmission required to complete the scheduled task.” *Scahill*, 2:14-17. *Scahill* teaches that in prior art systems, “under certain conditions, e.g., if located in a roaming data network, the user may be on a tariff which would mean that each aborted data transmission still incurred a cost for the user.” *Scahill*, 2:20-22. *Scahill* makes clear that its scheduling method addresses this issue by allowing an application to specify a particular network to avoid roaming costs. *Scahill*, 29:13-16, 42:21-23.

105. *Scahill* discloses an example of a network request for a home, non-roaming network, such as a WLAN: “an application requiring to transfer 1 Mb over a min 128 kb/s downlink over a WLAN network could specify a network profile as follows: ‘WLAN, -1, 128000,-1,-1,-1,-1’ with delay of 172800 secs and duration of 62 secs.” *Scahill*, 42:25-31. In my opinion, a POSITA would have understood that *Scahill*’s “WLAN” request is a request for a non-roaming network because the user may access the network without roaming network charges, while allowing the user to switch to a different access point within the home WLAN using wireless mesh technology.

106. In addition, in my opinion, a POSITA would also have been familiar with the distinction between a non-roaming network and a potentially costly roaming network in the WWAN context:

A wireless device is typically associated with a home network operator and, when available, uses the home network for wireless communications. The home network operator may have partner network operators that provide roaming service to the wireless device. That is, the roaming network operators permit the wireless device to connect to the home network via the roaming network when the wireless device is outside of the device's home network's coverage area.

Srinivasan, [0003]. A POSITA would have understood that the cost of using a roaming network can far exceed that of a non-roaming network. *Scahill*, 2:20-22 (disclosing that “if located in a roaming data network, the user may be on a tariff which would mean that each aborted data transmission still incurred a cost for the user”); *Cole*, [0008] (“Also, the bandwidth available for each scheme may vary as well as the cost of using each scheme. . . . In cases where the connection involves a usage based fee schedule, operation of bandwidth intensive applications may significantly affect the user's service bill.”).

107. The result is that a POSITA would also have been motivated to implement a network request to specify the need for the more cost-effective, non-

roaming WWAN network for certain tasks. For example, where an application requires transmission of a significant source of data (e.g., the download of a large media file), it would be desirable for the application to be able to specify the use of a non-roaming network to minimize costs for the user. *Cole*, [0008]. A POSITA would thus have been motivated to implement *Scahill*'s network requests to allow for an application to account for these cost differences.

108. In my opinion, a POSITA would thus have been motivated and found it obvious to implement *Rao/Scahill*'s network requests (as discussed in claim 8) to allow an application to specify one of a roaming network or a non-roaming network to use for the network request. Such a modification would be well within a POSITA's abilities (e.g., adding a single field to *Scahill*'s network profile requests), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification.

11. Claim 10: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a scheduled time request as part of a request for an execution time.”

109. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 10. As explained in Section VII.A.4.d (claim 1(c)), *Rao/Scahill* discloses that an application's request specifies “application specific requirement(s)” (also called “connection conditions”) for the application, “which the application scheduler

library component 14a,b of each client application 12a,b . . . must take into account when seeking to schedule performance of a task.” *Scahill*, 19:3-7, 24:14-17.

110. One such connection condition is a “timing interval Δt requested by the client application since the task was last performed.” *Scahill*, 24:19-23. For example, an application could specify a “delay of 172800 secs and duration of 62 secs.” *Scahill*, 42:25-31. A POSITA would have understood *Scahill*’s timing interval Δt to be a “scheduled time request” because it is a request for a particular time, as indicated by the delay of precise duration (specified in seconds) after the scheduled connection window (the “period of time of predetermined duration” that was scheduled at “an absolute time value”). *Scahill*, 19:25-26, 26:17-23.

111. In my opinion, *Scahill* therefore discloses that the application scheduler receives, from an application, a timing interval (“scheduled time request”) as part of its request for an execution time via an API. From this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*’s architecture such that *Rao/Scahill*’s application scheduler receives, from an application, a timing interval as part of its request for an execution time via an API. A POSITA would have recognized that, by receiving the timing interval in the application’s request, *Rao/Scahill*’s application scheduler schedules an execution time for the next performance of the application’s task that is appropriate for the task. As *Scahill* explains, “[t]he timing interval Δt may vary for

some client applications,” e.g., depending on the “amount of data” that “require[s] backing up” or dependency among tasks. *Scahill*, 24:26-32.

12. Claim 11: “The wireless end-user device of claim 1, further comprising a user interface allowing a user to specify one or more of the network access policy controls to be associated with a given one of the plurality of executable applications.”

112. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 11. According to *Rao*, “policies 520 may be configurable by a user by any suitable means and/or mechanism. For example, the agent 326 may provide a configuration mechanism such as a user interface, graphical or otherwise, design[ed] and constructed for configuring or specifying the policies 520.” *Rao*, [0183]. These policies 520 are intended to be associated with one of applications 338a-338n. *Rao*, [0182], [0189].

13. Claim 13: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a device usage state.”

113. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 13. *Rao* discloses that, in addition to making a determination based on whether the application is in the foreground, *Rao*’s remote access client “may determine any other characteristics or statistics of the application 338a-338n [for the purpose of assessing priority] such as ... total execution time, and/or frequency

of use.” (*Rao*, [0188]-[0189].) A POSITA would have understood these characteristics reflect a “device usage state” because they reflect how the device is being used on an application-specific basis.

