

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Singh et al.
U.S. Patent No.: 8,862,757 Attorney Docket No.: 50048-0058IP1
Issue Date: October 14, 2014
Appl. Serial No.: 14/273,546
Filing Date: May 9, 2014
Title: AUTOMATIC MULTIMEDIA UPLOAD FOR PUBLISHING
DATA AND MULTIMEDIA CONTENT

DECLARATION OF DR. PATRICK TRAYNOR

I, Patrick Gerard Traynor, of Gainesville, Florida, declare that:

I. QUALIFICATIONS AND BACKGROUND INFORMATION

1. My name is Patrick Gerard Traynor, and I have been retained by TikTok Inc. to provide my opinions and conclusions as an expert on unpatentability grounds in this *Inter Partes* review (IPR) concerning U.S. Patent No. 8,862,757. My qualifications for forming these conclusions are summarized below.

2. I earned a B.S. in Computer Science from the University of Richmond in 2002 and an M.S. and Ph.D. in Computer Science and Engineering from the Pennsylvania State University in 2004 and 2008, respectively. My dissertation, entitled “Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks,” focused on security problems that arise in cellular belowstructure when gateways to the broader Internet were created.

3. I am currently a Professor in the Department of Computer and Information Science and Engineering (CISE) at the University of Florida. I was hired

under the “Rise to Preeminence” Hiring Campaign and serve as the Associate Chair for Research in my Department. I also hold the endowed position of the John and Mary Lou Dasburg Preeminent Chair in Engineering.

4. Prior to joining the University of Florida, I was an Associate Professor from March to August 2014 and an Assistant Professor of Computer Science from 2008 to March 2014 at the Georgia Institute of Technology. I have supervised many Ph.D., M.S., and undergraduate students during the course of my career.

5. My area of expertise is security, especially as it applies to mobile systems and networks, including cellular networks. As such, I regularly teach students taking my courses and participating in my research group to program and evaluate software and architectures for mobile and cellular systems. I have taught courses on the topics of network and systems security, cellular networks, and mobile systems at both Georgia Tech and the University of Florida. I also advised and instructed the Information Assurance Officer Training Program for the United States Army Signal Corps in the Spring of 2010.

6. I have received numerous awards for research and teaching, including being named a Kavli Fellow (2017), a Fellow of the Center for Financial Inclusion (2016), and a Research Fellow of the Alfred P. Sloan Foundation (2014). I also won the Lockheed Inspirational Young Faculty Award (2012), was awarded a National Science Foundation (NSF) CAREER Award (2010), and received the Center for

Enhancement of Teaching and Learning at Georgia Tech's "Thanks for Being a Great Teacher" Award (2009, 2012, 2013).

7. I have published over 100 articles in top conferences and journals in the areas of information security, mobile systems, and networking. Many of my results are highly cited, and I have received multiple "Best Paper" Awards. I have also written a book entitled "Security for Telecommunications Networks", which is used in wireless and cellular security courses at a number of top universities.

8. I am a Senior Member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). I am also a member of the USENIX Advanced Computing Systems Association.

9. I serve as an Associate Editor for IEEE Security and Privacy Magazine, have been the Program Chair for eight conferences and workshops, and have served as a member of the Program Committee for over 50 different conferences and workshops. I am also currently the Security Subcommittee Chair for the ACM US Technology Policy Committee (USACM).

10. I was a co-Founder and Research Fellow for a private start-up, Pindrop Security, from 2012 to 2014. Pindrop provides anti-fraud and authentication solutions for Caller-ID spoofing attacks in enterprise call centers by creating and matching acoustic fingerprints. Pindrop Security currently employs over 200 people, and their technology is based off of my research (US Patent 9,037,113 B2).

11. I was a co-Founder and Chief Executive of a private start-up, CryptoDrop. CryptoDrop developed a ransomware detection and recovery tool to provide state of the art protection to home, small business, and enterprise users. This technology was also based off of my research (US Patent 10,685,114 B2).

12. I was also a co-Founder and Chief Executive of a private start-up, Skim Reaper. Skim Reaper developed tools to detect credit card skimming devices, and worked with a range of banks, international law enforcement, regulators, and retailers. This technology was also based off of my research (US Patent 10,496,914 B2).

13. I am a named inventor on ten US patents. These patents detail methods for determining the origin and path taken by phone calls as they traverse various networks, cryptographically authenticating phone calls, providing a secure means of indoor localization using mobile/wireless devices, detecting credit card skimmers, identifying cloned credit cards, and blocking ransomware from encrypting data.

14. My curriculum vitae, included with this declaration, includes a list of publications on which I am a named author. It contains further details regarding my experience, education, publications, and other qualifications to render an expert opinion in connection with this proceeding.

15. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in mobile systems and

networks; my experience in teaching those subjects; and my experience in working with others involved in those fields. In addition, I have analyzed the following publications and materials, in addition to other materials I cite in my declaration:

- U.S. Patent No. 8,862,757 to Singh et al. (“the ’757 patent”) (TTI-1001)
- Excerpts from the Prosecution History of the ’757 patent (TTI-1002)
- U.S. Patent Application Publication 2005/0209927 to Aaltonen et al. (“Aaltonen”) (TTI-1005)
- U.S. Patent Application Publication 2006/0129631 to Na et al. (“Na”) (TTI-1006)
- *HTTP: The Definitive Guide* (“Gourley”) (TTI-1007)
- U.S. Patent Application Publication 2013/0315235 to Foo (“Foo”) (TTI-1009)
- International Publication No. 2008/028508 to Drescher et al. (“Drescher”) (TTI-1010)
- U.S. Patent Application Publication 2007/0174393 to Bosschaert et al. (“Bosschaert”) (TTI-1011)
- Internet Archive of Mobic.com, “Ericsson Unveils the First Bluetooth Phone,” available at <https://web.archive.org/web/20001002110758/http://www.mobic.com:80/>, last accessed February 29, 2024 (TTI-1012).

- U.S. Patent No. 8,041,816 to Ozaki et al. (“Ozaki”) (TTI-1015)
- Japanese Patent Application Publication No. 2005-303511 to Takahashi Susumu and Certified Translation (“Takahashi”) (TTI-1016)
- International Publication No. 2005/109781 to Lind et al. (“Lind”) (TTI-1017)
- U.S. Patent Application Publication No. 2008/0165982 to Hankey et al. (“Hankey”) (TTI-1018)
- Hypertext Transfer Protocol -- HTTP/1.1, June 1999 (“RFC2616”) (TTI-1019)
- U.S. Patent No. 7,181,207 to Chow et al. (“Chow”) (TTI-1020)
- U.S. Patent No. 7,333,955 to Graves et al. (“Graves”) (TTI-1021)
- U.S. Patent No. 8,095,463 to Hartmaier (“Hartmaier”) (TTI-1022)
- WAP MMS Architecture Overview Version 25-April-2001 (“MMS_1.0”) (TTI-1023)
- U.S. Patent Application Publication No. 2009/0007196 to Ganesan (“Ganesan”) (TTI-1024)

II. LEGAL PRINCIPLES

A. Anticipation

16. I have been informed that a patent claim is invalid as anticipated under 35 U.S.C. § 102 if each and every element of a claim, as properly construed, is found

either explicitly or inherently in a single prior art reference. Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes the claimed limitations, it anticipates.

17. I have been informed that a claim is invalid under 35 U.S.C. § 102(a) if the claimed invention was known or used by others in the U.S., or was patented or published anywhere, before the applicant's invention. I further have been informed that a claim is invalid under 35 U.S.C. § 102(b) if the invention was patented or published anywhere, or was in public use, on sale, or offered for sale in this country, more than one year prior to the filing date of the patent application (critical date). And a claim is invalid, as I have been informed, under 35 U.S.C. § 102(e), if an invention described by that claim was described in a U.S. patent granted on an application for a patent by another that was filed in the U.S. before the date of invention for such a claim.

B. Obviousness

18. I have been informed that a patent claim is invalid as "obvious" under 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to a POSITA, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called "secondary considerations" of non-obviousness, which include: (i) "long felt need" for the claimed invention, (ii) commercial success

attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) “copying” of the claimed invention by others. For purposes of my analysis above and because I know of no indication from the patent owner or others to the contrary, I have applied a date of December 28, 2007, as the date of invention in my obviousness analyses, although in many cases the same analysis would hold true even at an earlier time than December 28, 2007.

19. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge or “common sense” of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

III. OVERVIEW OF CONCLUSIONS FORMED

20. This expert Declaration explains the conclusions that I have formed based on my analysis. To summarize those conclusions:

- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-26 of the '757 patent are obvious over Aaltonen in view of Drescher.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-26 of the '757 patent are obvious over Takahashi in view of Na.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-26 of the '757 patent are obvious over Takahashi in view of Na and Lind.

IV. BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE '757 PATENT

21. Based on the foregoing and upon my experience in this area, a person of ordinary skill in the art ("POSITA") relating to the subject matter of the '757 patent as of December 28, 2007 would have had a Bachelor of Science in electrical engineering, computer engineering, computer science, or a related field, with at least two years of relevant work experience in wireless communications and/or

networking systems, or equivalents thereof. Additional graduate education could substitute for professional experience, and *vice versa. Id.*

22. Based on my experiences, I have a good understanding of the capabilities of a POSITA as I was such an individual at the time of the Critical Date. Moreover, I have taught, participated in organizations, and worked closely with many such persons over the course of my career.

V. INTERPRETATIONS OF THE '757 PATENT CLAIMS AT ISSUE

23. I have been informed by Counsel and understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by one of ordinary skill. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office. Counsel has also informed me, and I understand that, the words of the claims should be interpreted as they would have been interpreted by one of ordinary skill at the time of the invention was made (not today). I have been informed by Counsel that I should use December 28, 2007 as the point in time for claim interpretation purposes.

VI. OVERVIEW OF THE '757 PATENT

A. The '757 Patent

24. The '757 patent relates to “a BluetoothTM (BT) enabled mobile device for publishing data and multimedia content on one or more websites automatically or with minimal user intervention.” TTI-1001, 2:3-5, 1:36-40, 3:6-34, 5:18-64. As

shown in FIGS. 2 and 4, a mobile device 202 includes a client application 203. TTI-1001, 3:41-42. “In the case of a mobile device 202 with limited memory and processing capabilities, the client application 203 partitions the multimedia content of large files stored on the mobile device 202 into multiple data segments using the segmentation module 203i,” and the “segmentation module 203i generates segment identifiers and tags the data segments with the segment identifiers.” TTI-1001, 7:23-29, 7:52-57, 5:4-9, 8:62-9:3. “The segment identifiers may, for example, be one or more of transaction identifiers, sequence numbers, and timestamps. The segment identifiers are used later by a back end service 401b of the publishing service 401 to reassemble the data segments in a predetermined sequence to create a multimedia object.” TTI-1001, 7:57-62. “The tagged data segments are transferred from the client application 203 of the mobile device 202 to a publishing service 401 via a network 402 as illustrated in FIG. 4.” TTI-1001, 5:9-11, 8:65-9:3. “The data transfer module 203i transfers the data, the tagged data segments, and the multimedia content from the client application 203 to the publishing service 401 via a network 402 for publishing on the websites automatically.” TTI-1001, 7:29-33, 7:62-67.

25. As shown in FIG. 4, the “publishing service 401” includes “a front end service 401a, a back end service 401b, and a database 401d.” TTI-1001, 8:1-2. According to the ’757 patent, a “protocol is provided for synchronizing user publishing information between the client application 203 and the publishing service

401.” TTI-1001, 8:4-6. “The back end service 401b” includes “a data reassembly module 401c.” TTI-1001, 8:13-14. “If the back end service 401b receives the multimedia content in the form of multiple data segments, the data reassembly module 401c reassembles the data segments in a predetermined sequence using the segment identifiers. The back end service 401b then creates a multimedia object from the transferred data and multimedia content. The multimedia object is transferred from the back end service 401b to the front end service 401a and then published on the websites selected by the user.” TTI-1001, 8:14-22, 9:3-11.

B. Prosecution History

26. The '757 patent issued from U.S. Application No. 14/273,546, which was filed on May 9, 2014 and claims priority through continuation applications to Provisional Application No. 61/017,202 filed on December 28, 2007. TTI-1001, (21)-(22), (63), (60). In the sole office action, claims 1-30 were rejected based on nonstatutory obviousness-type double patenting over claims 1-5 of U.S Application No. 14/109,972. TTI-1002, 69-76. The examiner also rejected dependent claims 6-8, 14-16, 21-23, and 28-30 under §112 for reciting new subject matter. *Id.* The applicant filed a terminal disclaimer and amended the claims to obviate the §112 rejection. TTI-1002, 98-122, 189, 193. All claims were then allowed with an examiner's amendment of independent claims 1, 9, 17, and 24 to include “wherein transferring the data segment further comprises one or more of: configuring the data

segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service.” TTI-1002, 201-221.

VII. ANALYSIS OF AALTONEN AND DRESCHER

A. Overview of Aaltonen

27. Aaltonen discloses techniques for “uploading content” with “greater control over the uploading of content by allowing ... entities to define instructions to which upload of the media content is subject.” TTI-1005, Abstract, ¶¶[0008]-[0009]. For example, Aaltonen discloses “recipient,” “sender,” and “network instructions” that “define how content is uploaded to a recipient.” TTI-1005, ¶¶[0063]-[0082]. An example system is illustrated in Aaltonen’s FIG. 6¹.

¹ Throughout its disclosure, Aaltonen refers to the “sender 102” and “content manager 106” interchangeably when discussing the uploading of content. TTI-1005, ¶¶[0060]-[0062], [0114]-[0115]. The same is true for the “recipient 104” and “upload agent 110” when discussing receiving upload content. *Id.*

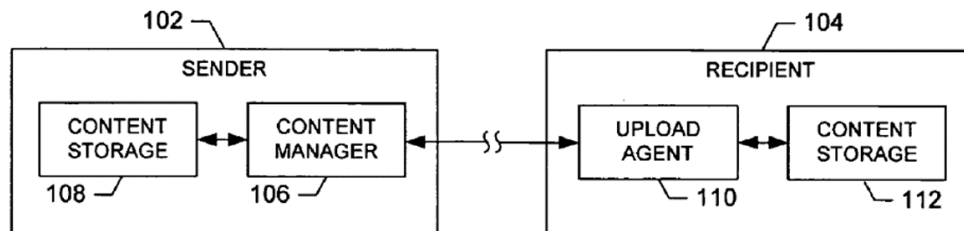


FIG. 6.

TTI-1005, FIG. 6.

28. Aaltonen discloses that the content manager can “break up the upload content into a plurality of portions” in response to instructions. TTI-1005, ¶[0012], [0069]. Aaltonen’s uploads can be performed with Hypertext Transfer Protocol (“HTTP”) techniques that divide uploaded content into “a number of content data packets.” TTI-1005, ¶¶[0101]-[0106]. These HTTP packets include “information ... uniquely describing the packets” to include “one or more identifiers identifying the sender 102, the information packet, requested content and/or the upload session.” TTI-1005, ¶[0102]. Aaltonen discloses two specific HTTP techniques of uploading content – “HTTP POST” and “HTTP PUT.” TTI-1005, ¶¶[0016], [0101], [0111]-[0113], FIGS. 9, 11.

29. Aaltonen also discloses techniques for recovering from an “interruption” during an upload. TTI-1005, ¶¶[0106]-[0114], FIGS. 9, 11. For example, Aaltonen’s FIG. 11 description discloses a process where, after an interruption, the sender “request[s] the bit-length or size of the portion of the upload content 114

received by the upload agent 110” and then “continue[s] uploading the content to the recipient 104” based on the response. TTI-1005, ¶¶[0111]-[0113], FIG. 11.

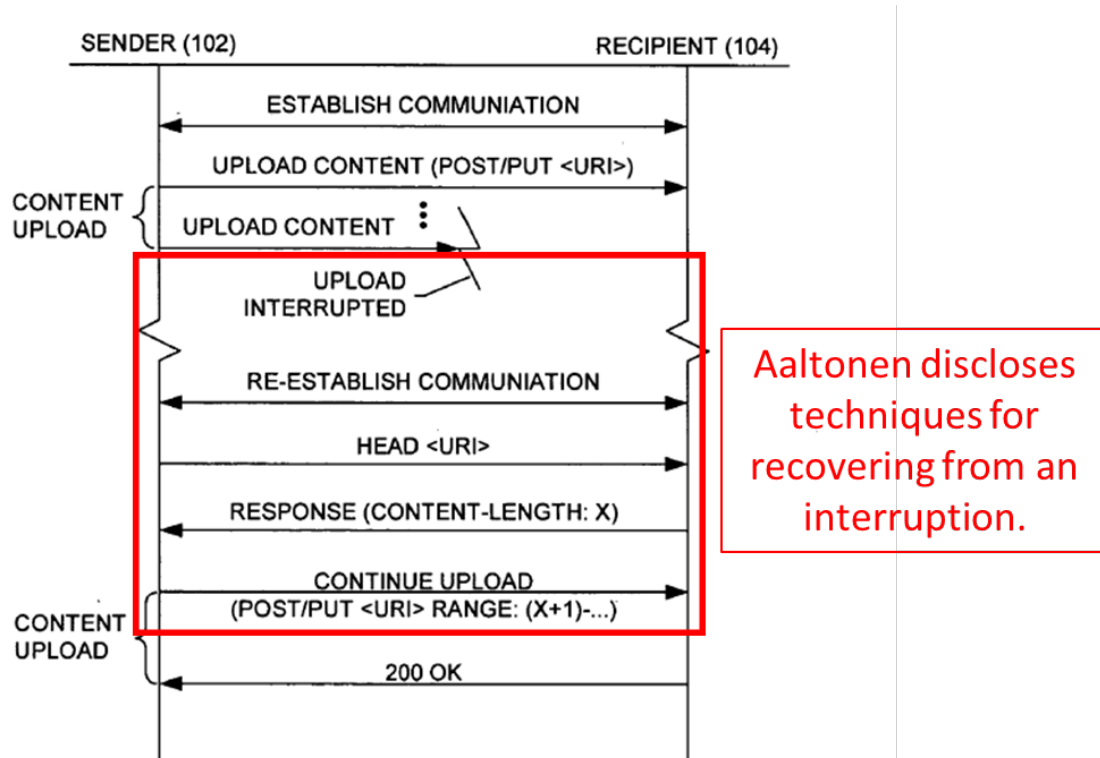


FIG. 11.

TTI-1005, FIG. 11².

B. Overview of Drescher

30. Drescher discloses techniques for “sending large files or resources over the internet” where “chunks of the file are sent over parallel HTTP connections.”