114. From *Rao/Scahill*, a POSITA would have understood that, based on these “device usage state” characteristics, *Rao/Scahill*’s remote access client may dynamically change an association of network access policy controls. For instance, when the application requesting network access is in the background, *Rao/Scahill*’s remote access client may associate a policy to deprioritize that application (and prioritize foreground applications) in light of how the device is being used (e.g., a total execution time or frequency of use does not exceed a predetermined threshold). *Rao/Scahill*’s remote access client may then dynamically change the associated policy to prioritize that background application if how the device is being used necessitates a change in policy (e.g., the total execution time or frequency of use now exceeds a predetermined threshold) to prevent starvation of the background application.

14. Claim 15: “The wireless end-user device of claim 1, wherein the launch manager receives a list identifying executable applications that have a launch control policy in effect.”

115. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 15. *Rao* discloses that “filter 22” (i.e., “launch manager”) receives an

indication of the applications that have one of policies 520 (“launch control policy”) in effect.

116. *Rao* discloses identifying the applications that have a policy in effect because, in *Rao*, “the policies 520 may be specified by the name of the application 338a-338n.” *Rao*, [0182]. *Rao* discloses that filter 322 (“launch manager”) receives a list identifying these applications because “[t]he policies may be accessible by the agent 326, configured into the agent 326, or loaded by the agent 326,” and “the agent 326 indicates the priority to the filter 322 for management of network packet queues 540a-540n to apply the indicated priorities.” *Rao*, [0183], [0190]. A POSITA would have understood such an indication of the policies, specified by the name of the application, to be the claimed “list.”

117. *Rao* discloses that remote access client 120 may “use a routing table 538 for determining how to route network packets of the client via the agent 326.” *Rao*, [0181]. According to *Rao*, “[i]n one embodiment, the filter 322 is not aware of the application 338a-338n by name but can associate priorities to network packets of applications 338a-338n via the routing table 538.” *Rao*, [0190]. *Rao* explains that “[a]n application 338a-338n corresponding to a network packet can be identified by any combination of source and destination identifiers.” *Rao*, [0190]. “As such, in some embodiments, the agent 326 indicates priorities to the

filter 322 by routing information instead of application name.” *Rao*, [0190]. *Rao* discloses that, “the agent 326 provides to the filter 322 a mapping between applications 338a-338n, such as by application name or process id, to the routing information in a routing table 538.” *Rao*, [0190]. Thus, in *Rao/Scahill*’s architecture, filter 322 receives a list identifying executable applications that have a launch control policy in effect.

15. Claim 16: “The wireless end-user device of claim 1, wherein the launch manager is integrated with an operating system broadcast intent manager.”

118. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 16. The ’700 patent does not claim to have invented an “operating system broadcast intent manager.” And it does not explain how to integrate a launch manager into an operating system. Instead, the ’700 patent discloses that the claimed techniques “that allow for intercepting a service activity intention to launch, and applying a background service policy set or a network protection service policy set can be designed into the OS itself.” ’700 patent, 67:33-37. The ’700 patent provides an example, disclosing that “the intercept and policy implementation functions can be designed into the activity manager, broadcast intent man[a]ger, media service manager, service manager, or other application or service activity management function in the Android OS.” ’700 patent, 67:37-41. In my opinion, the ’700 patent therefore recognizes that the claimed “broadcast

intent manager” is a known type of application management function in an operating system such as the iPhone OS or Android OS. It is not an element that was allegedly invented by the inventors of the ’700 patent.

119. The ’700 patent goes on to acknowledge that “[o]ne of ordinary skill in the art will recognize that . . . [the claimed prioritization techniques] can be designed into application launch management functions in the iPhone OS, windows mobile OS, windows PC OS, Blackberry OS, Palm OS, and other OS designs.” ’700 patent, 67:41-49. But the ’700 patent does not provide any details regarding this integration. Instead, the ’700 patent appears to have assumed that this integration of the claimed launch manager was something known in the art and within the level of skill of a POSITA.

120. As discussed in Section VII.A.4.e (claim 1(d)), in *Rao/Scahill*’s architecture, filter 322 selects and schedules execution times and is the claimed “launch manager.” *Rao* discloses that its devices “typically operate under the control of operating systems, which control scheduling of tasks and access to system resources.” *Rao*, [0128]. *Rao* further discloses that its device can run any known commercial operating systems: “The computing device 102 can be running any operating system such as . . . any embedded operating system, any real-time operating system, any open source operating system, any proprietary operating system, any operating systems for mobile computing devices, or any other

operating system capable of running on the computing device and performing the operations described herein.” *Rao*, [0128].

121. In my opinion, a POSITA would thus have been motivated to implement *Rao/Scahill*’s architecture in known, commercially available operating systems. As of the ’700 patent’s priority date, this would have included Android OS. *Huq*, 7:14-17. A POSITA would have understood that *Rao*’s filter would be implemented into the component of the operating system that schedules tasks and access to system resources. In the case of Android OS, this would include the “broadcast intent manager” for managing scheduling and intents. ’700 patent, 67:37-41. The result is that a POSITA would have understood how to implement *Rao*’s filter (the “launch manager”) into the broadcast intent manager of an operating system.

16. Claim 17: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager deprioritizes one or more processes associated with a given one of the executable applications.”