² Annotations and color added to figures unless otherwise noted.

TTI-1010, Cover, 4:15-30³, 5:1-11, 10:14-11:7. Additionally, Drescher discloses a “HTTP PUT” technique (like Aaltonen) for uploading “chunks” of content where the server sends a “HTTP Response” to the sender’s PUT request that includes “the server’s log of progress of processing the PUT Request.” TTI-1010, 11:27-12:10, 15:23-16:16. Drescher describes its techniques as “ParHTTP,” a similar protocol to standard HTTP. *Id.*, 9:16-21, 10:1-12, 12:12-20, 25:32-26:21.

31. Drescher also discloses that ParHTTP techniques include a method of “communicating identities between the client 200 and the server 100.” TTI-1010, 25:32-26:21, 34:13-35:4, 35:26-36:21. More specially, “the identity provider 220 adds to each HTTP request 802 sent to the server (i.e. for each chunk, and for PUT and GET) a header for each identity to be transferred,” the identity including “a name particularising the ... [party] involved in the request.” *Id.*

C. A POSITA would have combined Aaltonen and Drescher

32. It would have been obvious to a POSITA to combine the teachings of Aaltonen and Drescher such that (1) Drescher’s HTTP responses would have been used to acknowledge data blocks that are successfully received by the recipient via Aaltonen’s HTTP PUT/POST techniques and (2) Drescher’s party-identifying information would have been added to Aaltonen’s HTTP PUT and POST messages.

³ The citations to Drescher refer to the page number of the PDF document.

TTI-1010, 14:10-15, 25:32-26:21, 34:13-35:4, 35:26-36:21; TTI-1005, ¶¶[0012], [0016], [0028], [0069], [0093]-[0094], [0096], [0101]-[0102], [0111], [0113], FIG. 10; *see above* §VII.A, §VII.B. A POSITA would have combined Aaltonen and Drescher for at least the reasons below.

33. First, a POSITA would have recognized that, for segmented uploads, sending an acknowledgement (e.g., Drescher's HTTP response) for each block and/or segment would have indicated to the sender's content manager 106 that the successfully received data could be deleted from the sender's 102 content storage 108. TTI-1005, ¶¶[0093]-[0094], [0096], [0101]; TTI-1010, 14:10-15. A POSITA would have known from Aaltonen's disclosure that acknowledgements indicate to the content manager 106 that it can "delete or otherwise remove the upload content from the content storage 108, thereby relieving at least a portion of the storage capacity burden on the sender, such as for subsequent storage of other content." TTI-1005, ¶¶[0093]-[0094], [0096], [0101].

34. Second, Drescher's HTTP responses would have acted as another means for the sender's content manager 106 to expediently detect service interruptions (e.g., a HTTP response indicating a failure would alert the sender that the recipient did not receive the most recently transmitted block and/or segment). TTI-1005, ¶¶[0111]-[0114], FIG. 11; TTI-1010, 14:10-15, 15:23-16:16.

35. Third, a POSITA would have recognized that adding party-identifying information to data transfer messages would improve the security of Aaltonen's data transfer techniques, as the recipient would be able to identify when a message has been received from a source that does not have "permission" to upload data to the recipient. TTI-1010, 25:4-26:21, 34:13-35:4, 35:26-36:21.

36. Combining Aaltonen and Drescher would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results; and (2) use of known techniques to improve similar devices in the same way.

37. A POSITA would have had a reasonable expectation of success in implementing the Aaltonen-Drescher combination. Aaltonen already describes acknowledging data transfers in a segmented data transfer protocol, and the combination of Drescher's teachings would have simply involved adding Drescher's known HTTP responses to Aaltonen's similar HTTP techniques. TTI-1005, ¶¶[0093]-[0094], [0096], [0101]; TTI-1010, 14:10-15. For at least this reason, the Aaltonen-Drescher combination is predictable and foreseeable and would have been implemented with a reasonable expectation of success. *Id.* Additionally, Aaltonen already discloses adding identifying information to HTTP messages – like Drescher. TTI-1010, 25:4-26:21, 34:13-35:4, 35:26-36:21; TTI-1005, ¶¶[0102], [0104]. Moreover, both Aaltonen and Drescher disclose segmented data transfer protocols

implemented in the HTTP protocol, which would have indicated to a POSITA that the techniques of Aaltonen and Drescher would have been compatible, without requiring undue experimentation. TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1010, Cover, 10:14-15:9.

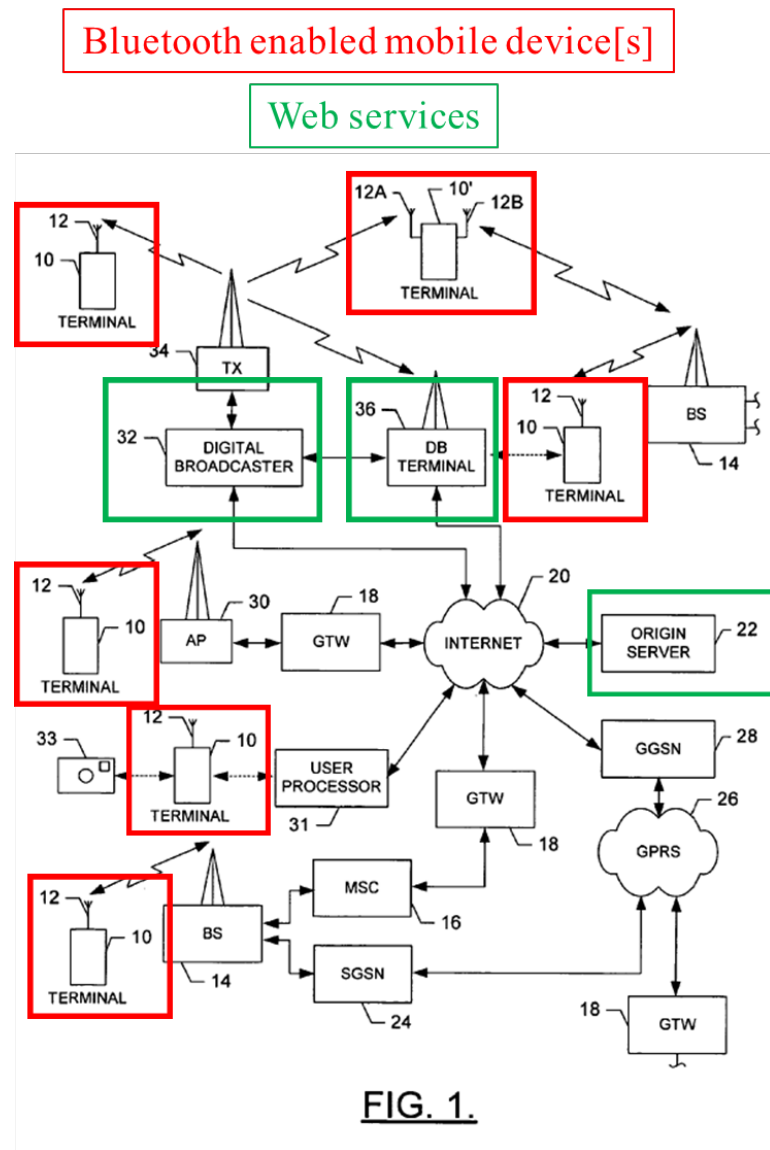
38. In the combined Aaltonen-Drescher system, Drescher's HTTP responses and identifying information would have been used to supplement Aaltonen's existing acknowledgements and identifiers, such that (1) data blocks successfully received by the recipient's upload agent 110 would have been indicated to the sender's content manager 106 with an HTTP response indicating the successful upload and (2) data blocks would identify a particular user of the system (a "party"). TTI-1010, 14:10-15, 25:32-26:21; TTI-1005, ¶¶[0012], [0028], [0069], [0093]-[0094], [0096], [0101]-[0102], FIG. 10.

D. Claims 1-26 are rendered obvious by Aaltonen in view of Drescher

1[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:

39. To the extent the preamble is limiting, the Aaltonen-Drescher combination renders obvious element 1[pre]. For example, Aaltonen discloses a method of "uploading content" ("***file transfer***") from a "mobile station" and/or "terminal 10" ("***mobile device***") to a "recipient" that can be an "origin server 22," "digital broadcaster 32," and/or "digital broadcast receiver 36" ("***a web service***").

TTI-1005, ¶¶[0002]-[0007], [0008], [0038]-[0046], [0056], [0097], [0111]-[0115], FIGS. 1, 5, 9-11. Additionally, Aaltonen discloses that its “content manager” can “break up ... upload content into a plurality of portions” (“*segmented file transfer*”). TTI-1005, ¶[0069]; *see below* 1[c]. Reproduced below is an example network in Aaltonen.



TTI-1005, FIG. 1.

40. Aaltonen discloses that “mobile terminals” (“**mobile device[s]**”) include “mobile telephones,” “digital cameras,” “personal digital assistants [PDAs],” “pagers,” and “laptop computers.” TTI-1005, ¶¶[0003], [0052]. Like Aaltonen, the ’757 patent similarly describes that “**mobile device[s]**” include “mobile phone[s],” “PDAs,” “laptop computers,” and “digital cameras.” TTI-1001, 9:37-60.

41. Additionally, Aaltonen discloses that mobile terminals include a “Bluetooth (BT) transceiver” and can “share and/or obtain data in accordance with Bluetooth transfer techniques” (“**a Bluetooth enabled mobile device**”). TTI-1005, ¶[0056], *see also id.* ¶¶[0036]-[0037], [0046]. Additionally, it was well known before the Critical Date that the Bluetooth protocol was compatible with mobile devices. TTI-1009, ¶¶[0016], [0031], [0042]-[0043], [0049], [0054]; TTI-1011, ¶[0034], Claim 6; TTI-1012.

42. The ’757 patent does not define a “**web service**,” but describes “publishing data and multimedia content on one or more websites” (an example of a “**web service**”). TTI-1001, 3:30-40, *see also id.* 5:65-6:8. Similarly, Aaltonen discloses that a recipient of an upload can be a “origin server 22,” “digital broadcaster 32,” “digital broadcast receiver[s] 36” and that uploads occur over the “Internet” (a network hosting “**web service[s]**”). TTI-1005, ¶¶[0005], [0033], [0038], [0041]-[0042], [0056]-[0057]. Additionally, Aaltonen discloses a “digital broadcaster 32” (“**one or more web services**”) coupled to “terminal[s] 10” over a

“digital broadcast network” (e.g., the “Internet 20”). TTI-1005, ¶¶[0039]-[0042]. Aaltonen’s “digital broadcast receiver[s] 36” (“*one or more web services*”) also includes “one or more interface means for sharing and/or obtaining data from electronic devices, such as terminals 10,” particularly “a network interface means 60, for sharing and/or obtaining data from a network, such as the Internet 20.” TTI-1005, ¶¶[0041]-[0046]. Aaltonen’s mobile station can use “a conventional Web browser for communicating in accordance with ... HTTP.” TTI-1005, ¶[0057]. A POSITA would have recognized and found obvious that the above services would have hosted “websites,” and accordingly, the recipient in the Aaltonen-Drescher combination would have been a “*web service*” at least for this reason. *Id.*

1[a]. providing a software module on the Bluetooth enabled mobile device, the software module, performing;

43. The Aaltonen-Drescher combination renders obvious element 1[a]. For example, Aaltonen discloses a “content manager 106” application (“*a software module on the Bluetooth enabled mobile device*”) that is “capable of uploading one or more piece(s) of content from a content storage 108 in memory ... of the sender, to the recipient.” TTI-1005, ¶¶[0060]-[0062] (disclosing that the content manager 106 is “software operated by the sender 102” (“*a software module*”)), [0115], FIG. 6. Aaltonen’s content manger 106 performs various functions, for example, formatting and uploading of content, deleting successfully uploaded content from the sender’s memory, and recovering from upload interruptions. TTI-1005,

¶¶[0060]-[0062], [0066], [0069], [0073], [0077]-[0080], [0084]-[0086], [0089]-[0094], [0111]-[0115].

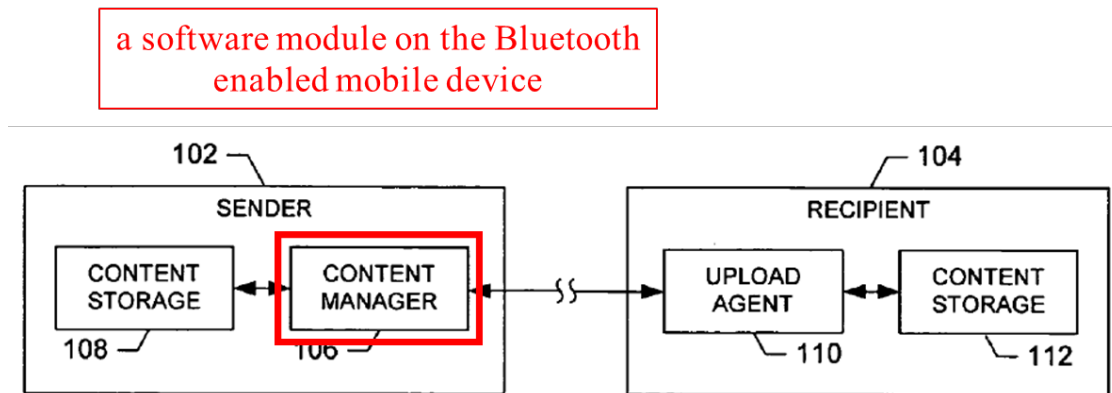


FIG. 6.

TTI-1005, FIG. 6.

1[b]. determining size of a file to be transferred;

44. The Aaltonen-Drescher combination renders obvious element 1[a] in multiple ways.

45. First, Aaltonen discloses “instructions” for uploading content “such that the upload content is transcoded, truncated or otherwise processed based upon the type, size and/or quality of the content” (“***size of a file to be transferred***”). TTI-1005, ¶¶[0012]-[0013], [0059], [0069], [0078]. The sender’s (“mobile terminal”) content manager of the Aaltonen-Drescher combination would have “***determin[ed]***” the size of the file to be uploaded when specifying instructions based on content size.

46. Second, Aaltonen discloses that a sender generates an “upload request,” also referred to as an “upload descriptor,” that includes “information necessary to

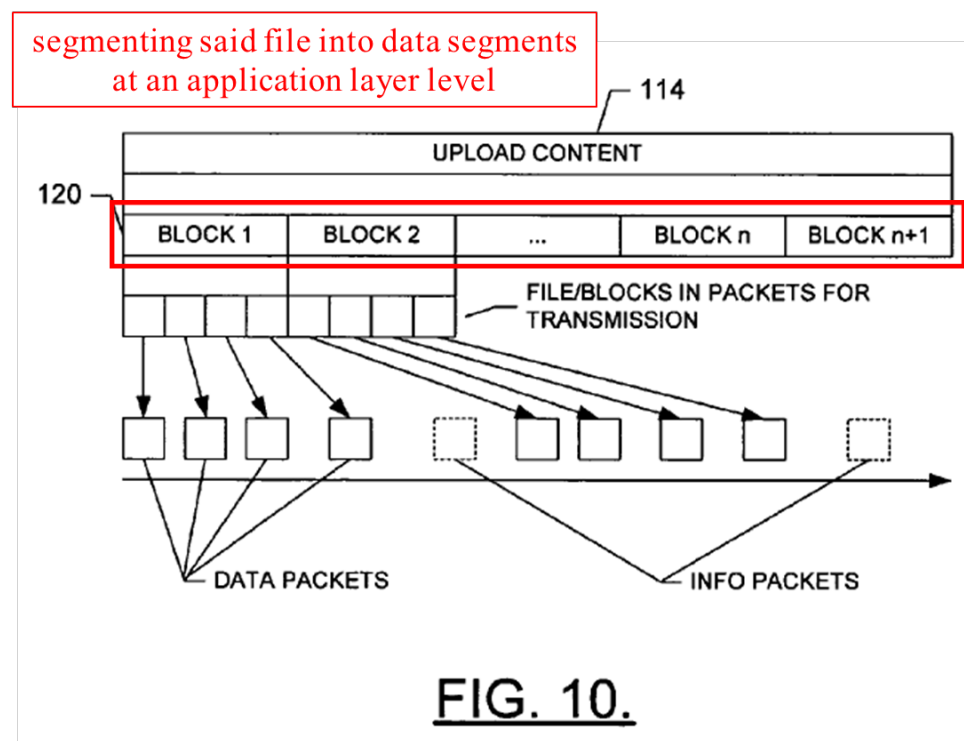
describe the content (e.g., size, quality, etc.)” (“*size of a file to be transferred*”). TTI-1005, ¶¶[0096], [0100]. The sender’s content manager of the Aaltonen-Drescher combination would have “*determin[ed]*” the size of the file to be uploaded when providing the “information” describing the content size in the upload descriptor.

47. Third, Aaltonen discloses the content manager of the sender “selecting or receiving a selection of one or more pieces of content to be uploaded to the recipient” and “select[ing] piece(s) of content based upon the type, size and/or quality of the respective piece(s) of content” (“*size of a file to be transferred*”). TTI-1005, ¶¶[0084]. The content manager of the Aaltonen-Drescher combination would have “*determin[ed]*” the size of the file to be uploaded when selecting uploads based on content size as this information is used to determine what content is uploaded from the sender’s content storage.

1[c1]. segmenting said file into data segments at an application layer level;

48. The Aaltonen-Drescher combination renders obvious element 1[c1]. Aaltonen discloses “breaking ... content into a plurality of portions” in response to “instructions” (“*segmenting said file into data segments*”). TTI-1005, ¶¶[0012], [0069]; *see above* 1[pre]. Aaltonen’s FIG. 10 illustrates an example of an upload with several content “groups or blocks” (“*data segments*”). TTI-1005, ¶¶[0028], [0102], FIG. 10. Additionally, because the Aaltonen-Drescher combination uses

HTTP, a widely known application layer protocol, a POSITA would have understood and found obvious that the segmentation of a file is “*at an application layer level.*” TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]. For example, Gourley corroborates this understanding and discloses that HTTP is part of the “application layer” (or “web application layer”). TTI-1007, 76, 249-251. Moreover, during the prosecution history of the ’757 patent, the applicant acknowledged that “[i]t is well known that both HTTP and XML-RPC are application layer protocols.” TTI-1002, 115.



TTI-1005, FIG. 10.