122. In my opinion, the combination of *Rao* and *Scahill* renders obvious claim 17. As discussed in Section VII.A.4.e (claim 1(d)), filter 322 (i.e., “launch manager”) arranges network communications from the applications into a queue according to their respective policies. As an example, policies 520 may “define prioritization based on whether an application is running in the foreground or the

background of the client 105.” *Rao*, [0182]. *Rao* discloses that “policies 520 may be specified conditionally, such as if one application 338a is running, a second application 338b may have a higher or lower priority.” *Rao*, [0182]. Thus, for example, where a first application moves from the foreground to the background and a second application moves from the background to the foreground, a POSITA would have understood that filter 322 would deprioritize the first application (and its associated processes or network packets) based on the policies identified by agent 326. *Rao*, [0189], [0190]. A POSITA would have understood that prioritizing one application over another is equivalent to deprioritizing one application below another, since prioritizing and deprioritizing are actions that determine the priority of one application relative to another application.

B. Ground 2: Claim 3 Is Obvious Over *Rao* in view of *Scahill* and *Oestvall*

1. Overview of the *Oestvall* Reference

123. *Oestvall* is directed to techniques to “preserve or conserve resources” on mobile devices. *Oestvall*, [0002]. Like the ’700 patent, *Oestvall*’s teachings relate to “multi-tasking devices,” such as “devices with an operating system that can run several applications at the same time.” *Oestvall*, [0007]. This includes foreground and background applications. *Oestvall*, [0006]. In my opinion, because *Oestvall* is directed to the same field of endeavor as the ’700 patent (application

management in devices that can run multiple applications), *Oestvall* is analogous art to the '700 patent.

124. According to *Oestvall*, conventional schedulers used for multitasking may be configured such that “applications will continue to run even when not actually in active use,” which results in the applications continuing to “use some system resources, even when residing in the ‘background’.” *Oestvall*, [0006]. *Oestvall* addresses these resource challenges using techniques that “deny[] system resources and services to background applications that do not meet predefined ‘trust’ or certification criteria.” *Oestvall*, [0010].

2. Rationale for Modifying *Rao* in view of *Scahill* and *Oestvall*

125. *Rao* contrasts “an application [that] is running in the foreground and currently in active use by the user” with “an application running in the background.” *Rao*, [0003]. *Rao*, however, does not expressly describe how to determine whether an application is running in the background. As a result, a POSITA implementing *Rao*’s teachings (in the context of *Rao/Scahill*’s combination discussed above) would have looked to similar references, including *Oestvall*, to understand how to classify an application as running in the background.

126. *Oestvall* teaches “[a]n application is in the background if it is not being interacted with by an end-user and it presents no user interface with which a

user could interact.” *Oestvall*, [0006]. A POSITA would have understood that *Oestvall* addresses concerns associated with conventional classification methods. *Freund*, 8:63-9:3, 10:5-43. *Oestvall*’s approach would advantageously allow *Rao/Scahill*’s architecture to more accurately classify background applications and enable *Rao/Scahill*’s prioritization based on whether an application is running in the foreground or background. A POSITA would have been motivated to identify applications running in the background because, as *Rao* and *Oestvall* both recognize, such background applications can cause latency that affects foreground applications. In my opinion, a POSITA would have understood that the proposed combination involves (1) combining prior art elements according to known methods to yield predictable results; and (2) use of a known technique to improve similar devices in the same way.

127. In my opinion, a POSITA would have also had a reasonable expectation of success in modifying *Rao/Scahill*’s architecture in view of *Oestvall*’s classification method. *Rao*’s device already evaluates whether a device is in active use (e.g., to classify whether an application is running in the foreground), and *Oestvall*’s classification method only requires the additional step of determining that the application presents no user interface with which a user could interact. Such a modification would have been well within a POSITA’s abilities to implement.

3. Claim 3: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access controlled activity or a blocked activity based on a lack of user interaction with that application for a period of time.”

128. In my opinion, the combination of *Rao*, *Scahill*, and *Oestvall* renders obvious claim 3. *Rao* discloses that remote access client 120, including agent 326 (“activity classifier”), determines whether an application associated with the network packet is running in the background. *Rao*, [0188]; *see also Rao*, [0189]. As discussed in Section VII.A.5 (claim 2), a background application is a “network access controlled activit[y]” because it may be deprioritized for network access.

129. As discussed above, a POSITA would have been motivated to modify *Rao/Scahill*’s architecture to classify whether an application is running in the background based on *Oestvall*’s assessment that an application “is not being interacted with by an end-user and it presents no user interface with which a user could interact.” *Oestvall*, [0006]. This assessment from *Oestvall* would have been evaluated over a set period of time (rather than happening instantaneously) so as to ensure that a user was indeed not interacting with a particular application. *Wyld*, 2:38-43, 4:49-57, 5:12-19 (discussing the benefits of using a predefined idle time prior to executing background tasks). In *Rao/Scahill*’s architecture, filter 322 classifies whether an application is running in the background based on a determination that an application is not being interacted with by an end-user for a

period of time and the application presents no user interface with which a user could interact.

C. Ground 3: Claims 6, 14, 18, and 19 Are Obvious Over *Rao* in view of *Scahill* and *Montemurro*

1. Overview of *Montemurro*

130. *Montemurro* is directed to application-aware optimization of network transmissions from a mobile device. Just as *Rao* and *Araujo* teach application-aware prioritization of network access, *Montemurro* discloses application-aware “policy-based routing of communications among two or more modes of wireless communication.” (*Montemurro*, [0001].) In my opinion, because *Montemurro* is directed to at least the same problem as the ’700 patent (optimizing use of network resources on an application-by-application basis), *Montemurro* is analogous art to the ’700 patent.