49. Additionally, segmenting a file for upload was well known in the art prior to the Critical Date. For example, Drescher discloses a “method of transferring

a resource over a network between a first station and a second station, the method comprising: dividing the resource into a plurality of chunks, [and] sending the chunks between the first and second station.” TTI-1010, 4:15-28, 9:31-12:24. As another example, Bosschaert discloses a “network stack” that includes a “chunking, or fragmentation, layer 204” that is “responsible for breaking a message into a series of smaller messages, sometimes called chunks or fragments, which can be transmitted separately and re-combined at the other side to reconstruct the message.” TTI-1011, ¶¶[0037]-[0039].

1[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

50. The Aaltonen-Drescher combination renders obvious element 1[c2]. For example, as discussed in 1[pre], Aaltonen discloses a method of segmented data transfer (“***said file is segmented***”). See above 1[pre]; see also below 1[f1].

51. Aaltonen discloses “instruction[s]” based on the “type, size and/or quality of the content to be uploaded” (“***size of said file***”) which “define processing the content, such as ... breaking the content into a plurality of portions” (“***said file is segmented in part due to size of said file***”). TTI-1005, ¶¶[0012], [0059], [0063]-[0069]. Additionally, segmenting data transfers due to a file size was well known in the art. Indeed, as one example, Foo discloses that a “buffer transfer session ... used to transfer multiple data objects sequentially” can be used for “data objects whose size exceeds a certain threshold.” TTI-1009, ¶¶[0036], [0135].

52. In the Aaltonen-Drescher combination, instructions – which specify “breaking ... content into a plurality of portions” – are also defined “based upon the connectivity of the recipient and/or respective senders” (“*said file is segmented in part due to ... data transfer capabilities of said Bluetooth enabled mobile device*”). TTI-1005, ¶¶[0011], [0066], [0071], [0091]-[0092]. Aaltonen discloses that instructions can include “state information” regarding “connectivity” at the sender or recipient, the “network over which the content is uploaded, such as traffic on the network(s) and/or bandwidth available to the recipient and/or the sender on the network(s),” and/or “actual movement and/or predicted movement of the recipient and/or sender” (“*data transfer capabilities of the Bluetooth enabled mobile device*”). TTI-1005, ¶¶[0011], [0066], [0080], [0091]-[0092]. Additionally, Aaltonen discloses sender and/or recipient specified “network instructions” that can be defined based on the “available bandwidth to the recipient 104 and/or the sender 102” – a network parameter that is directly correlated to the capability of data transfer (“*data transfer capabilities of the Bluetooth enabled mobile device*”). TTI-1005, ¶¶[0011]-[0012], [0075]-[0079].

1[c3]. wherein data in said data segments comprises one or more of said file data and data associated with said file data;

53. The Aaltonen-Drescher combination renders obvious element 1[c3]. For example, Aaltonen discloses that uploaded data in the data blocks (“*data segments*”) includes “graphic, audio and/or video content” and “text content” (“*file*”).

data”). TTI-1005, ¶¶[0049], [0068], [0070], [0072], [0084]. This is consistent with the ’757 patent, which describes “data and multimedia content may, for example, comprise image files, audio files, video files, text files, or any combination thereof” (“*file data*”). TTI-1001, 4:22-25, *see also* 2:48-54, 5:55-59.

54. The ’757 patent frequently refers to “files associated with the captured data and the multimedia content” (“*data associated with said file data*”). TTI-1001, 4:27-28, *see also* 2:34-36, 4:55-57, 5:12-17, 6:33-52, 7:18-22, 7:48-51, 8:37-42. However, the specification of the ’757 patent does not define or suggest examples of “*data associated with said file data*.” *See generally*, TTI-1001. Instead, the ’757 patent claims examples of “*data associated with file data*” that include “at least one of name of the file, size of the file, and type of file.” TTI-1001, Claim 2. For the purposes of my analysis, I have used the examples provided in the ’757 patent’s claims as a guide when analyzing this claim feature. *Id.* These are not the only examples of “*data associated with multimedia data*,” as this feature would have been understood by a POSITA.

55. Aaltonen discloses “HTTP header information,” which can be included in all data blocks and HTTP messages. TTI-1005, ¶¶[0112]-[0113]. A POSITA would have recognized or found obvious that “HTTP header information,” for example, in Aaltonen-Drescher’s HTTP PUT or POST messages, would have included both the “name of the file” and “type of the file” being uploaded (“*data*

associated with multimedia data”). TTI-1010, 32:25-28 (“content type specified in the ‘Content-Type’ header”). Indeed, Gourley discloses that HTTP PUT messages include both the “name of the file” and “type of the file.” TTI-1007, 54-55.

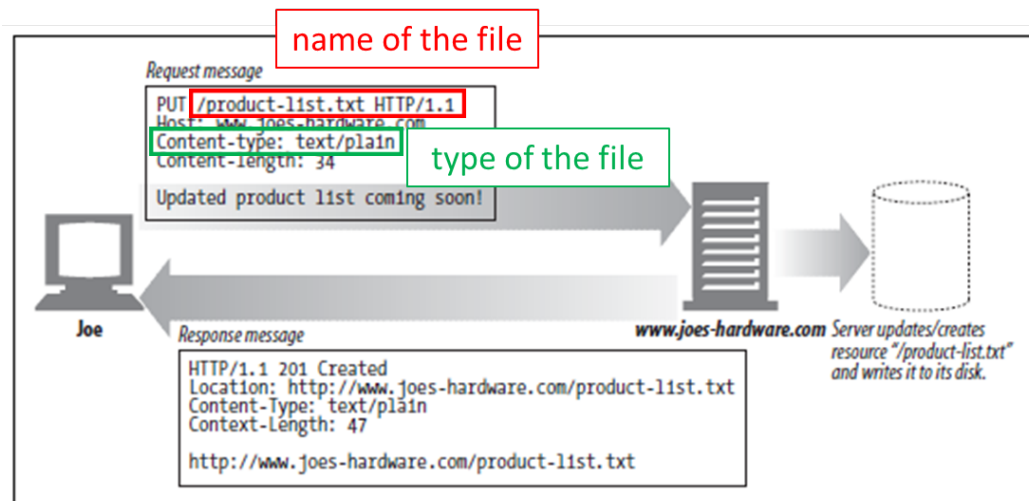


Figure 3-9. PUT example

TTI-1007, 55 (FIG. 3-9).

56. Additionally, Aaltonen discloses that an “instruction” can be defined “that orders or prioritizes piece(s) or object(s) within the upload content, such as ordering graphic, audio and/or video content of the upload content” (thus indicating the “type of file” and “*data associated with said file data*”). TTI-1005, ¶[0072]. Indeed, Aaltonen provides examples where the upload and/or storing of content at the recipient changes based on the file type. *Id.* (disclosing examples of “a sports video clip with audio commentary” where “the sender could order video at a higher priority” and “a song clip with an audio track and a video of the singer” where “the sender may order the audio a higher priority”). A POSITA would have recognized

or found obvious that the type of a file would have been indicated in the Aaltonen-Drescher HTTP message headers, as it was well known before the Critical Date that HTTP message headers contained this information. TTI-1010, 32:25-28; TTI-1007, 54-55.

1[d]. applying a segment identifier to each of said data segments at the application layer level;

57. The Aaltonen-Drescher combination renders obvious element 1[d]. For example, Aaltonen discloses that “information packets... added to the packet stream” “between groups or blocks 120 of one or more packets” or alternatively “data packets” within its HTTP data “blocks” (“***data segments***”) include “an identifier that identifies placement of the respective data packet in relation to the other data packets of the content,” “a number of packets in the block of packets,” “information ... uniquely describing the packets,” and “a table uniquely identifying the block of packets before or after the respective information [or data] packet” (all examples of “***applying a segment identifier to each of said data segments***”). TTI-1005, ¶¶[0101]-[0104] (disclosing that “information otherwise included in the information packets can alternatively be added to one or more of the data packets”), FIG. 10.

58. Additionally, a POSITA would have found it obvious that this information would have been included for or in “***each***” data block (“***each data segment***”), for example, to enable the upload agent to identify the proper placement of blocks in the upload content as they are received. *Id.* A POSITA would have

recognized and found obvious that identifying information would preferentially be added for or to “*each*” block as Aaltonen relies upon this information to “monitor the received ... blocks of packet(s) or portion of the upload content,” determine if an interruption has been experienced, and “facilitate recovery of the upload session” following an interruption. TTI-1005, ¶¶[0106]-[0107]. Neglecting to add identifying information to any one block would create the potential that a recipient would not be able to properly dispose of that data block.

59. Additionally, because Aaltonen-Drescher uses HTTP, a POSITA would have understood and found obvious that these identifiers would have been applied to data blocks “*at the application layer level.*” See above, 1[b]; TTI-1002, 115. Indeed, Gourley corroborates this understanding and discloses that HTTP is part of the “application layer” (or “web application layer”). TTI-1007, 76, 249-251.

1[e]. applying a user identifier to each of said data segments for the destination web service at the application layer level, wherein said user identifier uniquely identifies a particular user of the web service;

60. The Aaltonen-Drescher combination renders obvious element 1[e]. For example, Aaltonen discloses that “information packets... added to the packet stream” “between groups or blocks 120 of one or more packets” or alternatively “data packets” within its HTTP data “blocks” (“*data segments*”) include “identifiers identifying the sender 102.”. TTI-1005, ¶¶[0101]-[0102], [0104] (disclosing that “information otherwise included in the information packets can alternatively be added to one or

more of the data packets”). Moreover, Aaltonen uploads content “in a number of content data packets” “to the recipient in accordance with an HTTP POST or HTTP PUT technique,” and according to Drescher, HTTP PUT and POST messages would have had a user identifier disclosing “a name particularising the ... [party] involved in the request” (“*applying a user identifier*” that “*uniquely identifies a particular user of the web service*”). TTI-1005, [0101], [0111]; TTI-1010, 25:32-26:21, 34:13-35:4, 35:26-36:21. Drescher discloses that this information is preferentially added “for each chunk, and ... PUT” (each of the data packets in the Aaltonen-Drescher combination). *Id.* Also, at least because Drescher teaches that user identity is used to determine “permissions” to allow “access to resources at the server” (the recipient in the Aaltonen-Drescher combination), this information is “*for the destination web service.*” TTI-1010, 25:4-25, 34:13-35:4, 35:26-36:21. Because each data packet within a data block includes this information, “*each*” data block (“*each data segment*”) would have included this information. *See above* 1[c1]-1[d].

61. Additionally, a POSITA would have understood and found obvious that these identifiers would have been applied to packets “*at the application layer level.*” *See above* 1[d]. Further, the “upload” is directed to the “*web service,*” and the applied “*user identifier*” is for “*each destination web service,*” for example, because Aaltonen discloses that the “upload agent 110” (within the “recipient 104” (“*web service*”)) can “utilize the [identifiers within] information packets to monitor the

received data packets,” or more particularly, “track the number and accuracy of the received packets, blocks of packet(s) or portion of the upload content 114.” TTI-1005, ¶[0104].

1[f1]. transferring a data segment along with said user identifier and said segment identifier to the web service,

62. The Aaltonen-Drescher combination renders obvious element 1[f]. For example, Aaltonen discloses that, when the sender uploads content to a recipient, the content can be uploaded in “groups or blocks 120 of one or more packets” (*“transferring a data segment ... to the web service”*). TTI-1005, ¶¶[0101]-[0106], FIGS. 8-11. Aaltonen’s FIG. 10 depicts an example upload.

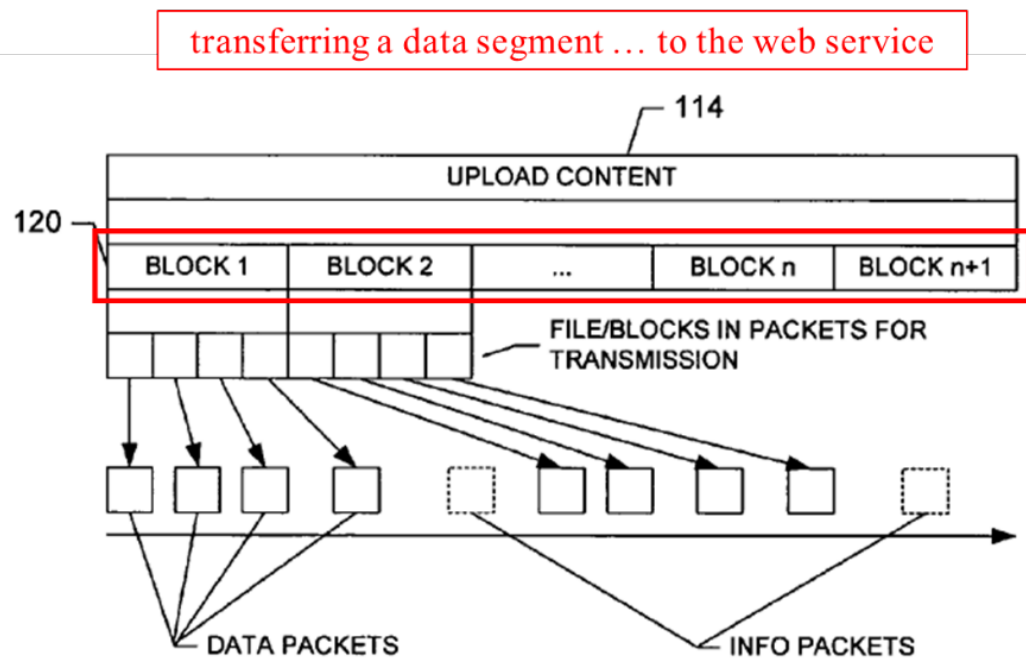


FIG. 10.

TTI-1005, FIG. 10.

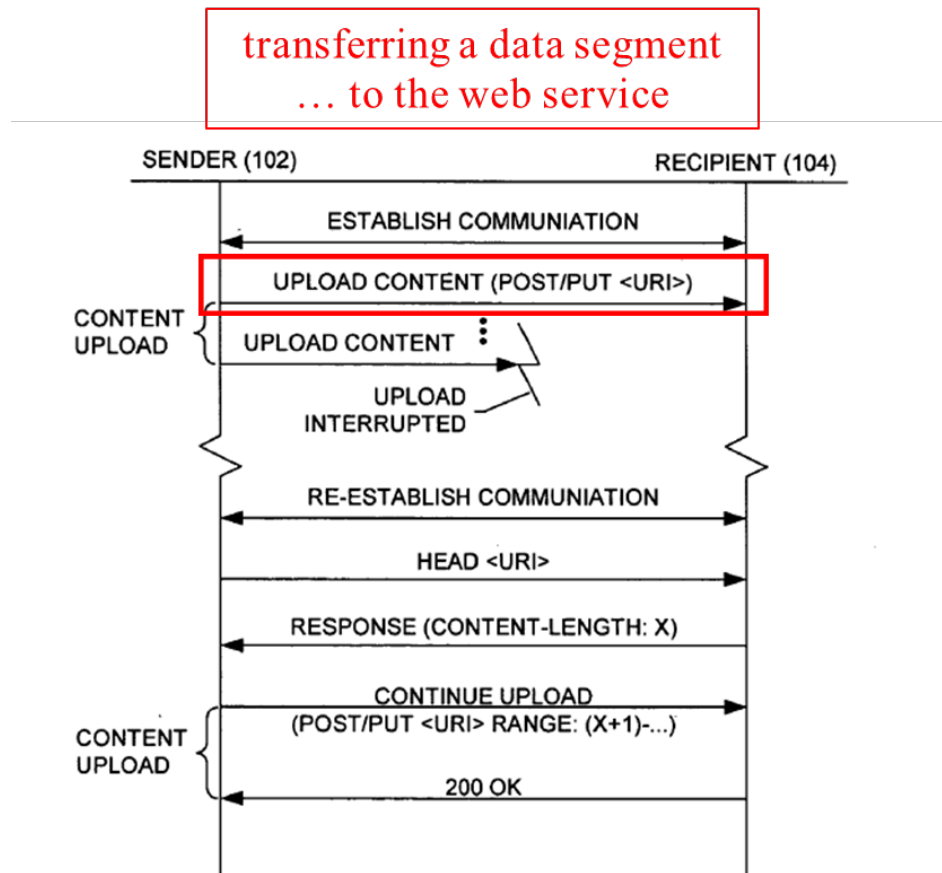


FIG. 11.

TTI-1005, FIG. 11.

63. As discussed above, each “block” of packets includes at least one “*user identifier*” and “*segment identifier*.” See above 1[d]-1[e]. This information is transferred “*along with*” the block as the identifier information is used by the recipient’s upload agent 110 for block identification and disposition (e.g., to “identify if, and when, an upload session has experienced an interruption”). TTI-1005, ¶¶[0102]-[0113].

1[f2]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service;

64. The Aaltonen-Drescher combination renders obvious element 1[f2]. For example, Aaltonen discloses that the identifying information added to information or data packets for each data block (“***configuring the data segment***”) can include “a table uniquely identifying the block of packets before or after the respective information packet, ... and the correct order of the packets.” TTI-1005, ¶[0103]. A POSITA would have recognized and found obvious that this identifying information would have been used by the Aaltonen-Drescher combination’s upload agent 110 to reassemble received data blocks in the correct order when storing the content. Indeed, Aaltonen discloses that “[b]y directly or indirectly identifying the order of the data packets, the upload agent can subsequently order the packets in the same order that the recipient [*sic* – ‘sender’] uploaded the packets” (“***configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service***”). TTI-1005, ¶[0103].

65. Additionally, Aaltonen discloses that “the content manager can process ... content accordingly before uploading the content (“***configuring the data segment***”). TTI-1005, ¶¶[0089]-[0092]. For example, the recipient can specify an “upload schedule” that includes “instructions to which uploading of the content is

subject” (“*configuring the data segment so that the data segment can be received by the web service*”). *Id.*

1[g]. receiving a reply message from the web service, corresponding to successful completion of said data segment transfer;

66. The Aaltonen-Drescher combination renders obvious element 1[g]. For example, Aaltonen discloses that, during an upload, the recipient’s upload agent can “acknowledge the upload” by sending “an acknowledgement to the terminal, or more particularly the content manager 106” (“*a reply message from the web service*”). TTI-1005, ¶¶[0093]-[0094], [0096], [0101]. A POSITA would have recognized or found obvious that, for segmented uploads, the upload agent 110 would have sent an acknowledgement to the content manager 106 for each “*successful completion of said data segment transfer,*” for example, to indicate to the sender’s content manager 106 that a data “block” has been received successfully and can subsequently be deleted from the sender’s 102 content storage 108. TTI-1005, ¶¶[0093]-[0094], [0096], [0101]. Indeed, Aaltonen discloses that acknowledgements indicate to the content manager 106 that it can “begin uploading content 114 to the recipient 104” and “delete or otherwise remove the upload content from the content storage 108, thereby relieving at least a portion of the storage capacity burden on the sender, such as for subsequent storage of other content.” TTI-1005, ¶¶[0093]-[0094], [0096], [0101].