131. Devices with such connectivity capabilities are “dual or multi-mode devices,” with “radio access technologies that provide access to” multiple network types, including WLAN and WWAN. *Montemurro*, [0003]. But, according to *Montemurro*, “[t]here are costs associated with application access from these different networks,” and thus, “[i]t is therefore desirable to have a mechanism that seeks to optimize communications.” *Montemurro*, [0004].

132. To optimize communications, *Montemurro* discloses “a rules engine 206A (sometimes called an inference engine) as a policy-based mechanism using rules to configure TCP/IP network stack 206B and connection table 206C to coordinate communications for applications 204 using communications interfaces (also referenced as network interfaces) 208.” *Montemurro*, [0027]. *Montemurro* discloses that this rules engine “configures the communication operations with a set of rules/policies that could include various factors such as radio access technology (e.g. for high/low bandwidth properties), cost, presence, time of day, location (e.g. geo-based policies, network roaming), destination IP address, application type, and Quality of Service (QoS) requirements, among others.” *Montemurro*, [0027].

133. Rules engine 206A “configures (i.e. modifies, periodically in response to changes of state and the evaluation of its rules) connection table 206C and routing table 210 to optimize the flow of communications over multiple communications modes (e.g., interfaces 208 and respective networks 104 and 106).” *Montemurro*, [0034]. On a state change, e.g., “time of day,” “connecting/disconnecting of device 102 with a specific network 104 and 106,” rules engine 206A executes and modifies “routing table 210 to ensure that data goes out [to] the most appropriate network (via respective network interface 208).”

Montemurro, [0035]. Rules engine 206A further “determine[s] which interface would be best used to service a particular application.” *Montemurro*, [0035].

2. Rationale for Modifying *Rao* in view of *Scahill* and *Montemurro*

134. As described in Section VII.A.3 (Motivation), a POSITA would have been motivated to improve *Rao*’s architecture by including *Scahill*’s scheduling method. *See Rao*, [0179]-[0195]. In implementing this modification, a POSITA would have recognized opportunities to provide further optimization in scenarios where *Rao/Scahill*’s device has access to multiple networks.³ *Rao*, [0095] (describing the ability to connect to multiple networks); *see also Montemurro*, [0003], [0004]. This would have motivated a POSITA to seek solutions beyond *Rao* and *Scahill* to achieve such additional efficiencies.

135. According to *Montemurro*, devices with multiple radio access technologies can access more than one network, including at least “wireless local area networks (WLAN)” and “wireless wide area networks (WWAN) (e.g. cellular technologies like GSM/GPRS EDGE, UTMS, HSPA, CDMA, WCDMA, etc.).”

³ Although *Scahill*’s scheduling method allows an application to request a particular network (e.g., claim 8), this network request is driven by the application—not the device (i.e., remote access client). *Montemurro*’s user-configurable network policies allow the remote access client (or a user through a user interface) to optimize the communication flow over multiple communications modes. *Montemurro*, [0036] (disclosing a user interface).

Montemurro, [0003]. But, according to *Montemurro*, “[t]here are costs associated with application access from these different networks.” *Montemurro*, [0004]. And “[t]here are also quality considerations such as the speed at which content can be delivered, or in the case of streaming media, the quality of service at which the content is delivered.” *Montemurro*, [0004]. As a result, according to *Montemurro*, “[i]t is therefore desirable to have a mechanism that seeks to optimize communications for multi-mode capable devices, that is, that seeks to improve communications for multi-mode capable devices.” *Montemurro*, [0004].

136. Given *Rao*’s disclosure of multi-mode devices that “can connect to one or more networks 104” (*Rao*, [0095]), a POSITA would have been motivated to further improve *Rao/Scahill*’s architecture by augmenting *Rao/Scahill*’s policies to include *Montemurro*’s policy-based data routing for multimode operations. A POSITA would have understood that incorporating these teachings into *Rao/Scahill* would have yielded several improvements, such as the ability to “offer different usage models depending on the mode of wireless operation selected.” *Montemurro*, [0003]. *Montemurro*’s policies would have enabled *Rao*’s prioritization architecture to further account for “factors such as radio access technology (e.g. for high/low bandwidth properties), cost, presence, time of day, location (e.g. geo-based policies, network roaming), destination IP address,

application type, and Quality of Service (QoS) requirements, among others.”

Montemurro, [0029].

137. In my opinion, a POSITA would have had a reasonable expectation of success in modifying *Rao/Scahill*’s architecture in view of *Montemurro*’s policies to augment *Rao/Scahill*’s existing policies. *Rao* discloses that “[o]ne ordinarily skilled in the art will recognize and appreciate the multitude of ways to define client-side application priorities.” *Rao*, [0182]. *Rao* explains that policies 520: (1) “may be provided by or downloaded via the gateway 340,” (2) “may comprise any type and format of syntax and/or language for specifying a policy,” and (3) “may be provided via any type and/or form of medium.” *Rao*, [0183]. These policies 520 also “may be configurable by a user by any suitable means and/or mechanism.” *Rao*, [0183]. The result is that the modification would have been well within the abilities of an ordinary artisan to implement.

138. In my opinion, a POSITA would have therefore found it obvious to modify *Rao* in view of *Araujo* and *Montemurro* and would have understood that the proposed combination involves (1) combining prior art elements according to known methods to yield predictable results; and (2) use of a known technique to improve similar devices (methods, or products) in the same way.

- 3. Claim 6: “The wireless end-user device of claim 1, wherein according to at least one of the network access policy controls, the launch manager delays an execution time for one or more of the applications until a connection is made through the at least one wireless modem to an alternative network.”**

139. In my opinion, the combination of *Rao*, *Scahill*, and *Montemurro* renders obvious claim 6. As discussed in Section VII.C.2 (Motivation), a POSITA would have been motivated by *Montemurro* to augment *Rao*’s policies to additionally account for “radio access technology (e.g. for high/low bandwidth properties), cost, presence, time of day, location (e.g. geo-based policies, network roaming), . . . and Quality of Service (QoS) requirements.” *Montemurro*, [0029]. In particular, a POSITA would have been motivated to augment *Rao*’s policies to allow an application differing network access on different networks. Such augmented policies would cause a launch manager to delay an application’s execution time until a connection is made through a device’s modem to an alternative network.

140. For example, a POSITA would have been motivated to augment *Rao*’s policies to allow an application to have differing network access over a WLAN network than over a WWAN network. When a WLAN network is available, an application’s network access may be specified by a policy that permits large transmissions of data, while when only a WWAN network is

available, an application's network access may be specified by a policy that does not permit large transmissions of data. According to *Montemurro*, a "CRM application (i.e. one of 204B) only synchronizes the sales contact database (e.g. 120B) over a WLAN network 106." *Montemurro*, [0029]. In *Rao/Scahill/Montemurro*'s architecture, agent 326 would identify a radio access technology policy for a CRM application, and filter 322 ("launch manager") would delay executing the CRM application synchronization while *Rao*'s is connected to a WWAN, to minimize data usage. But, once the device connects to a home WLAN, filter 322 would allow execution of the application under the network policy.

141. The claimed "delay" was well known in the art. For example, the prior art *Cole* reference disclosed delaying data transfer until a "cheaper or free network is available": "Another situation where load monitoring is useful is when a communication link is being heavily utilized, and that link requires payment, either by units of time or data transfer. In cases where the load is expected to result in appreciable network costs, the connection manager 260 may determine if a free, but possibly slower, network is available and prompt the user to determine if the connection should be switched to reduce cost. Alternately, the connection manager 260 may suggest that the data transfer be postponed to a time when a cheaper or free network is available." *Cole*, ¶ [0082].

4. **Claim 14: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a change in a current network to which the at least one wireless modem is providing network access.”**

142. In my opinion, the combination of *Rao*, *Scahill*, and *Montemurro* renders obvious claim 14. As discussed in Section VII.C.2 (Motivation), a POSITA would have been motivated by *Montemurro* to augment *Rao*’s policies to specify one policy for an application’s network access over a WLAN network and another policy for an application’s network access over a WWAN network to account for “various factors such as radio access technology (e.g. for high/low bandwidth properties)” and “location (e.g. geo-based policies, network roaming).”

Montemurro, [0029]. In my opinion, a POSITA would have understood that a radio access technology policy or a location-based policy focused on network roaming would meet the features of claim 14. *See* Section VII.A.10 (claim 9) (discussing network roaming). As a mobile device moves from one location within a home network (e.g., WLAN) to a second, out-of-network location (e.g., WWAN or roaming WWAN), the corresponding network access policy would change. Thus, for example, a “CRM application (i.e. one of 204B) only synchronizes the sales contact database (e.g. 120B) over a WLAN network 106.” *Montemurro*, [0029]. If the mobile device with the CRM application moves from one location with a

WWAN connection to a second location with a WLAN connection, the network access policy would change with the network change, allowing synchronization over the WLAN network. And as discussed in Section VII.A.4.c (claim 1(b)), such policy change would be dynamic; that is, the policy would change along with the change in location and associated change in network.

- 5. Claim 18: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager controls the frequency at which a given one of the executable applications is allowed network access.”**

143. In my opinion, the combination of *Rao*, *Scahill*, and *Montemurro* renders obvious claim 18. *Montemurro* discloses “a set of rules/policies that could include various factors such as radio access technology (e.g. for high/low bandwidth properties), cost, presence, time of day, location (e.g. geo-based policies, network roaming), destination IP address, application type, and Quality of Service (QoS) requirements, among others.” *Montemurro*, [0029]. As discussed in Section VII.C.2 (Motivation), a POSITA would have been motivated to modify *Rao/Scahill*’s architecture in view of *Montemurro*’s policies to augment *Rao/Scahill*’s existing policies. These policies (including those discussed above) would each enable filter 322 (“launch manager”) to control the frequency at which a given one of the executable applications is allowed network access.

144. For example, QoS requirements include the rate of transmission.

Different applications have different QoS requirements, including bit rates and variability in transmission times, including rate of transmission. For example, a voice application may require a transmission every 20 ms, but an e-mail application would require less frequent transmission to satisfy QoS requirements. Similarly, QoS can consider “latency variation” or jitter, both of which account for the stability of the transmission rate (i.e., frequency of transmissions). *Scahill* at 33:1-3 (listing “jitter” as a measure of connection quality), 45:8-13 (“the average difference between interpacket send time and interpacket receive time”). A POSITA would have understood that a policy directed to transmission rates would advantageously allow optimization of transmission on an application-by-application basis.