67. To the extent Aaltonen does not explicitly disclose that an acknowledgement is sent for each data segment transfer, Drescher renders this obvious. TTI-1010, 11:27-12:10, 15:23-16:16. Drescher discloses a “HTTP PUT” technique (like Aaltonen) for uploading “chunks” of content where the server sends a “HTTP Response” to the sender’s PUT request that includes “the server’s log of progress of processing the PUT Request.” TTI-1010, 11:27-12:10, 15:23-16:16. In the Aaltonen-Drescher combination, Drescher’s “HTTP Response” would have been used by the recipient to “acknowledge” (via the “progress” indicated by a “HTTP Response”) a successful transmission of a “block” or “segment” sent using a “HTTP PUT” technique ⁴ (*“a synchronization message from the web service, corresponding to successful completion of said data segment transfer”*). TTI-1010, 11:27-12:10, 15:23-16:16; TTI-1005, ¶¶[0016], [0101], [0111], [0113]. Indeed, because the HTTP response includes “the server’s log of progress of processing the PUT Request,” this response indicates the “progress” of the upload, and accordingly,

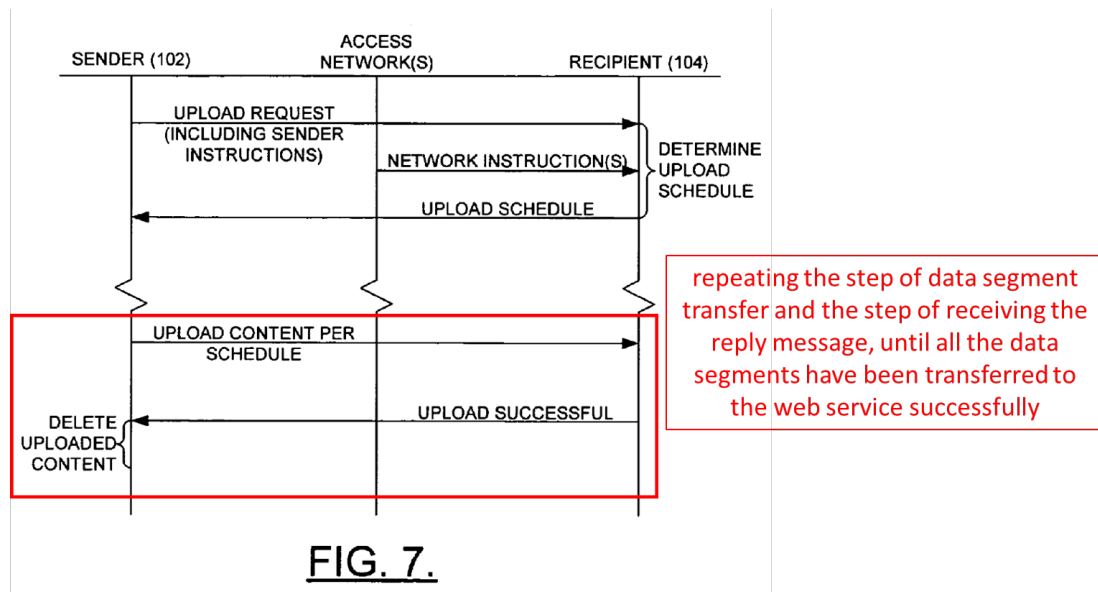
⁴ Although Drescher discloses its “HTTP response” with respect to a HTTP PUT request, responses could also easily be sent in response to other HTTP techniques, for example, Aaltonen’s “HTTP POST” technique. TTI-1005, ¶¶[0016], [0101], [0111], [0113].

a *successful completion of said data segment transfer* (and vice-versa) as a failed upload would be indicated in the server's processing log. *Id.*

68. Additionally, it was well known in the art to send acknowledgement messages for each portion of a segmented upload. As one example, Foo discloses a "buffer transfer service" for "portable electronic devices" where the "second device" (the receiver) sends an "acknowledgement (Ack) packet" for every correctly received data packet. TTI-1009, ¶¶[0002]-[0004], [0009], [0097]. Foo explains that sending either an "Ack" or a "Nack" packet in response to each received packet "help[s] ensure data integrity." TTI-1009, ¶¶[0009], [0097].

1[h]. repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully,

69. The Aaltonen-Drescher combination renders obvious element 1[h]. For example, Aaltonen discloses that an upload continues until all data blocks have been transferred ("*repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully*"). TTI-1005, ¶¶[0101]-[0114] ("the sender 102 can continue to upload the content to the recipient 104 until the upload session has concluded"); FIGS. 7, 9-11. Aaltonen's FIG. 11 illustrates an example upload process that experiences an interruption, while Aaltonen's FIG. 7 illustrates an upload with no such interruption.



TTI-1005, FIG. 7.

1[i]. wherein if segmented file transfer is interrupted, the software module further performs:

70. The Aaltonen-Drescher combination renders obvious element 1[i]. Aaltonen discloses various techniques for recovering from an “interruption” during an upload (“*segmented data transfer is interrupted*”). TTI-1005, ¶¶[0106]-[0114], FIGS. 9, 11. For example, Aaltonen’s FIG. 11 description discloses a process where, after an interruption, the sender “request[s] the bit-length or size of the portion of the upload content 114 received by the upload agent 110” and then “continue[s] uploading the content to the recipient 104” based on the response. TTI-1005, ¶¶[0111]-[0113], FIG. 11. A POSITA would have understood and found obvious that the above operations would have been performed by the sender’s 102 “content manager 106” (“*the software module further performs*”) as Aaltonen discloses that

the “sender 102 (e.g., terminal 10) ... generally operates under control of a computer program product (e.g., content manager 106 ...)” (“*software module*”). TTI-1005, ¶[0115].

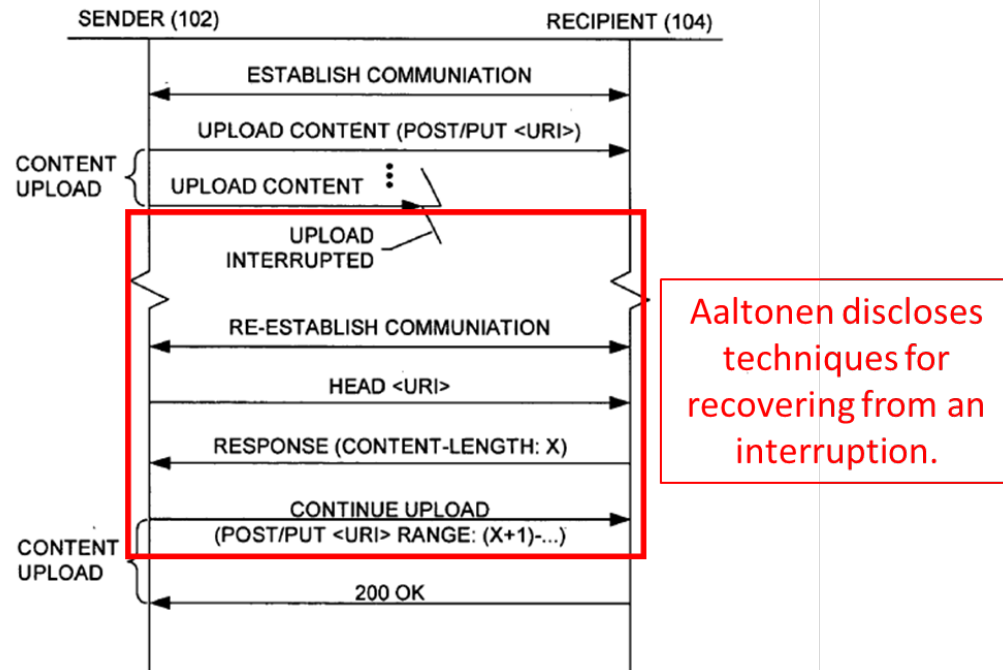


FIG. 11.

TTI-1005, FIG. 11.

1[j]. sending, a synchronization request to the web service from the application layer level, corresponding to asking the web service about the segmented file data the web service has successfully received, for said user identifier, after a network connection to the web service is restored;

71. The Aaltonen-Drescher combination renders obvious element 1[j]. For example, Aaltonen discloses that, upon experiencing an interruption, the sender can “re-establish communication with the recipient” and then “request the bit-length or size of the portion of the upload content 114 received by the upload agent 110”

(“*sending, a synchronization request to the web service ... corresponding to asking the web service about the segmented file data the web service has successfully received ... after a network connection to the web service is restored*”). TTI-1005, ¶[0111]-[0112]. The request from the sender would have been “*for said user identifier*,” for example, because the interrupted upload is associated with said user identifier. TTI-1005, ¶[0102], FIG. 11; *see above* 1[e].

72. Additionally, Aaltonen discloses that the “request” sent from the sender can be “in accordance with the HTTP HEAD technique” (“*from the application layer level*” – recall that HTTP operates at the application layer). TTI-1005, ¶[0112]; *see above* 1[c1].

73. Aaltonen’s FIG. 11 illustrates the sender re-establishing communication with the recipient and sending a synchronization message to the recipient following an interruption.

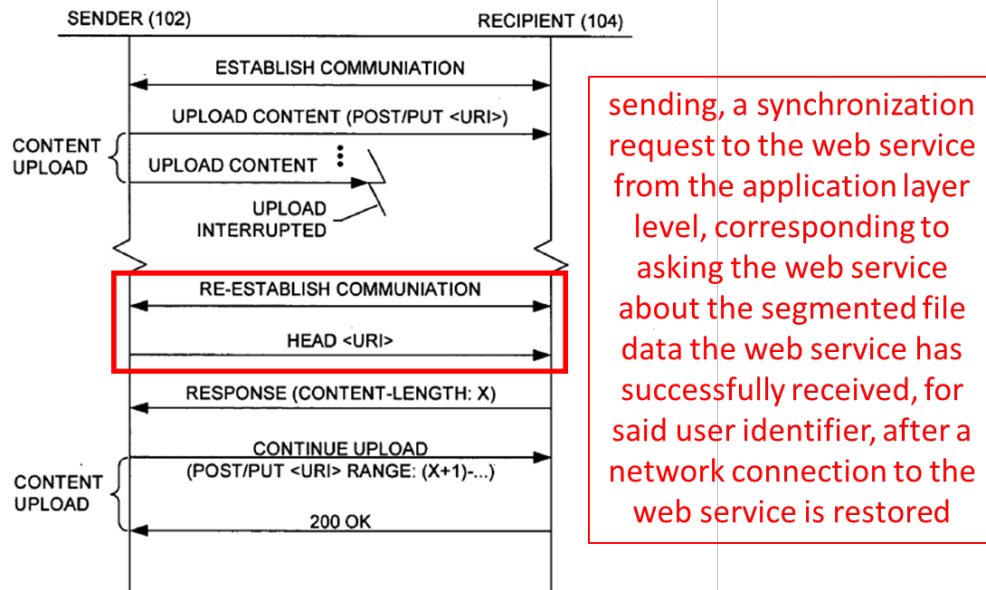


FIG. 11.

TTI-1005, FIG. 11.

1[k]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to indicate segmented file data said web service has successfully received, for said user identifier; and

74. The Aaltonen-Drescher combination renders obvious element 1[k]. For example, Aaltonen discloses that after receiving a synchronization request from a sender, the recipient (e.g., the upload agent) “can return a response, such as a 200 OK message, that includes header information (e.g., content-length) specifying the size of the received portion of the upload content” (“***receiving, a synchronization reply back from the web service ..., corresponding to indicate segmented file data said web service has successfully received***”). TTI-1005, ¶[0112]. Additionally, the “200 OK message” is a HTTP message sent “***at the application layer level***” in

response to a HTTP HEAD message. *Id.*; *see above* 1[c1]. A POSITA would have recognized or found obvious that the request from the sender would be “*for said user identifier*,” for example, because the interrupted upload and content is associated with said user identifier that identifies the sender of the upload content. TTI-1005, ¶[0102], FIG. 11; *see above* 1[e]. Aaltonen’s FIG. 11 illustrates the recipient sending a synchronization reply to the sender following a synchronization message.

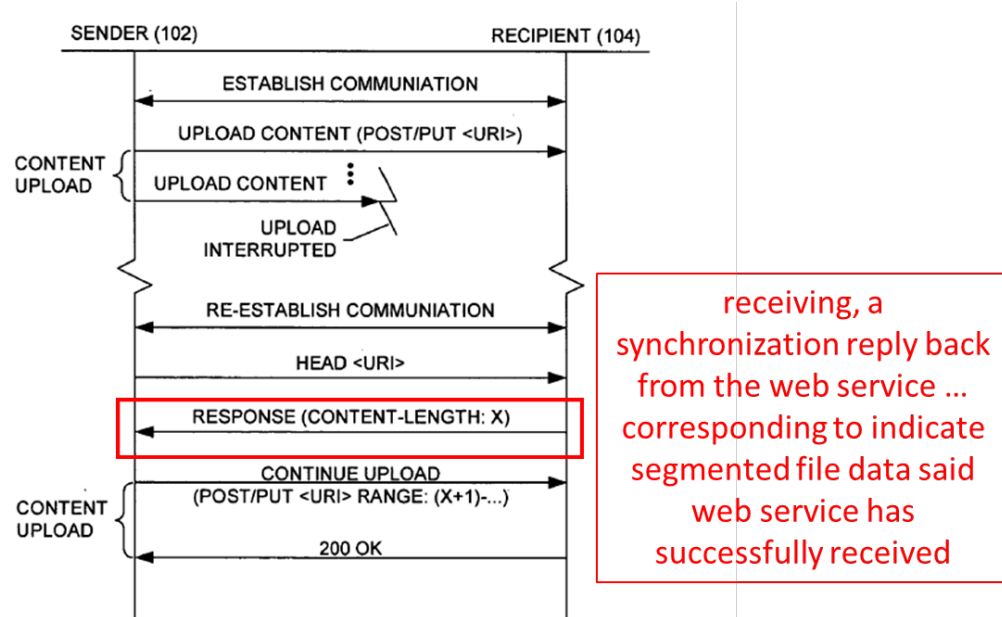


FIG. 11.

TTI-1005, FIG. 11.

1[l]. resuming, the segmented file transfer based on the synchronization reply received, for said user identifier.

75. The Aaltonen-Drescher combination renders obvious element 1[l]. For example, Aaltonen discloses that “[u]pon receiving the length of the received portion

of the upload content, the sender can continue uploading the content to the recipient 104 to complete the uploading session” (“*resuming, the segmented file transfer based on the synchronization reply received*”), or more specifically, the sender uploads “only the remaining portion of the content” (“a range of ‘X+1’ through the last bit of the upload content”). TTI-1005, ¶¶[0113]-[0114], FIG. 11. A POSITA would have recognized or found obvious that the sender resuming the upload would be “*for said user identifier*,” for example, because the upload is associated with said user identifier. TTI-1005, ¶[0102], FIG. 11; *see above* 1[e].

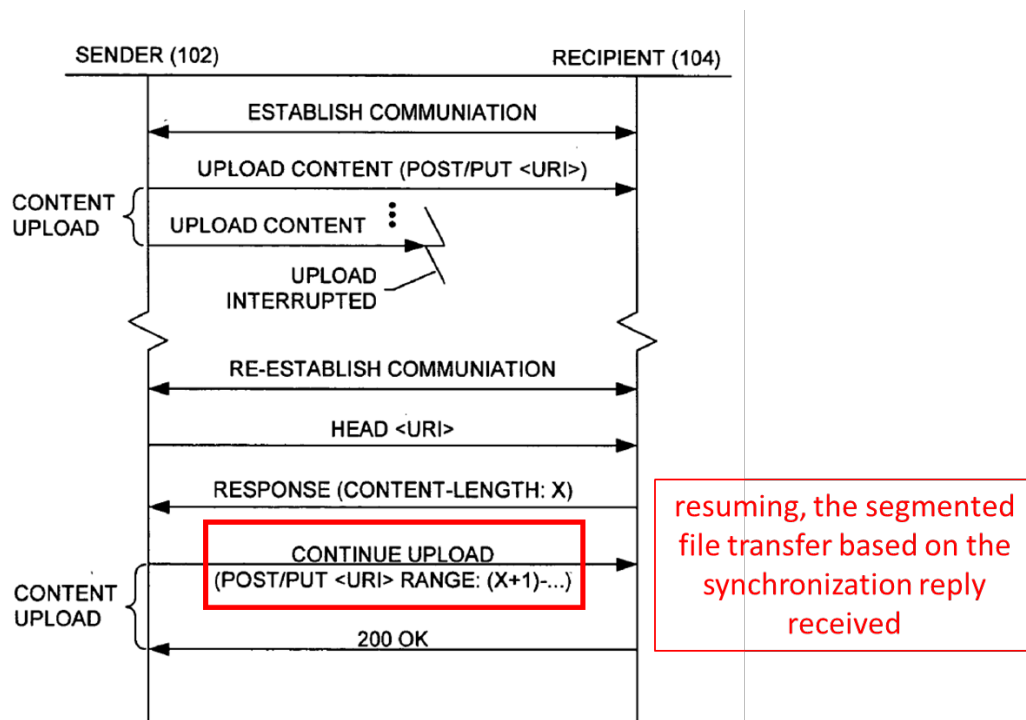


FIG. 11.

TTI-1005, FIG. 11.

2. The method of claim 1, wherein the data associated with file data is at least one of name of the file, size of the file, and type of file.

76. The Aaltonen-Drescher combination renders obvious claim 2. For example, as discussed above in 1[c3], a POSITA would have recognized or found obvious that “HTTP header information” (“*data associated with file data*”), for example, in Aaltonen-Drescher’s HTTP PUT or POST messages, would have included both the “*name of the file*” and “*type of the file*” being uploaded. TTI-1010, 32:25-28. Indeed, Gourley discloses that HTTP PUT messages include both the “*name of the file*” and “*type of the file.*” TTI-1007, 54-55.