145. Additionally, a policy based on time of day, as disclosed by *Montemurro* and incorporated into the *Rao/Scahill* combination above, would alter the frequency at which an application is allowed network access to a particular network type. *Montemurro*, [0035]. This may include a policy that only allowed certain applications, such as applications requiring high bandwidth, to access a given network type (such as a cellular network or WLAN) during off-peak hours (e.g., before 9 am or after 5 pm) so as to optimize data usage. If that policy were

removed (or changed), the frequency at which that application is allowed network access would change accordingly.

146. In my opinion, based on the evidence above, a POSITA would have been motivated and found it obvious to augment *Rao/Scahill*'s policies in view of *Montemurro* to include policies that control the frequency at which that application is allowed network access. Such a modification would be well within a POSITA's abilities (e.g., adding additional policies to *Rao*'s remote access client), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification.

- 6. Claim 19: "The wireless end-user device of claim 1, wherein, based on an amount of network data communicated to or from a given one of the executable applications, the activity classifier dynamically changes at least one network access policy control associated with that given executable application."**

147. In my opinion, the combination of *Rao*, *Scahill*, and *Montemurro* renders obvious claim 19. *Montemurro* discloses "a set of rules/policies that could include various factors such as radio access technology (e.g. for high/low bandwidth properties), cost, presence, time of day, location (e.g. geo-based policies, network roaming), destination IP address, application type, and Quality of Service (QoS) requirements, among others." *Montemurro*, [0029]. As discussed in Section VII.C.2 (Motivation), a POSITA would have been motivated to modify

Rao/Scahill's architecture in view of *Montemurro*'s policies to augment *Rao/Scahill*'s existing policies. *Rao/Scahill*'s "agent 326," which operates as the claimed activity classifier, would dynamically apply cost policies based on an amount of network data communicated to or from a given one of the executable applications.

148. Indeed, a POSITA would have understood that different network connections have different bandwidth availability, different associated costs, and in some instances may only allow a fixed amount of bandwidth or data transmission time (also sometimes referred to as "minutes" in the cellular context) before incurring costs or the network administrator throttles the network speed. *Cole*, [0008], [0150], [0151] ("If the connection terms specify a fixed amount of data, the connection manager 260 will warn the user if they are near the threshold of data transfer."). The result is that, if a network access policy is cost-based, as taught by *Montemurro*, *Rao/Scahill*'s agent 326 would advantageously only allow an application to use a certain amount of data before dynamically associating a new network access policy for that application so as to prevent the user from incurring unexpected costs (e.g., by going over a fixed monthly data allotment). This application-specific limit might be, for example, a predetermined percentage of the overall fixed data allotment (for an entire device) so as to ensure that one

application does not alone lead to exceeding allowable limits and preserves network resources for other applications.

149. In my opinion, from this, a POSITA would have been motivated and found it obvious to augment *Rao/Scahill*'s policies in view of *Montemurro* to apply cost policies based on an amount of network data communicated to or from a given one of the executable applications. Such a modification would be well within a POSITA's abilities (e.g., adding additional policies to *Rao*'s remote access client), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification.

D. Ground 4: Claims 12 and 20 Are Obvious Over *Rao* in view of *Scahill* and *Araujo*

1. Overview of the *Araujo* Reference

150. *Araujo* is directed to application-aware optimization of power management on a mobile device. Just as *Rao* teaches application-aware prioritization for a mobile device and *Scahill* teaches application-aware scheduling to optimize transmission timing and reduce conflicts, *Araujo* teaches a complementary application-aware power management scheme for a mobile device. Like the '700 patent, *Araujo* implements policies to control activities of "wireless telephones" on an application-by-application basis. (*Araujo*, [0014].) In my opinion, because *Araujo* is directed to the same field of endeavor as the '700 patent

(application management in devices that can run multiple applications), *Araujo* is analogous art to the '700 patent.

151. *Araujo* describes “a power-management policy” for managing a “machine’s power usage.” *Araujo*, Abstract. Like the '700 patent, the policy can be implemented on “[e]lectronic devices,” such as “computers, wireless telephones, audio/video equipment, etc.” *Araujo*, [0014]. The policy is applied to “justify [a program] consuming power, depending on what the program is doing.” *Araujo*, [0004]. By implementing this policy, a device can better “use the energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” *Araujo*, [0001].

152. In *Araujo*, programs are classified using various criteria, and selectively provided resources based on their classifications. *Araujo*, [0027]. A program’s ability to consume power is “based on a finding as to whether the program’s status justifies the consumption of power under the circumstances that are present.” *Araujo*, [0027]. “For example, some programs may have a ‘VIP’ status that allows them to consume power in situations where other programs would not be permitted to consume power.” *Araujo*, [0027].

153. *Araujo* teaches that a device can provide resources to programs differently, based on whether the programs are “worth[y]” to consume power. *Araujo*, [0027]. A program can generate “a request to perform an action using a

particular device,” such as “a request to send data over a wireless network.”

Araujo, [0020], [0024]. The device can then perform various responsive operations, including allowing the request (*Araujo*, [0035], [0037], Fig. 3 (302)) or blocking the request (*Araujo*, [0035], [0040], [0050], Fig. 4 (406)). The responsive operations are provided by a “request deflector component” that is “interposed” between a “program” and a “[device] driver,” and configured to “intercept[] requests and respond[] to the requests based on power management considerations before the request reach[es] a device driver.” *Araujo*, [0024], [0050].

2. Rationale for Modifying *Rao* in view of *Scahill* and *Araujo*

154. It would have been obvious to a POSITA to modify *Rao/Scahill*’s architecture to include *Araujo*’s power management policy to provide a combination that better “use[s] the energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” *Araujo*, [0001].