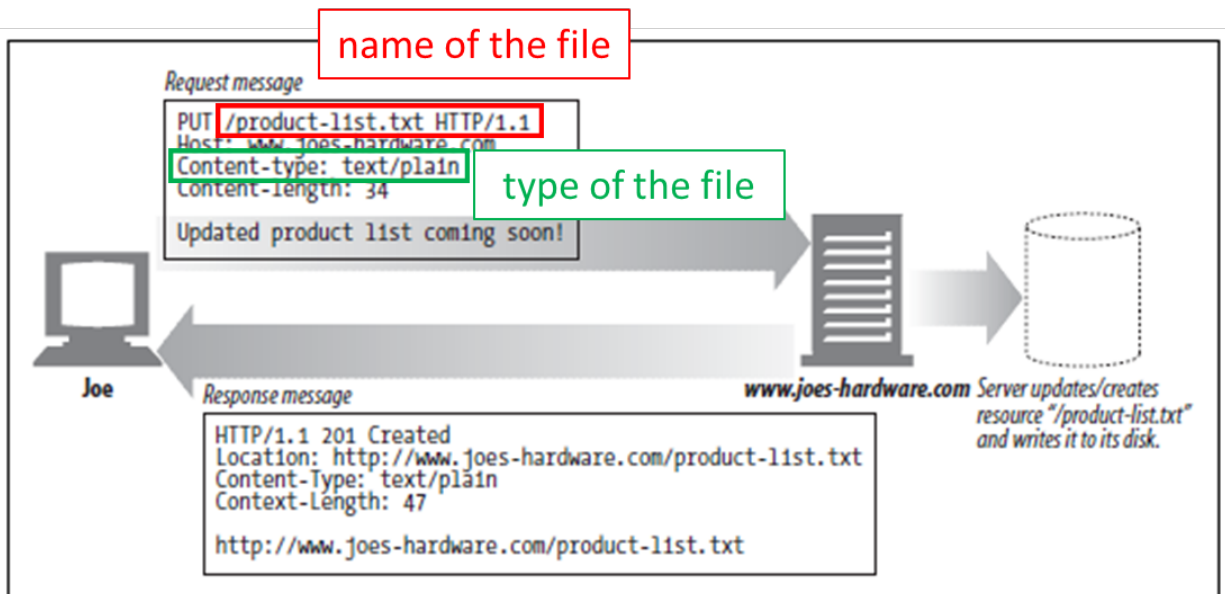


Figure 3-9. PUT example

TTI-1007, 55 (FIG. 3-9).

77. Additionally, Aaltonen discloses that an “instruction” can be defined “that orders or prioritizes piece(s) or object(s) within the upload content, such as

ordering graphic, audio and/or video content of the upload content” (“*type of file*”). TTI-1005, ¶[0072]. Indeed, Aaltonen provides examples where the upload and/or storing of content at the recipient changes based on the file type – indicated with *data associated with [the] multimedia data*. *Id.*, (disclosing examples of “a sports video clip with audio commentary” where “the sender could order video at a higher priority” and “a song clip with an audio track and a video of the singer” where “the sender may order the audio a higher priority”). A POSITA would have recognized or found obvious that the type of a file would have been indicated in the Aaltonen-Drescher HTTP message headers, as it was well known before the Critical Date that HTTP message headers contained this information. TTI-1007, 54-55.

3. The method of claim 1, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

78. The Aaltonen-Drescher combination renders obvious claim 3. For example, Aaltonen discloses that uploaded data (“*said file data*”) can include “graphic, audio and/or video content” (“*one or more of audio data, video data, image data*”) and “text content” (“*text data*”). TTI-1005, ¶¶[0049], [0068], [0070], [0072], [0084]. Additionally, a POSITA would have understood and found obvious that Aaltonen would have transferred “*digital data*” at least because Aaltonen discloses “MPEG-2 transport streams” – a common digital video coding standard. TTI-1005, ¶¶[0049]-[0050].

4. The method of claim 1, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

79. The Aaltonen-Drescher combination renders obvious claim 5. For example, Drescher discloses an example of a user identifier that includes “a username and password pair” (“*wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, [or] user-password*”). TTI-1010, 25:32-26:21; *see above* 1[e].

5. The method of claim 1, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

80. The Aaltonen-Drescher combination renders obvious claim 5. For example, as discussed above in 1[c2], instructions can be defined “based upon the connectivity of the recipient and/or respective senders,” and Aaltonen discloses that the terminal is “capable of transmitting and/or receiving data, content or the like in accordance with... a cellular (e.g., 1G, 2G, 2.5G, 3G, etc.) communication technique” (“*said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection*”). TTI-1005, ¶¶[0011], [0039], [0049], [0053], [0066], [0071], [0091]-[0092]; *see above* 1[c2].

6. The method of claim 1, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

81. The Aaltonen-Drescher combination renders obvious claim 6. For example, as discussed above in 1[c1], Aaltonen uses HTTP, which a POSITA would have understood and found obvious is an “*Internet protocol suite*” that segments data at the “*application layer*.” TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1007, 76, 249-251. Moreover, it was well known in the art before the Critical Date that the “*application layer*” was “*layer 7 in the OSI layer stack*.” TTI-1015, 2:58-4:40 (describing the “OSI model”).

7. The method of claim 1, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment.

82. The Aaltonen-Drescher combination renders obvious claim 7. For example, as discussed above in 1[d], Aaltonen discloses that data packets can include “an identifier that identifies placement of the respective data packet in relation to the other data packets of the content” (*wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment*). TTI-1005, ¶¶[0101]-[0104]; *see above* 1[d].

8[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprises:

83. To the extent the preamble is limiting, the Aaltonen-Drescher combination renders obvious element 8pre[pre] for similar reasons as discussed above for 1[pre].

8[a]. determining size of file to be transferred by the Bluetooth enabled mobile device;

84. The Aaltonen-Drescher combination renders obvious element 8[a] in multiple ways for similar reasons as discussed above for 1[b].

8[b]. transferring said file by the Bluetooth enabled mobile device to the web service, wherein transferring said file comprises, the Bluetooth enabled mobile device, performing:

85. The Aaltonen-Drescher combination renders obvious element 8[b]. For example, Aaltonen discloses a method of “uploading content” (“***transferring said file***”) from a “mobile station” and/or “terminal 10” (“***by the Bluetooth enabled mobile device ... the Bluetooth enabled mobile device, performing***”) to a “recipient” that can be an “origin server 22,” “digital broadcaster 32,” and/or “digital broadcast receiver 36” (“***to the web service***”). TTI-1005, ¶¶[0002]-[0007], [0008], [0038]-[0046], [0056], [0097], [0111]-[0115], FIGS. 1, 5, 9-11; *see above* 1[pre].

8[c1]. segmenting said file into data segments at an application layer level;

86. The Aaltonen-Drescher combination renders obvious element 8[c1] for similar reasons as discussed above for 1[c1].

8[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

87. The Aaltonen-Drescher combination renders obvious element 8[c2] for similar reasons as discussed above for 1[c2].

8[c3]. wherein data in said data segments comprises one or more of said file data and data associated with said file data;

88. The Aaltonen-Drescher combination renders obvious element 8[c3] for similar reasons as discussed above for 1[c3].

8[d]. applying a segment identifier to each of said data segments at the application layer level;

89. The Aaltonen-Drescher combination renders obvious element 8[d] for similar reasons as discussed above for 1[d].

8[e]. applying a user identifier to each of said data segments for the destination web service at the application layer level, wherein said user identifier uniquely identifies a particular user of the web service;

90. The Aaltonen-Drescher combination renders obvious element 8[e] for similar reasons as discussed above for 1[e].

8[f1]. transferring a data segment along with said user identifier and said segment identifier to the web service,

91. The Aaltonen-Drescher combination renders obvious element 8[f1] for similar reasons as discussed above for 1[f1].

8[f2]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service;

92. The Aaltonen-Drescher combination renders obvious element 8[f2] for similar reasons as discussed above for 1[f2].

8[g]. receiving a reply message from the web service, corresponding to successful completion of said data segment transfer;

93. The Aaltonen-Drescher combination renders obvious element 8[g] for similar reasons as discussed above for 1[g].

8[h]. repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully,

94. The Aaltonen-Drescher combination renders obvious element 8[h] for similar reasons as discussed above for 1[h].

8[i]. wherein if segmented file transfer is interrupted, the Bluetooth enabled mobile device further performs:

95. The Aaltonen-Drescher combination renders obvious element 8[i] for similar reasons as discussed above for 1[i].

8[j]. sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;

96. The Aaltonen-Drescher combination renders obvious element 8[j] for similar reasons as discussed above for 1[j].

8[k]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said file by said web service, for said user identifier; and

97. The Aaltonen-Drescher combination renders obvious element 8[k] for similar reasons as discussed above for 1[k].

8[l]. resuming, the segmented file transfer by transferring remaining data of the file based on the synchronization reply received, for said user identifier.

98. The Aaltonen-Drescher combination renders obvious element 8[1] for similar reasons as discussed above for 1[1].

9. The method of claim 8, wherein the data associated with file data is at least one of name of the file, size of the file, and type of file.

99. The Aaltonen-Drescher combination renders obvious claim 9 for similar reasons as discussed above for claim 2.

10. The method of claim 8, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

100. The Aaltonen-Drescher combination renders obvious claim 10 for similar reasons as discussed above for claim 3.

11. The method of claim 8, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

101. The Aaltonen-Drescher combination renders obvious claim 11 for similar reasons as discussed above for claim 4.

12. The method of claim 8, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

102. The Aaltonen-Drescher combination renders obvious claim 12 for similar reasons as discussed above for claim 5.

13. The method of claim 8, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

103. The Aaltonen-Drescher combination renders obvious claim 13 for similar reasons as discussed above for claim 6.

14. The method of claim 8, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment.

104. The Aaltonen-Drescher combination renders obvious claim 14 for similar reasons as discussed above for claim 7.

15[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:

105. To the extent the preamble is limiting, the Aaltonen-Drescher combination renders obvious element 15[pre] for similar reasons as discussed above for 1[pre].

15[a]. providing a software module on the Bluetooth enabled mobile device, the software module, performing;

106. The Aaltonen-Drescher combination renders obvious element 15[a] for similar reasons as discussed above for 1[a].

15[b]. determining size of a file to be transferred;

107. The Aaltonen-Drescher combination renders obvious element 15[b] in multiple ways for similar reasons as discussed above for 1[b].

15[c1]. segmenting said file into a first data segment and a second data segment at an application layer level;

108. The Aaltonen-Drescher combination renders obvious element 15[c1]. For example, as discussed above in 1[c1], Aaltonen discloses “breaking ... content into a plurality of portions” (“**segmenting said file into ... data segment[s] at an application layer level**”). TTI-1005, ¶[0012], [0069], FIG. 10; *see above* 1[pre], 1[c1]. As evident in Aaltonen’s FIG. 10, the “upload content 114” includes a plurality of data “blocks,” including at least a “**first**” and a “**second**” data segment. TTI-1005, ¶[0012], [0069], FIG. 10; *see above* 1[c1]. Additionally, segmenting data for uploads was well known in the art prior to the Critical Date. *See above* 1[c1].

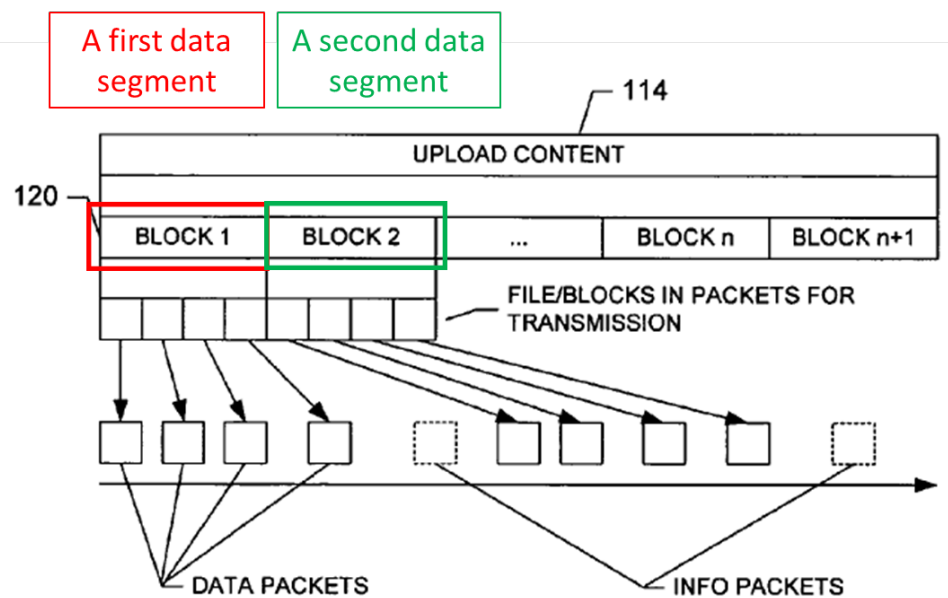


FIG. 10.

TTI-1005, FIG. 10.

109. Additionally, because the Aaltonen-Drescher combination uses HTTP, a POSITA would have understood and found obvious that data would have been segmented “*at an application layer level.*” TTI-1005, ¶¶[0016], [0056], [0098], [0101], [0111]-[0113]; TTI-1007, 76, 249-251.

15[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

110. The Aaltonen-Drescher combination renders obvious element 15[c2] for similar reasons as discussed above for 1[c2].

15[d]. wherein the second data segment comprises file data and the first data segment comprises data associated with file data, and wherein the data associated with file data is at least one of name of the file, size of the file, and type of the file;

111. The Aaltonen-Drescher combination renders obvious element 15[d]. For example, as discussed in claim 2, a POSITA would have recognized or found obvious that “HTTP header information,” for example, in Aaltonen-Drescher’s HTTP PUT or POST messages (“*data segments*” to include the “*first data segment*”), would have included both the “*name of the file*” and “*type of the file*” being uploaded. Indeed, Gourley discloses that HTTP PUT messages include both the “*name of the file*” and “*type of the file.*” TTI-1007, 54-55.

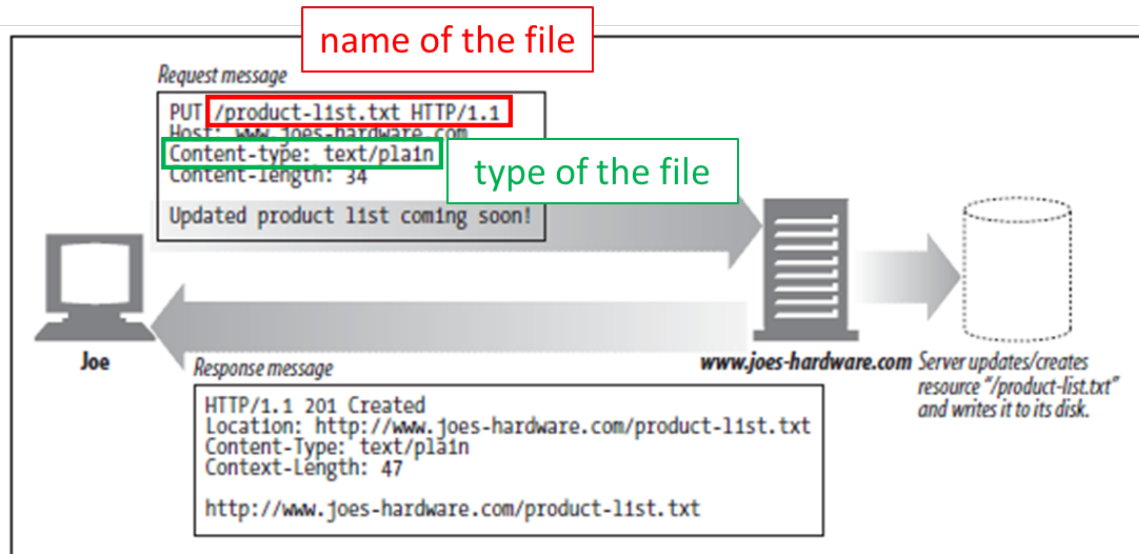


Figure 3-9. PUT example

TTI-1007, 55 (FIG. 3-9).

112. Additionally, Aaltonen discloses that an “instruction” can be defined “that orders or prioritizes piece(s) or object(s) within the upload content, such as ordering graphic, audio and/or video content of the upload content” (thus indicating the “*type of file*” associated with the “*file data*”). TTI-1005, ¶[0072]. Indeed, Aaltonen provides examples where the upload and/or storing of content at the recipient changes based on the file type. *Id.*, (disclosing examples of “a sports video clip with audio commentary” and “a song clip with an audio track and a video of the singer”); *see above* 1[c1] and claim 2. A POSITA would have recognized or found obvious that the type of a file would have been indicated in the Aaltonen-Drescher HTTP message headers, as it was well known before the Critical Date that HTTP message headers contained this information. TTI-1010, 32:25-28; TTI-1007, 54-55.

113. As discussed above, the Aaltonen-Drescher combination’s data blocks (“*data segments*”), to include the “*second data segment[s]*,” include “*file data*.” See above 1[pre], 1[c1].

15[e1]. applying a user identifier to said first data segment at the application layer level, wherein the user identifier uniquely identifies a particular user of the web service;

114. The Aaltonen-Drescher combination renders obvious element 15[e1] for similar reasons as discussed above for 1[c1], 1[e], and 15[c1]. For example, because Aaltonen-Drescher’s “*user identifier[s]*” can be applied to each block, they can be applied to “*said first data segment*.” See above 1[c1], 1[e], and 15[c1].

15[e2]. transferring said first data segment along with said user identifier to the web service,

115. The Aaltonen-Drescher combination renders obvious element 15[e2] for similar reasons as discussed above for 1[c1], 1[f1], and 15[c1]. For example, Aaltonen’s FIG. 10 illustrates the “*first data segment*” as part of the “upload content 114” that is transferred to the recipient. TTI-1005, FIG. 10; see above 1[c1], 1[f1], 15[c1].

15[e3]. receiving a reply message from the web service, corresponding to successful completion of said first data segment transfer;

116. The Aaltonen-Drescher combination renders obvious element 15[e3] for similar reasons as discussed above for 1[c1], 1[g], and 15[c1]. For example, as discussed above, the recipient in the Aaltonen-Drescher combination sends a “*reply*

message” for each “**successful completion of [a] data segment transfer**” to include a “**first**” segment. *See above*, 1[c1], 1[g], 15[c1].

15[f1]. applying said user identifier to said second data segment at the application layer level;

117. The Aaltonen-Drescher combination renders obvious element 15[f1] for similar reasons as discussed above for 1[c1], 1[e], and 15[c1]. For example, because Aaltonen-Drescher’s “**user identifier[s]**” can be applied to each block, they can be applied to “**said second data segment.**” *See above* 1[c1], 1[e], 15[c1].

15[f2]. transferring said second data segment along with said user identifier to the web service;

118. The Aaltonen-Drescher combination renders obvious element 15[f2] for similar reasons as discussed above for 1[c1], 1[f1], and 15[c1]. For example, Aaltonen’s FIG. 10 illustrates the “**second data segment**” as part of the “upload content 114” that is transferred to the recipient. TTI-1005, FIG. 10; *see above* 1[c1], 1[f1], and 15[c1].

15[f3]. receiving a reply message from the web service, corresponding to successful completion of said second data segment transfer,

119. The Aaltonen-Drescher combination renders obvious element 15[f3] for similar reasons as discussed above for 1[c1], 1[g] and 15[c1]. For example, as discussed above, the recipient in the Aaltonen-Drescher combination sends a “**reply message**” for each “**successful completion of [a] data segment transfer**” to include a “**second**” segment. *See above*, 1[c1], 1[g], and 15[c1].

15[g]. wherein if said second data segment transfer is interrupted, the software module further performs:

120. The Aaltonen-Drescher combination renders obvious element 15[g] for similar reasons as discussed above for 1[i] and 15[c1]. As discussed above, the Aaltonen-Drescher combination would have recovered from interruptions for the “*second*” data segment of the upload content in a similar fashion to the “first” segments. *See above* 1[i].

15[h]. sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;

121. The Aaltonen-Drescher combination renders obvious element 15[h] for similar reasons as discussed above for 1[j].

15[i]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said second data segment by said web service, for said user identifier; and

122. The Aaltonen-Drescher combination renders obvious element 15[i] for similar reasons as discussed above for 1[j] and 15[c1]. As discussed above, the Aaltonen-Drescher combination would have sent “*a synchronization reply back from the web service*” for the “*second*” data segment of the upload content. *See above* 1[j].

15[j]. resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received, wherein resuming the segmented file transfer of remaining data comprises, the software module, performing:

123. The Aaltonen-Drescher combination renders obvious element 15[j]. For example, as discussed in 1[l], Aaltonen discloses that “[u]pon receiving the length of the received portion of the upload content, the sender can continue uploading the content to the recipient 104 to complete the uploading session” (“***resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received***”), or more specifically, the sender uploads “only the remaining portion of the content.” TTI-1005, ¶¶[0113]-[0114]; *see above* 1[l].

15[k]. creating one or more data segments at the application layer level, wherein the one or more data segments comprise remaining data of the second data segment;

124. The Aaltonen-Drescher combination renders obvious element 15[k]. For example, when Aaltonen recovers from an upload, the sender 102 sends “only the remaining portion of the content” (thus, ***creating one or more data segments at the application layer level*** (per 15[c1]), ***wherein the one or more data segments comprise remaining data of the second data segment***). TTI-1005, ¶¶[0113]-[0114]. Moreover, a POSITA would have recognized or found obvious that the ***create[ed] one or more data segments*** would have included the ***remaining data of the second data segment*** at least because retransmitting successfully received data would be

inefficient. Further, because this claim limitation recites creating ‘one or more’ segments, simply transmitting the entirety of the ‘*remaining data of the second data segment*’ as one data segment would satisfy the claim. TTI-1005, ¶¶[0113]-[0114].

125. Additionally, because Aaltonen-Drescher uses HTTP, a POSITA would have understood and found obvious that data blocks would have been created “*at the application layer level.*” TTI-1007, 76, 249-251.

15[l]. applying a segment identifier to each of said one or more data segments at the application layer level;

126. The Aaltonen-Drescher combination renders obvious element 15[l] for similar reasons as discussed above for 1[d].

15[m]. applying said user identifier to each of said one or more data segments at the application layer level; and

127. The Aaltonen-Drescher combination renders obvious element 15[m] for similar reasons as discussed above for 1[e].

15[n]. transferring said one or more data segments along with said user identifier and said corresponding segment identifier to the web service,

128. The Aaltonen-Drescher combination renders obvious element 15[n] for similar reasons as discussed above for 1[f1].

15[o]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service.

129. The Aaltonen-Drescher combination renders obvious element 15[o] for similar reasons as discussed above for 1[f2].

16. The method of claim 15, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

130. The Aaltonen-Drescher combination renders obvious claim 16 for similar reasons as discussed above for claim 3.

17. The method of claim 15, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

131. The Aaltonen-Drescher combination renders obvious claim 17 for similar reasons as discussed above for claim 4.

18. The method of claim 15, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

132. The Aaltonen-Drescher combination renders obvious claim 18 for similar reasons as discussed above for claim 5.

19. The method of claim 15, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

133. The Aaltonen-Drescher combination renders obvious claim 19 for similar reasons as discussed above for claim 6.

20. The method of claim 15, wherein the segment identifier corresponds to the content range of the remaining data of the file.

134. The Aaltonen-Drescher combination renders obvious claim 20 for similar reasons as discussed above for claim 7.

21[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:

135. To the extent the preamble is limiting, the Aaltonen-Drescher combination renders obvious element 21[pre] for similar reasons as discussed above for 1[pre].

21[a]. determining size of file to be transferred by the Bluetooth enabled mobile device;

136. The Aaltonen-Drescher combination renders obvious element 21[a] in multiple ways for similar reasons as discussed above for 1[b].

21[b]. transferring said file by the Bluetooth enabled mobile device to the web service, wherein transferring said file comprises, the Bluetooth enabled mobile device, performing:

137. The Aaltonen-Drescher combination renders obvious element 21[b] for similar reasons as discussed above for 8[b].

21[c1]. segmenting said file into a first data segment and a second data segment at an application layer level,

138. The Aaltonen-Drescher combination renders obvious element 21[c1] for similar reasons as discussed above for 15[c1].

21[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

139. The Aaltonen-Drescher combination renders obvious element 21[c2] for similar reasons as discussed above for 1[c2].

21[d]. wherein the second data segment comprises file data and the first data segment comprises data associated with file data, and wherein the data

associated with file data is at least one of name of the file, size of the file, and type of the file;

140. The Aaltonen-Drescher combination renders obvious element 21[d] for similar reasons as discussed above for 15[d].

21[e1]. applying a user identifier to said first data segment at the application layer level, wherein the user identifier uniquely identifies a particular user of the web service;

141. The Aaltonen-Drescher combination renders obvious element 21[e1] for similar reasons as discussed above for 15[e1].

21[e2]. transferring said first data segment along with said user identifier to the web service,

142. The Aaltonen-Drescher combination renders obvious element 21[e2] for similar reasons as discussed above for 15[e2].

21[e3]. receiving a reply message from the web service, corresponding to successful completion of said first data segment transfer;

143. The Aaltonen-Drescher combination renders obvious element 21[e3] for similar reasons as discussed above for 15[e3].

21[f1]. applying said user identifier to said second data segment at the application layer level;

144. The Aaltonen-Drescher combination renders obvious element 21[f1] for similar reasons as discussed above for 15[f1].

21[f2]. transferring said second data segment along with said user identifier to the web service;

145. The Aaltonen-Drescher combination renders obvious element 21[f2] for similar reasons as discussed above for 15[f2].

21[f3]. receiving a reply message from the web service, corresponding to successful completion of said second data segment transfer,

146. The Aaltonen-Drescher combination renders obvious element 21[f3] for similar reasons as discussed above for 15[f3].

21[g]. wherein if said second data segment transfer is interrupted, the Bluetooth enabled mobile device performs:

147. The Aaltonen-Drescher combination renders obvious element 21[g] for similar reasons as discussed above for 15[g].

21[h]. sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;

148. The Aaltonen-Drescher combination renders obvious element 21[h] for similar reasons as discussed above for 1[j].

21[i]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said second data segment by said web service, for said user identifier; and

149. The Aaltonen-Drescher combination renders obvious element 21[i] for similar reasons as discussed above for 15[i].

21[j]. resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received, wherein resuming the segmented file transfer of remaining data comprises, the Bluetooth enabled mobile device, performing:

150. The Aaltonen-Drescher combination renders obvious element 21[j] for similar reasons as discussed above for 15[j].

21[k]. creating one or more data segments at the application layer level, wherein the one or more data segments comprise remaining data of the second data segment;

151. The Aaltonen-Drescher combination renders obvious element 21[k] for similar reasons as discussed above for 15[k].

21[l]. applying a segment identifier to each of said one or more data segments at the application layer level;

152. The Aaltonen-Drescher combination renders obvious element 21[l] for similar reasons as discussed above for 15[l].

21[m]. applying said user identifier to each of said one or more data segments at the application layer level; and

153. The Aaltonen-Drescher combination renders obvious element 21[m] for similar reasons as discussed above for 1[e].

21[n]. transferring said one or more data segments along with said user identifier and said corresponding segment identifier to the web service,

154. The Aaltonen-Drescher combination renders obvious element 21[n] for similar reasons as discussed above for 1[f1].

21[o]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service.

155. The Aaltonen-Drescher combination renders obvious element 21[o] for similar reasons as discussed above for 1[f2].

22. The method of claim 21, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

156. The Aaltonen-Drescher combination renders obvious claim 22 for similar reasons as discussed above for claim 3.

23. The method of claim 21, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

157. The Aaltonen-Drescher combination renders obvious claim 23 for similar reasons as discussed above for claim 4.

24. The method of claim 21, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

158. The Aaltonen-Drescher combination renders obvious claim 24 for similar reasons as discussed above for claim 5.

25. The method of claim 21, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

159. The Aaltonen-Drescher combination renders obvious claim 25 for similar reasons as discussed above for claim 6.

26. The method of claim 21, wherein the segment identifier corresponds to the content range of the remaining data of the file.

160. The Aaltonen-Drescher combination renders obvious claim 26 for similar reasons as discussed above for claim 7.

VIII. ANALYSIS OF TAKAHASHI AND NA, AND TAKAHASHI, NA, AND LIND

A. Overview of Takahashi

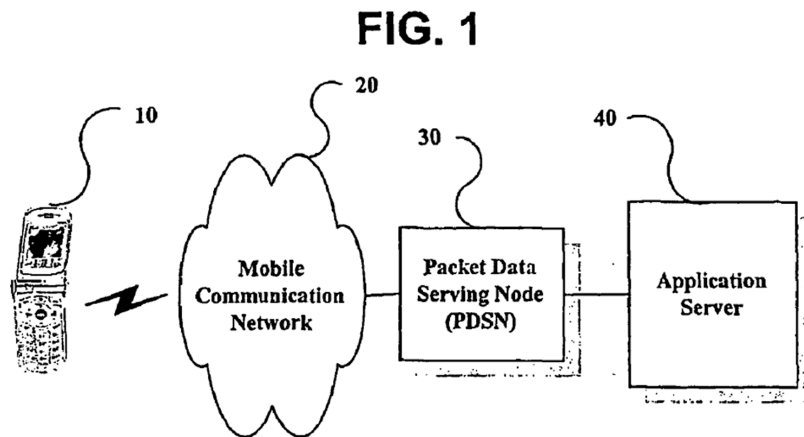
161. Takahashi discloses a mobile phone provided with a division means for dividing image data into a predetermined size and a transmission means for sequentially transmitting the image data divided by the division means to a server via a network by an HTTP request. TTI-1016, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049), (0057), (0080). Takahashi's mobile phone "adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion" on the server side. TTI-1016, ¶(0058). The mobile phone repeats transmissions of divided image data until all the image data is transmitted. TTI-1016, ¶¶(0068), (0072), FIG. 8. The server receives the HTTP request from the mobile phone and "combin[es] the divided image data, if the image data has been divided, to be one image data... to be published to a web page." TTI-1016, ¶¶(0061)-(0063), (0051), (0073), (0075)-(0079), (0089), FIG. 9.

162. In Takahashi, if "the size of image data is larger than a predetermined size, the mobile phone 1 automatically divides the captured image data into

predetermined size [data] and sequentially transmits the divided image data to a preset transmission destination (image storage server...) as an HTTP request.” TTI-1016, ¶¶(0049), (0057) (“an image dividing unit 17 for dividing the image data into multiple [data] for every predetermined size when the image data is equal to or larger than a predetermined size (for example, 10 KB)”), (0080) (“the captured image data is divided into predetermined size (for example, 9 kb) [data] and sequentially transmitted as an HTTP request to the image storage server 4”). Takahashi explains that in the prior art, “there is a problem that a high-definition image cannot be transmitted to a server from a mobile telephone terminal since the data more than a predetermined size (typically 10 KB) cannot be transmitted due to a restriction on the service provided by a communication provider.” TTI-1016, ¶(0003). Using Takahashi’s mobile terminal, “a high-definition image can be transmitted even if there is a limit to the size of data that can be transmitted.” TTI-1016, ¶¶(0030), (0034), (0051) (“Since the image data is transmitted after being divided, high definition images can be transmitted and received without being influenced by any restriction (10 KB or less) in terms of the service provided by a communication provider.”), [0080] (“Large-capacity image data with high image quality can be transmitted without being influenced by a service limitation posed by a communication provider.”).

B. Overview of Na

163. Similar to Takahashi, Na discloses techniques for “controlling a media message upload” by exchanging “requests and responses [] between a communication terminal and a server.” TTI-1006, Abstract, ¶[0001]. As an example, Na discloses “the communication terminal uploads the media data by organizing it into a plurality of segments.” TTI-1006, Abstract. An example system is illustrated in Na’s FIG. 1 that includes “a mobile communication terminal 10” and “an application server 40.” TTI-1006, Abstract, ¶¶[0051]-[0052], FIG. 1.



TTI-1006, FIG. 1.

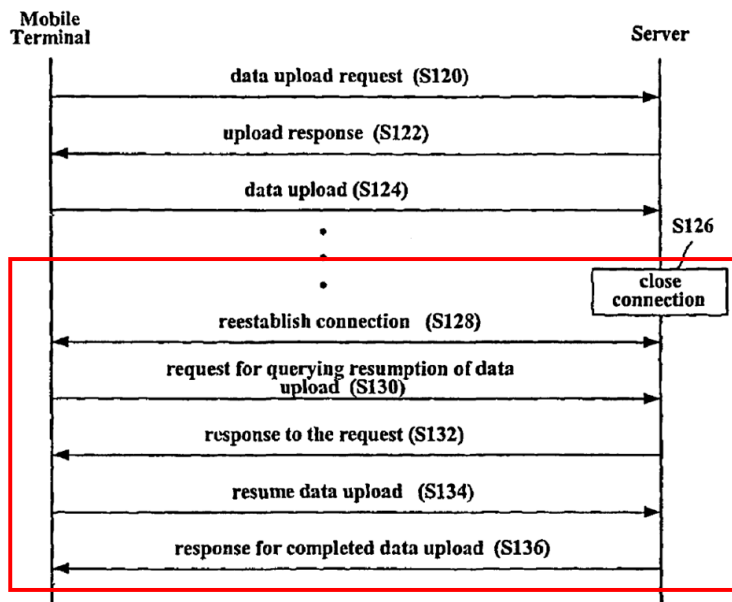
164. Na discloses that the HTTP protocol is used for the media data uploading. *See e.g.*, TTI-1006, ¶¶[0052], [0062]. Na explains that, per the HTTP 1.1 protocol, the media data is segmented, data segments are transmitted in request messages, and “response messages are repeatedly issued ... for verification of the upload.” TTI-1006, ¶¶[0003]-[0004]. It had been known that “a request message

and a response message need to be issued each time a segment of the media data is uploaded.” TTI-1006, ¶¶[0003]-[0005].

165. Na also discloses techniques for recovering from an “interruption” such as “an upload error” or “a closed connection” during an upload. TTI-1006, ¶¶[0054], [0070], [0084]-[0087], [0091]-[0094], [0113]-[0115], [0139]-[0157], FIGS. 11-12, 24-25. For example, Na’s “FIG. 12 illustrates a flowchart of the procedure for resuming an upload of media data when the communication connection is reestablished,” which includes that the “mobile communication terminal 10 sends a request message querying the sequence number of the segment from which the upload should be resumed to the application terminal 40,” “the application server 40 issues a response message for the query” including “the sequence number of the segment from which the upload should be resumed,” and the “mobile communication terminal 10 resumes the upload with reference to the sequence number included in the response message from the application server 40” until “the resumed media data upload is completed.” TTI-1006, ¶¶[0036], [0091]-[0097]. As another example, Na’s FIG. 25 “illustrates a flowchart of the procedure for resuming an upload of media data stopped by a user,” which includes that “the mobile communication terminal 10 issues a request message for querying where to resume the upload” (S130), “the application server 40 issues a response message [indicating] the range of data that has been uploaded already,” (S132), and the mobile

communication terminal 10 “begins uploading a portion of the media data that needs to be uploaded additionally (S134),” until “the resumed upload of the media data is completed normally.” TTI-1006, ¶¶[0049], [0148]-[0157].

FIG. 25



Na discloses techniques for resuming an upload of media data from an interruption, e.g., a closed connection

TTI-1006, FIG. 25.

C. Overview of Lind

166. Lind describes “electronic devices which are capable of being interconnected by means of direct short range radio communication, such as by means of Bluetooth. Such devices include all mobile radio terminals devised for radio communication with a radio station, which radio station also may be mobile terminal or e.g. a stationary base station. Consequently, the term radio terminal

includes mobile telephones.” TTI-1017, 7:27-8:3⁵. In Lind, “a first electronic device in the form of a mobile telephone, and a second electronic device in the form of a passive accessory, such as a headset or a hands free kit” is “provided with means for communicating by radio, such as a Bluetooth connection.” TTI-1017, 8:11-16.

D. A POSITA would have combined Takahashi and Na

167. As discussed above, Takahashi discloses transmitting divided media data using HTTP requests. TTI-1016, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049). Takahashi does not appear to disclose details regarding the HTTP protocol and HTTP requests for transmitting divided media data. As such, a POSITA would have been motivated and would have found it obvious and straightforward to complement Takahashi with Na’s detailed disclosure of the HTTP protocol and the HTTP requests used for media data uploading. TTI-1006, ¶¶[0052], [0062].

168. Further, Takahashi recognizes instances where a “communication is not completed normally.” TTI-1016, ¶(0071). Likewise, a POSITA would have recognized that, in mobile communications such as Takahashi’s, “a data upload error caused by various communication problems may occur during an upload of media data,” and “it is desirable to resume the upload in such a way that the upload is

⁵ The citations to Lind refer to the page number of the PDF document.

resumed from the point at which the error occurred.” TTI-1006, ¶¶[0006]-[0008]. A POSITA would have been motivated and would have found it obvious and straightforward to incorporate Na’s teachings in Takahashi’s system to beneficially allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0008], [0010]. In doing so, a POSITA also would have been motivated and found it obvious and straightforward to implement Takahashi’s mobile phone to include a division/segment identifier with each divided media data to be uploaded so that the media data is “easily identified [by the server] when the data upload is stopped and resumed afterwards” and to receive from the server “a parameter about the range” of the divided media data “that has been successfully uploaded... when an upload error occurs” so that the mobile phone knows what part of the divided media data is to be uploaded additionally, as taught by Na. TTI-1006, ¶¶[0012]-[0013], [0022]. In this way, Takahashi’s mobile phone need not restart upload of the whole media data but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021].