155. Battery-powered devices, such as mobile phones, “have a finite amount of energy stored between charges.” *Araujo*, [0001]. As a result, these devices “generally attempt to use the energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” *Araujo*, [0001]. These mobile devices generally “employ power management schemes to govern the use of energy.” *Araujo*, [0001]. The prior art, however, recognized “a need . . . for an improved power management operation in portable devices and/or

portable wireless devices.” *Aleksic*, [0004]. This was especially true in the context of mobile devices running multiple applications and having access to multiple network connections. *D’Amore*, [0031]; *Araujo*, [0002]-[0005]. The prior art taught that one method of improving the power management in these devices was an application-aware prioritization of network communications that could optimize the power consumption of individual applications. *Araujo*, [0002]-[0005]; *D’Amore*, [0031]; *Rao*, [0080].

156. In addition to the benefits of extending the battery life of a mobile device’s battery by optimizing the power consumption of individual applications, a POSITA would have understood that improved, application-specific, power management schemes (that would, for example, include a reduced overlapping scheduling policy for each application) would also reduce competition among applications for other resources on a mobile device, including network resources and processing power. *D’Amore*, [0003], [0004], [0006]; *Oestvall*, [0021]; *Aleksic*, [0022]-[0025]. By selectively allowing applications to access network resources (including according to a scheduling policy), a power management scheme would reduce the competition among applications for those same network resources. *D’Amore*, [0006]. The result is that certain, prioritized applications, are allowed to access network resources at the expense of lower priority applications with greater efficiency, since the probability for an overlapping transmission to interfere with

the use of network resources is reduced. *D'Amore*, [0006]. This is particularly helpful, for example when accessing a network with lower bandwidth or running a high priority application requiring more processing power. Using a power management scheme and managing the power consumption of individual applications optimizes the consumption of other device resources, improving an application's operation and the user experience.

157. *Rao* recognized that its policy-based prioritization techniques “may improve or increase the performance, operational characteristics, and user experience on the client based on the applications running on the client.” *Rao*, [0080]. In my opinion, a POSITA would have understood that “performance, operational characteristics, and user experience” each implicate prioritization to improve the longevity of the battery in the device and availability of network resources. The result is that a POSITA would have been motivated to modify *Rao/Scahill* to include well-known, application-aware, power monitoring capabilities that had the potential to reduce power consumption, decrease energy expenses, optimize the allocation of network resources, and increase the longevity of *Rao*'s devices while operating on battery power.

158. *Araujo* discloses one example of a power management scheme that optimizes power consumption by applying application-aware prioritization of network communications based on a power state of a device. *Araujo*, [0003]-

[0005]. *Rao*'s architecture, as modified in view of *Araujo*'s power management policy, uses "energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge." *Araujo*, [0001].

159. To implement these improvements, a POSITA would have looked to *Araujo*'s disclosure to selectively allow or block an application from transmitting data to the network depending on various factors, including the power state of the device and the "status" of the particular application requesting network access. *Araujo*, [0025], [0027], [0028], Fig. 2. Using this functionality, *Rao/Scahill*'s architecture would assign a "status" to an application, use that status to determine if allowing an application request "justifies the consumption of power under the circumstances that are present," and regulate (e.g., blocking, allowing) the request based on that determination. *Araujo*, [0025], [0027].

160. *Araujo*'s request deflector functionality may be implemented in connection with or as part of the filter 322 of *Rao*'s remote access client, which—like *Araujo*'s request deflector—applies application-specific policies to determine how to handle applications' network access attempts. *Araujo*, [0024], [0050]; *Rao*, [0191]-[0193]. The request deflector determines whether applications have a "status [that] justifies the consumption of power under the circumstances that are present," and selectively blocks packets from proceeding to the network driver based on the determination. *Araujo*, [0027].

161. As an example, upon determining that an application's status justifies power consumption, the "request deflector component" would allow the network packets to proceed to the network driver for prioritization. Alternatively, upon determining that the application's status does not justify power consumption, the "request deflector component" would block the network packets from reaching the network driver. The "request deflector component" would provide an additional layer of control for regulating each of the applications' access to the network and enable implementation of power management policies.

162. Finally, in my opinion, a POSITA would have had a reasonable expectation of success in modifying *Rao* to incorporate *Araujo*. Modifying *Rao/Scahill*'s architecture to include *Araujo*'s power management functionality would have been well within the abilities of a POSITA. For example, *Araujo* discloses that its power management scheme "can be implemented as software that is stored in one or more of the data remembrance component(s) 704 and that executes on one or more of the processor(s) 702." *Araujo*, [0055]. This scheme could thus be implemented using the processor and memory units in *Rao*'s architecture. See *Rao*, [0119] (discussing processors), [0120] (discussing main memory unit). The resulting modification to *Rao*'s architecture would have involved combining prior art elements according to known methods to yield

predictable results and use of a known technique to improve similar devices in the same way.

3. Claim 12: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a power state of the device.”

163. In my opinion, the combination of *Rao*, *Scahill*, and *Araujo* renders obvious claim 12. According to *Araujo*, “[e]lectronic devices (e.g., computers, wireless telephones, audio/video equipment, etc.) normally employ some type of power management framework.” *Araujo*, [0014]. *Araujo* uses the term “power state” to describe the power condition of a device, such as a “wireless telephone.” *Araujo*, [0004], [0014] (referring to a “a wireless telephone” that “may turn off its display” in a low-power mode). *Araujo* discloses power management policies to be employed in a power management framework that are based on a power state of the device. *Araujo*, [0004], [0014].