169. As explained below in more detail, combining Takahashi and Na would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield

predictable results and (2) use of known techniques to improve similar devices in the same way.

170. Indeed, Takahashi already discloses transmitting divided media data using HTTP requests. TTI-1016, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049). Na also describes data transfers using HTTP requests according to a segmented data transfer protocol, such as HTTP 1.1 protocol. TTI-1006, ¶¶[0003]-[0004], [0052]-[0053]. As such, the techniques of Takahashi and Na would have been compatible, without requiring undue experimentation, and implementing Takahashi's mobile device based on Na's teachings to also allow "an upload of media data stopped by an upload error to be resumed from the point where the error occurred," as taught by Na (TTI-1006, ¶¶[0008], [0010]), would have simply involved implementing a known feature in Takahashi's similar system, making the Takahashi-Na combination predictable and foreseeable with a reasonable expectation of success. *Id.*

E. A POSITA would have combined Takahashi, Na, and Lind

171. Takahashi's techniques are implemented in a mobile phone. BD-1016, ¶¶(0049)-(0053), FIGS. 1-3. Similarly, Na's FIG. 1 shows a mobile phone as an example of "mobile communication terminal 10." TTI-1006, ¶[0051]. A POSITA would have recognized and found obvious that one functionality of a mobile phone is communication via Bluetooth. By the Critical Date, Bluetooth had been a

common feature of mobile phones. TTI-1009, ¶¶[0016], [0030]-[0031], [0042]-[0043], [0049], [0054]; TTI-1011, ¶[0034], Claim 6; TTI-1012; TTI-1005, ¶¶[0056], [0036]-[0037], [0046]; TTI-1018, ¶¶[0106]-[0107], [0165].

172. For example, Lind describes “electronic devices which are capable of being interconnected by means of direct short range radio communication, such as by means of Bluetooth. Such devices include all mobile radio terminals devised for radio communication with a radio station, which radio station also may be mobile terminal or e.g. a stationary base station. Consequently, the term radio terminal includes mobile telephones.” TTI-1017, 7:27-8:3. In Lind, “a first electronic device in the form of a mobile telephone, and a second electronic device in the form of a passive accessory, such as a headset or a hands free kit” is “provided with means for communicating by radio, such as a Bluetooth connection.” TTI-1017, 8:11-16.

173. A POSITA would have been motivated to include Bluetooth functionalities in the Takahashi-Na mobile phone to beneficially provide the mobile phone with capabilities for communicating with other Bluetooth accessories such as Bluetooth headsets commonly available at the time of the ’757 patent. Indeed, a POSITA would have recognized that a mobile phone “allows voice communications between the mobile communication terminal... and another mobile communication terminal,” as explicitly disclosed in Na (TTI-1006, ¶[0056]), for which a Bluetooth headset would beneficially provide a wireless hands-free means of voice

communication, as taught by Lind. TTI-1017, 6:9-10, 8:11-16, 9:12-16, 16:7-8. As such, a POSITA would have found it obvious to implement the Takahashi-Na mobile phone to be a Bluetooth enabled mobile device that supported Bluetooth functionalities, as taught by Lind. *Id.*

174. Additionally, implementing the Takahashi-Na mobile phone as a Bluetooth enabled mobile device would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) use of known techniques to improve similar devices in the same way. Such a modification to the Takahashi-Na mobile phone would have simply involved implementing Bluetooth functionality into Takahashi-Na's mobile phone, as taught by Lind, making the modification predictable and foreseeable with a reasonable expectation of success.

F. Claims 1-26 are rendered obvious by Takahashi in view of Na, and claims 1-26 are rendered obvious by Takahashi in view of Na and Lind.

175. Below, I discuss a ground based on the combination of Takahashi and Na, and a ground based on the combination of Takahashi, Na, and Lind. In the first ground, I explain why Takahashi's and Na's disclosures, when interpreted in view of the general knowledge of a POSITA, renders obvious claims 1-26. In the second ground, the I rely on the same disclosures of Takahashi and Na and explanation of obviousness as presented in the first ground, but additionally turn to Lind's

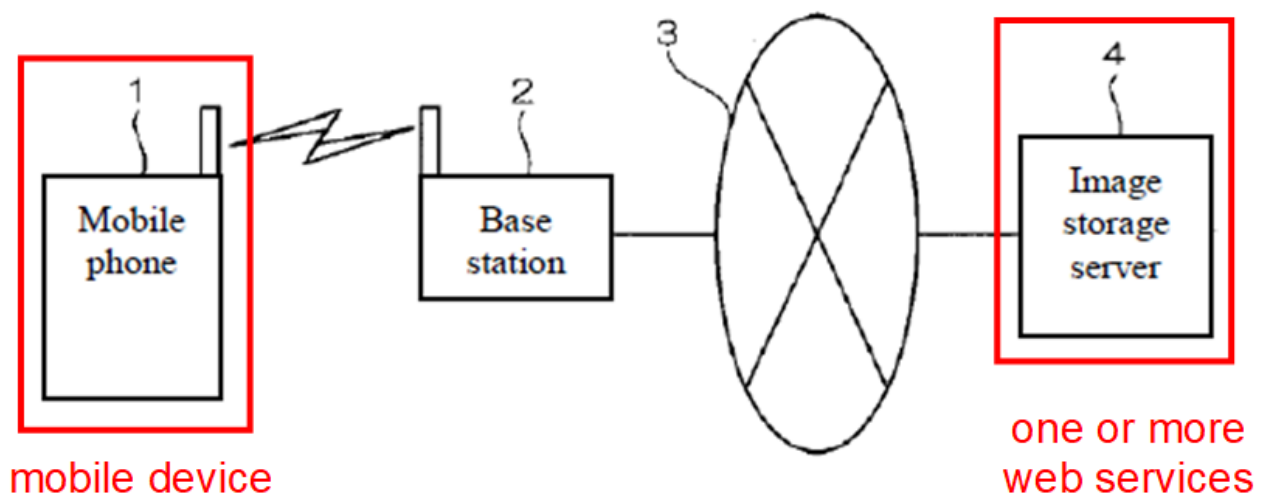
disclosure for a single feature of claims 1-26 (“*a Bluetooth enabled mobile device*”). Although all of the features of claims 1-26 are rendered obvious by Takahashi and Na as explained below, I have provided additional analysis of Takahashi, Na, and Lind for claims 1-26 to the extent it is argued that this single feature of claims 1-26 would not be rendered obvious by Takahashi and Na. Because the use of Takahashi and Na and corresponding obviousness arguments are similar for both grounds, I discuss these grounds together for ease of presentation.

1[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:

176. To the extent the preamble is limiting, both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[pre]. For example, Takahashi discloses a method of transmitting “image data”/“image file” that are divided into “predetermined size [data]”/“divided file[s]” (“*segmented file transfer*”) from a “mobile telephone”/“mobile terminal” (“*mobile device*”) to “an image storage server” that provides web services such as “publish[ing] the image data” “to a web page” (“*one or more web services*”). TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049), (0054), (0061), FIG. 1. Takahashi discloses that “the captured **image data**” is stored in a **file** with a “a predetermined **file name**”). TTI-1016, ¶(0056). Takahashi further discloses that the mobile phone 1 “carr[ies] out division and transmission of the **image file**” and “adopts a sequential transmission rule for **transmitting a divided file** one at a time and transmitting the

next divided file”) (“**segmented file transfer**”), TTI-1016, ¶(0058). Takahashi’s image storage server includes “a WEB (World Wide Web) server” and publishes images “on a web page.” TTI-1016, ¶¶(0052), (0054), (0061)-(0062), FIG. 2. Takahashi discloses sequentially sending the divided image data “as an HTTP request.” TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049).

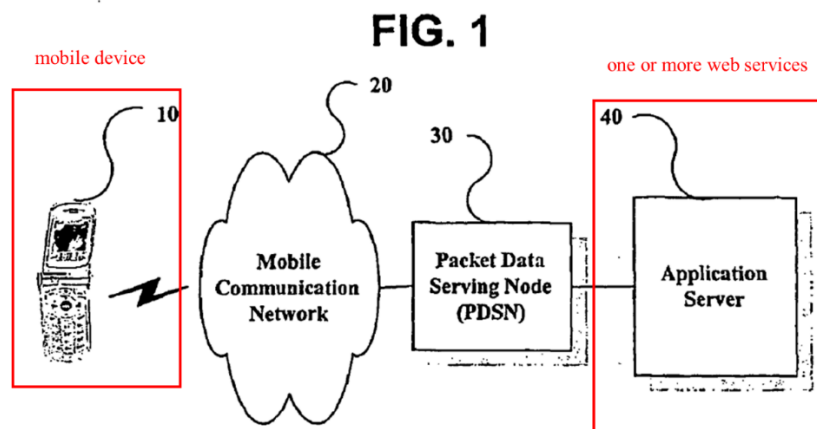
(FIG. 1)



TTI-1016, FIG. 1.

177. Similarly, Na discloses a “method for controlling a media message upload” (“**file transfer**”) that uploads “media data” from “a mobile communication terminal 10” (“**mobile device**”) to “an application server 40” (“**web service**”). TTI-1006, Abstract, ¶¶[0051]-[0052], FIGS. 1, 10-14, 24-25. Na discloses that data upload can be performed “in accordance with the HTTP 1.1 protocol,” “performing successive uploads of the segments” (“**segmented data transfer**”) of media data.

TTI-1006, ¶¶[0003]-[0004], [0022], [0052]-[0053], [0112]; *see below* 1[c1]. A POSITA would have recognized and found obvious that Na’s “media data upload” would be a *file* transfer because the data transfer over communication network is sometimes referred to as “file” transfer because the data would have been represented in a data structure that is generally referred to as a “data file” or simply “file.” *See e.g.*, TTI-1024, [0005] (referring to both “**share files** among computing devices across a network” and “**data to be shared**”), [0012] (disclosing “[i]n general, **the file is a collection of data** together with auxiliary data, if there is such auxiliary data.”). Moreover, Na corroborates this understanding and discloses that the **uploaded data** are stored as “a temporary **file**.” *See e.g.*, TTI-1006, Abstract, ¶[0068].



TTI-1006, FIG. 1.

178. Takahashi’s techniques are implemented in a mobile phone. BD-1016, ¶¶(0049)-(0053), FIGS. 1-3. Similarly, Na’s FIG. 1 shows a mobile phone as an example of “mobile communication terminal 10.” TTI-1006, [0051]. A POSITA

would have found it obvious and would have been motivated to implement Takahashi-Na's mobile phone to include Bluetooth functionality because, by the Critical Date, Bluetooth had been a common feature of mobile devices. TTI-1009, ¶¶[0016], [0030]-[0031], [0042]-[0043], [0049], [0054]; TTI-1011, ¶[0034], Claim 6; TTI-1012; TTI-1005, ¶¶[0056], [0036]-[0037], [0046]; TTI-1017, 3:4-22, 7:27-8:3; TTI-1018, [0106]-[0107], [0165].

179. To the extent that Takahashi-Na in view of the general knowledge of POSITA does not render obvious a Bluetooth enabled mobile device, Lind renders this obvious. As discussed above, Lind describes “electronic devices which are capable of being interconnected ... by means of Bluetooth... include[] mobile telephones.” TTI-1017, 7:27-8:3; *see above* §VIII.E. In Lind, “a mobile telephone” and “a headset or a hands free kit” are “provided with means for communicating by radio, such as a Bluetooth connection.” TTI-1017, 8:11-16. A POSITA would have been motivated to include Bluetooth functionalities in the Takahashi-Na mobile phone to beneficially provide the mobile phone with capabilities for communicating with other Bluetooth accessories such as Bluetooth headsets commonly available at the time of the '757 patent. *Id.* Indeed, a POSITA would have recognized that a mobile phone “allows voice communications between the mobile communication terminal... and another mobile communication terminal,” as explicitly disclosed in Na (TTI-1006, ¶[0056]), for which a Bluetooth headset would beneficially provide

a wireless hands-free means of voice communication. TTI-1017, 6:9-10, 8:11-16, 9:12-16, 16:7-8. As such, in view of the general knowledge of POSITA and Lind's explicit disclosure, a POSITA would have found it obvious to implement the Takahashi-Na mobile phone as a ***Bluetooth enabled mobile device*** that supported Bluetooth functionalities. *Id.*

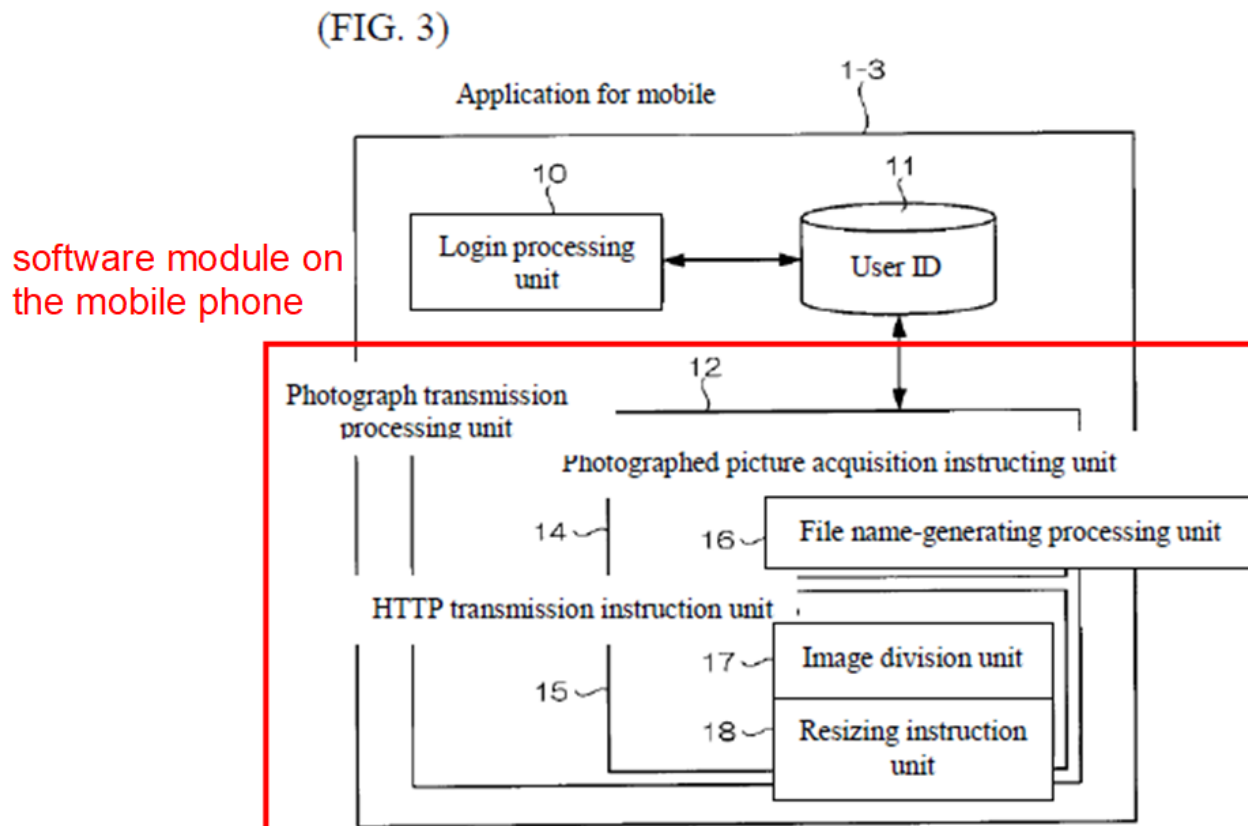
180. Additionally, as discussed above, a POSITA would have found it obvious to combine Takahashi-Na with Lind, and the predictable Takahashi-Na-Lind combination renders obvious the "***Bluetooth enabled mobile device***." See *above* §VIII.E.

181. The '757 patent does not define a "***web service***," but describes "publishing data and multimedia content on one or more websites" (a "***web service***"). TTI-1001, 3:30-40, *see also id.* 5:65-6:8. Similarly, Takahashi discloses its "image storage server" has "a WEB (World Wide Web) server" and publishes images "on a web page." TTI-1016, ¶¶(0052), (0054), (0061)-(0062), FIG. 2. Moreover, Na discloses that "media data uploads and downloads between a mobile communication terminal and a particular server on the network is also feasible through **web**/WAP connections. In such a wireless communication network, the HTTP 1.1 (rfc 2616) protocol is required for media data uploads from a mobile communication terminal to a server on the Internet." TTI-1006, ¶¶[0002]-[0003]. A POSITA would have recognized and found obvious that Takahashi's server would have hosted a website,

and accordingly, the server in Takahashi-Na would have been a “*web service*.” Indeed, Takahashi and Na each discloses operating over the “Internet,” which was well known in the art to host websites. TTI-1016, ¶(0050); TTI-1006, ¶¶[0002]-[0003].

1[a]. providing a software module on the Bluetooth enabled mobile device, the software module, performing;

182. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[a]. For example, Takahashi discloses a “software configuration of the mobile phone 1” that includes “an application for mobile [devices] 1-3 for performing the transmission control of the image data.” TTI-1016, ¶(0053), FIGS. 2-3. Takahashi’s application software is a ***software module***. As shown in FIG. 3 below, Takahashi’s application software includes “a photograph transmission processing unit 12” that includes “a photographed picture acquisition instruction unit 14 and an HTTP transmission instruction unit 15.” TTI-1016, ¶¶(0055)-(0056).



TTI-1016, FIG. 3.

183. Further, Na discloses a “method” performed by the “mobile communication terminal.” TTI-1006, ¶¶[0016]-[0020]. A POSITA would have recognized and found obvious that, in each of Takahashi-Na and Takahashi-Na-Lind, the mobile terminal would have included a *software module* to implement the disclosed method, for benefits such as flexibility and low cost of software implementations as opposed to hardware implementations. Moreover, it was common knowledge prior to the Critical Date that mobile terminals would include a software module to implement its functionalities. Aaltonen, for example, corroborates this understanding and discloses a “content manager 106” which is

“software” (“*a software module*”) of a mobile device that performs the disclosed method including functionalities such as formatting and uploading of content, deleting successfully uploaded content from the sender’s memory, and recovering from upload interruptions. TTI-1005, ¶¶[0060]-[0062], [0066], [0069], [0073], [0077]-[0080], [0084]-[0086], [0089]-[0094], [0111]-[0115].