164. In my opinion, a POSITA, when reviewing *Araujo*, would read *Araujo*’s disclosure in the context of the prior art that formed the background knowledge of an ordinary artisan. This would include *Cole*, which teaches that a “power state”—the same phrase used in *Araujo*—applies to the mobile device. Like *Araujo*, *Cole* teaches “[changing] the configuration of the communication functions in direct response to the power state of the mobile device 110.” *Cole*,

[0090]. As a result, a POSITA would have understood that in *Rao/Scahill*'s architecture, a power control state is a power state of the device.

165. According to *Araujo*, “devices may have various power states,” including “[o]n” and “off,” as well as “power states other than on and off.” *Araujo*, [0004]. *Araujo* explains that “a power management scheme may run a device in a lower-power-consumption mode.” *Araujo*, [0014]. As an example of a low-power mode, “a wireless telephone may turn off its display after some period of inactivity.” *Araujo*, [0014]. *Rao/Scahill*'s architecture, as modified by *Araujo*, applies network policies based on a device's power state (e.g., “a lower-power-consumption mode”) and based on whether an application's status is worthy of consuming power by accessing the Internet. *Araujo*, Abstract, [0004], [0014], [0027].

166. *Araujo* discloses that in one example of consuming power is executing a network transmission by using a network card (i.e., modem). *Araujo*, [0021]. *Araujo* discloses that, “[f]or the purpose of power management, a program [that is, application] may have a status that reflects its worthiness to consume power [by accessing the Internet]. For example, some programs may have a ‘VIP’ status that allows them to consume power in situations where other programs would not be permitted to consume power.” *Araujo*, [0027]. According to *Araujo*, “[p]rograms could be assigned binary VIP/non-VIP status, or their status could be indicated on

a scale.” *Araujo*, [0027]. Ultimately, “[w]hether a program is permitted to consume power [by accessing the Internet] could be based on a finding as to whether the program’s status justifies the consumption of power under the circumstances that are present.” *Araujo*, [0027].

167. As a result, in *Rao/Scahill*’s architecture, when a mobile device’s battery is low and a power conservation policy is in effect, a non-VIP application may be subject to a network access policy that restricts (or blocks) that application’s ability to consume power by utilizing a network connection. *Araujo*, [0027]. But when a power conservation policy is not in effect, that same non-VIP application may be subject to a different policy that does not restrict power consumption, including network access. *Araujo*, [0027]. As the device’s power state changes, *Rao/Scahill/Araujo*’s remote access client dynamically changes the policy associated with the non-VIP application. *Araujo*, [0025]; *Cole*, [0090].

4. Claim 20: “The wireless end-user device of claim 1, wherein the launch manager schedules corresponding execution times for applications to correspond with power state changes of the at least one wireless modem.”

168. In my opinion, the combination of *Rao*, *Scahill*, and *Araujo* renders obvious claim 20. As discussed in Section VII.D.2 (Motivation), a POSITA would have been motivated to modify *Rao/Scahill*’s architecture to include *Araujo*’s power management policy. When *Rao/Scahill/Araujo*’s device operates in “batch”

mode, this policy schedules task executions based on whether a wireless modem is powered on or off.

169. A POSITA would have understood that wireless devices, including *Rao*'s device, have the ability to operate "in a 'batch' mode that conserves power by allowing network transmissions to happen in bursts rather than at the time the time that the outbound network traffic is generated." (*Araujo*, [0043]; *see also Haughn*, [0049].) A POSITA would have further understood that this mode, which was a common feature of devices by the '700 patent's priority date, "facilitate a more efficient utilization of radio resources" by controlling when large data transmissions occur. (*Haughn*, [0049].)

170. *Araujo* discloses that "[t]he power management policy for such a mode might call for the network card [(i.e., modem)] to be turned on at specific intervals (e.g., every fifteen minutes)." *Araujo*, [0043]. When the modem is powered on and powered off, the modem undergoes a "power state change[]." According to *Araujo*, "[i]f a request to transmit data over a network is made between these intervals, the request may be delayed until the next time that the network card is scheduled to be powered on, at which time the request could be allowed." *Araujo*, [0043].

171. In my opinion, based on this disclosure, a POSITA would have been motivated and found it obvious to augment *Rao/Scahill*'s policies in view of

Araujo to include policies that schedule task executions based on whether a wireless modem is powered on or off when *Rao/Scahill*'s device operates in "batch" mode. The result is that *Rao/Scahill* filter 322 ("launch manager") applies *Araujo*'s power management policy to schedule execution times for applications to correspond with modem power state changes. Such a modification would be well within a POSITA's abilities (e.g., adding additional policies to *Rao*'s remote access client), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification.

VIII. DECLARATION

172. For the reasons set forth above, it is my opinion that claims 1-20 of the '700 patent would have been obvious in view of the prior art.

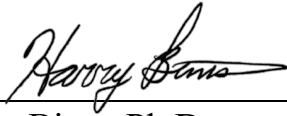
173. In signing this Declaration, I understand it will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I understand I may be subject to cross-examination in this case, and that any cross-examination will take place in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

174. I declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with knowledge that willful false

statements and the like so made are punishable by fine or imprisonment, or both,
under 18 U.S.C. § 1001.

175. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on this 8th day of August 2024.

By: 
Harry Bims, Ph.D.