1[b]. determining size of a file to be transferred;

184. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[b]. For example, Takahashi discloses, if “**the size of image data is larger than a predetermined size**, the mobile phone 1 automatically divides the captured image data into predetermined size [data] and sequentially transmits the divided image data to a preset transmission destination (image storage server...) as an HTTP request.” TTI-1016, ¶¶(0049), (0057), (0080). A POSITA would have recognized or found obvious that each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile phone (“*Bluetooth enabled device*” per 1[pre]) software application first determines the size of the image data (“*determining size of a file to be transferred*”) in order to determine whether to “automatically divide[] the captured image data into predetermined size [data].” *Id.*

185. Moreover, Na discloses when “uploading particular media data, the mobile communication terminal 10 issues an initial request message as shown in FIG. 2 comprising a command 100, a header 102, and a message body 104.” TTI-

1006, ¶[0062]. The header 102 includes “the number of total segments in the media data (X-Total-Seq, e.g., 10)” and “the length of one segment of the media data given in bytes (Content-Length...).” TTI-1006, ¶[0063]. A POSITA would have recognized or found obvious that the number of total segments to be transferred is the size of the whole data in bytes divided by the length of one segment in bytes. A POSITA would have also recognized or found obvious that the length of one segment to be transferred is the size of the whole media data divided by the number of total segments. *Id.* A POSITA would have recognized or found obvious that each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile terminal (“**Bluetooth enabled device**” per 1[pre]) software application first determines the size of the whole data (“**file**” per 1[pre]) in bytes (“**determining size of the file to be transferred**”) in order to insert the number of total segments to be transferred or the length of one segment into the header 102 of the request message when uploading the data. *Id.*

186. As another example, Na discloses “[w]hen uploading particular media data, the mobile communication terminal 10 issues a request message ... comprising ... a header 202” and “[t]he header 202 includes information on ... content length given in bytes (Contents-Length...)” (“**size of a file to be transferred**”). TTI-1006, ¶¶[0116]-[0117], [0122]. FIG. 15. A POSITA would have recognized or found obvious that each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile phone software application would have “**determined**” the size of the file to

be uploaded to insert the “content length” information in the header 202 of the request message when uploading the file.

1[c1]. segmenting said file into data segments at an application layer level;

187. As discussed above and in element 1[pre], both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[c1]. *See above* §§VIII.A-E. For example, Takahashi discloses “the mobile phone 1 automatically divides the captured image data into predetermined size [data]” (“***segmenting said file into data segments***”). *See e.g.*, TTI-1016, ¶¶(0049), (0057), (0060), (0080). Takahashi discloses the “divided image data” (“***data segments***”) are sequentially transmitted “by the application” for the mobile phone “as an HTTP request.” *Id.*; TTI-1016, (0053) (“an application for mobile [devices] 1-3 for performing the transmission control of the image data”), (0055)-(0057), (0068), (0085), (0090), FIG. 3. A POSITA would have understood and found obvious that Takahashi’s file would have been segmented “***at the application layer level***” because the file is segmented “by the application” for the mobile phone to be sequentially transmitted “as an HTTP request,” and HTTP is an application-level protocol. *Id.* Gourley corroborates this understanding and discloses that HTTP is part of the “application layer” (or “web application layer”). TTI-1007, 7, 76, 249-251. Moreover, during the prosecution history of the ’757 patent, the applicant acknowledged that “[i]t is well known that both HTTP and XML-RPC are application layer protocols.” TTI-1002, 115.

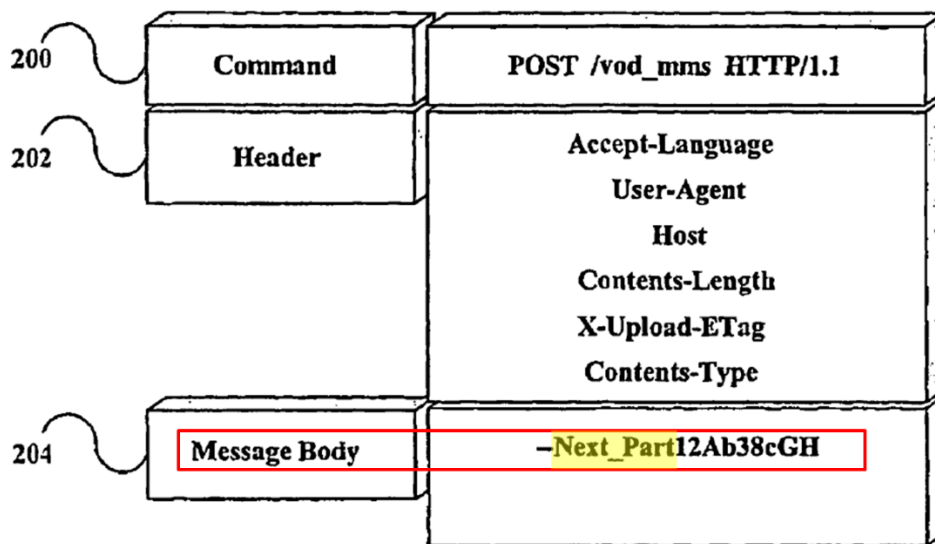
188. A POSITA would have understood that there were multiple different ways to format Takahashi's "divided image data" ("*data segments*") "as an HTTP request." For example, Na discloses that the "mobile communication terminal 10 connects to the application server 40 via the mobile communication network 20 and the wireless Internet and performs an operation for uploading particular media data stored in a memory thereof to the application server 40 in accordance with the HTTP 1.1 protocol." TTI-1006, ¶[0052]. "To upload the media data to the application server 40, the mobile communication terminal 10 issues an upload request message for each group of a predefined number (e.g., K) segments of the media data. In other words, the mobile communication terminal 10 sends an upload request message to the server 40 each time it uploads K segments (one segment is also called a sequence) of the media data." TTI-1006, ¶[0053]. As taught by this example of Na, each of Takahashi-Na's and Takahashi-Na-Lind's mobile phones would send an HTTP upload request message to the server for each "divided image data" ("*data segment*").

189. As another example, Na discloses uploading multiple parts/portions of "particular media data" ("*file*" per 1[pre]) by sending multiple request messages that include respective message bodies, where each message body is assigned "a unique ID of the message body" for uploading "**one portion**" (a "*segment*") of the data. TTI-1006, ¶[0112]. The "unique ID" is used to distinguish the "**one portion**" of the media data from "the **other portion**" of the media data. TTI-1006, ¶¶[0112]

(disclosing “allowing the application server 40 to match one portion of the message body that has been already uploaded and the other portion of the message body to be uploaded”), [0113] (disclosing “the mobile communication terminal 10 examines whether a portion of the media data that has been already uploaded exists in the application server 40 using the unique ID of the message body”).

190. Indeed, Na’s FIG. 15 shows an example “request message for a media data upload” that includes “a message body 204.” The message body 204 includes a *segment of the file*: “-Next_Part12Ab38cGH.” TTI-1006, ¶¶ [0039], [0116]-[0118].

FIG. 15



TTI-1006, FIG. 15.

191. Moreover, Na discloses “[t]he mobile communication terminal 10 and the application server 40 **repeatedly exchanges request messages including** the unique ID of **the message body** and response messages **during the upload of the**

media data.” TTI-1006, ¶¶ [0151]. As taught by this example of Na, each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile phones would have segmented image data (“*file*”) into divided image data (“*data segments*”) to upload each divided image data in respective “message bodies” of different “request messages,” each request message including “**the unique ID** of the message body” that identifies a specific portion of data sent in the message body of the respective request message.

192. Additionally, because Na discloses performing such operations “in accordance with the **HTTP** 1.1 protocol,” a POSITA would have understood and found obvious that data would have been segmented “*at the application layer level*” to be uploaded along with or included in multiple “request messages” pursuant to the HTTP protocol. TTI-1006, ¶¶[0052], [0111], [0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

1[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

193. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[c2]. For example, as discussed in 1[pre] and 1[c1], Takahashi discloses, “the mobile phone 1...sequentially transmits the divided image data to a preset transmission destination (image storage server...)” “as an HTTP request.” TTI-1016, ¶¶(0049), (0005)-(0008), (0020)-(0030), (0042)-(0043). Takahashi explicitly discloses the image data (“*said file*”) is divided (“*segmented*”) due in part to “the size of the image data” (“*size of said file*”) because when “the size of the

image data is larger than a predetermined size, the mobile phone 1 automatically divides the captured image data into predetermined size [data] and sequentially transmits the divided image data.” TTI-1016, ¶¶(0049), (0057), (0080).

194. A POSITA would have recognized or found obvious that the image data is divided (“*segmented*”) due in part to “a service limitation posed by a communication provider” to the mobile phone (“*data transfer capabilities of the Bluetooth enabled mobile device*”) because Takahashi discloses that the image data is divided (“*segmented*”) “**in consideration of** such circumstances” where the file size is large (e.g., “more than a predetermined size”) and the data transfer capability of the mobile terminal is limited “by a communication provider.” TTI-1016, ¶¶(0003)-(0008), (0051), (0080). Indeed, Takahashi discloses because of the consideration of both the file size and the limited data transfer capability of the mobile terminal, “the image data is transmitted after being divided,” and thus “high definition images can be transmitted and received without being influenced by any restriction (10 KB or less) in terms of the service provided by a communication provider” and advantageously allow “[l]arge-capacity image data with high image quality” to be transmitted to a server via a network by an HTTP request “without being influenced by a service limitation posed by a communication provider.” TTI-1016, ¶¶(0004), (0051), (0080).

195. Additionally, segmenting a file in part due to *file size and data transfer capabilities* was well known in the art. Indeed, as one example, Foo discloses that a “buffer transfer session ... used to transfer multiple data objects sequentially” can be used for “data objects whose size exceeds a certain threshold.” TTI-1009, ¶¶[0036], [0135]. As another example, Ganesan discloses that “the file 220 may be divided into any number of header and segment portions in consideration of a required transmission rate ..., and the minimum uploading and downloading capabilities of a network.” TTI-1024, ¶[0038]. As yet another example, Aaltonen discloses that “instructions” for uploading content can be defined “based upon the connectivity of the recipient and/or respective senders” and “available bandwidth to the recipient 104 and/or the sender 102” (“*data transfer capabilities of the Bluetooth enabled mobile device*”). TTI-1005, ¶¶[0011]-[0012], [0066], [0080], [0075]-[0079], [0091]-[0092].

1[c3]. wherein data in said data segments comprises one or more of said file data and data associated with said file data;

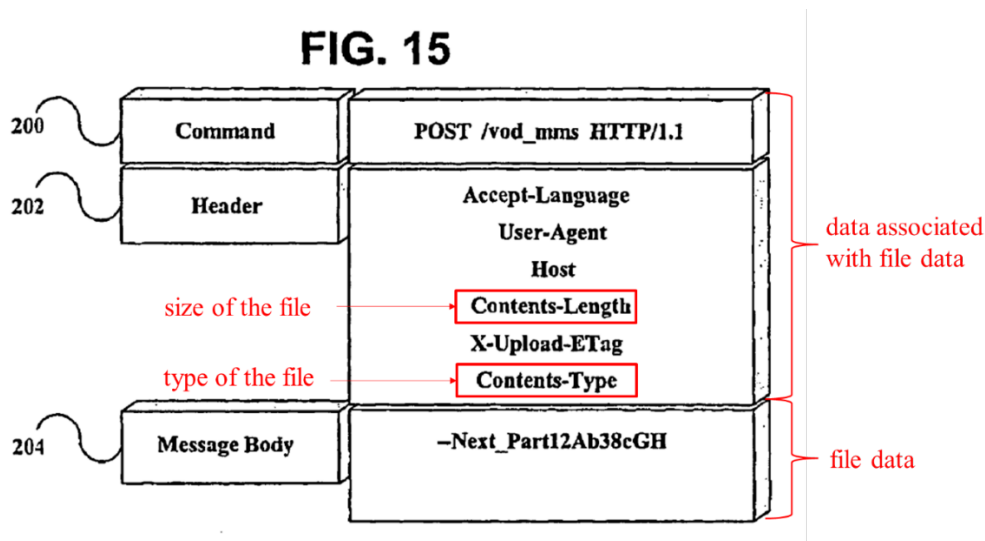
196. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[c3]. For example, Takahashi discloses that, at the mobile phone software application, the image data is divided and each data division (“*file data*”) is transmitted in a divided file (“*data segment*”) with a file name (“*data associated with said file data*”) so that “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is

written as an addition (combination) to the existing file.” TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070)-(0071), (0073), FIG. 5.

197. As discussed above in element 1[c1], Na also discloses that data in “group[s] of a predefined number (e.g., K) segments of the media data” and their corresponding request messages (“*data segments*” per 1[c1]) comprises “the media data” (“*file data*”) and data in the request messages such as “a command 100, a header 102” (“*data associated with said file data*”). TTI-1006, ¶¶[0053], [0062]. “The header 102 includes content type information (Contents-Type, e.g., application/vnd.Skt.message), the number of total segments in the media data (X-Total-Seq, e.g., 10), the sequence number of the segment to be uploaded (X-Seq-No, e.g., 1), a message ID assigned to the media data to be uploaded (X-Msg-id, e.g., a terminal number and upload time information in the form of MIN MMDDHHMMSS), the number of segments for which one acknowledgement response will be received from the application server 40 (X-Ack-Seq, e.g., 3), and the length of one segment of the media data given in bytes (Content-Length, e.g., 512).” TTI-1006, ¶[0063]; *see below* claim 2.

198. As another example, Na discloses a media data upload by sending data in multiple request messages (“*data segments*”) that include multiple parts/portions of “particular media data” (“*file data*”) as message bodies and “*data associated with said file data*” in the headers. TTI-1006, ¶[0112]-[0113], [0116]-[0118]. For

example, Na's FIG. 15 shows a "request message for a media data upload" that includes "a header 202, and a message body 204." TTI-1006, ¶¶ [0039], [0116]. The "message body 204" includes "*file data*": "-Next_Part12Ab38cGH." TTI-1006, ¶¶[0116]-[0118], FIG. 15. "The header 202 includes "*data associated with said file data*" such as "information on used languages (Accept-Language, e.g., en, ko), ..., content length given in bytes (Contents-Length, e.g., 1500), ... and contents type information (Contents-Type, e.g., multipart/mixed;boundary=Next_Part12Ab38cGH)." TTI-1006, ¶¶[0116]-[0117]; *see below* claim 2.



TTI-1006, FIG. 15.

199. A POSITA would have been motivated to implement Takahashi's mobile phone to transfer image data and data (e.g., a file name, a command, or a header with content type and content length) associated with the image data, as

suggested by Na, to facilitate the reception, storage, and publication of the image data at the server.

1[d]. applying a segment identifier to each of said data segments at the application layer level;

200. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[d]. For example, Takahashi discloses “[w]hen carrying out division and transmission of the image file from the mobile phone 1, the information showing the order of the file such as division number (Ex. 5/10, or the like)” (“***a segment identifier***”) “may be attached and sent, if needed” “in the HPPT [sic] transmission” that includes a division of the data (***applied to a data segment***). TTI-1016, ¶(0058). Additionally, as discussed in 1[c1], Takahashi discloses each data division is transmitted by the application of the mobile phone “as an HTTP request.” TTI-1016, ¶¶(0049), (0053) (“an application for mobile [devices] 1-3 for performing the transmission control of the image data”), (0055)-(0057), (0068), (0085), (0090), FIG. 3. A POSITA would have understood and found obvious that the division number would have been applied to each data division “***at the application layer level***” pursuant to the HTTP protocol. TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

201. A POSITA would have understood that there were multiple different ways to apply Takahashi’s division number (“***segment identifier***”) to each divided image data (“***data segment***”). For example, Na discloses that when “uploading

particular media data, the mobile communication terminal 10 issues an initial request message as shown in FIG. 2 comprising a command 100, a header 102, and a message body 104.” TTI-1006, ¶[0062]. As shown in FIG. 2, the header 102 includes “the sequence number of the segment to be uploaded (X-Seq-No, e.g., 1)” (*“a segment identifier”*). TTI-1006, ¶¶[0063], [0078].

202. Na further discloses that “the mobile communication terminal 10 issues an upload request message for each group of a predefined number (e.g., K) segments of the media data. In other words, the mobile communication terminal 10 sends an upload request message to the server 40 each time it uploads K segments (one segment is also called a sequence) of the media data.” TTI-1006, ¶[0053]. As taught by this example of Na, a POSITA would have understood or found obvious that each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile devices generates and sends an upload request message that includes the sequence/division number of the divided image data (*“segment”*) to be uploaded (*“applying a segment identifier to each data segment”*) so that Takahashi’s mobile device need not restart upload of the whole media data when the data upload is stopped but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021]; *see above* §VIII.D.

203. As another example, Na discloses “[w]hen generating a request message for uploading the particular media data, the mobile communication terminal

10 assigns a unique ID of the message body to the header of the message” (“*applying a segment identifier to each data segment*”). TTI-1006, ¶[0112]. As shown in FIG. 15, Na discloses “a request message ... comprising ... a header 202” that includes “**the ID of the message body** to be uploaded (**X-Upload-Etag**, e.g., a terminal number and upload time information in the form of MIN_MMDDHHMMSS such as 0114261166–0414232012),” (“*a segment identifier to each data segment*”). TTI-1006, ¶¶[0116]-[0117], FIG. 15.

204. Based on Na’s disclosure that “the mobile communication terminal issues a request message each time it uploads a segment of the media data in accordance with the HTTP 1.1 protocol,” a POSITA would have recognized and found obvious that a unique ID would have been applied to “*each*” message body of each request message because the ID is “unique” for each message body “so that the message body can be easily identified when the data upload is stopped and resumed afterwards.” TTI-1006, ¶¶[0012], [0022]. Moreover, a POSITA would have recognized and found obvious that this unique ID would preferentially be applied to “*each*” message body of each request message because Na relies upon this information to determine “whether a portion of the media data that has been already uploaded exists in the application server 40.” TTI-1006, ¶[0113]. Neglecting to add this unique ID to any one request message would create the potential that a recipient would not be able to properly dispose of the message body of that request message.

As taught by this example of Na, a POSITA would have understood or found obvious that each of Takahashi-Na's and Takahashi-Na-Lind's mobile devices generates and sends an upload request message that includes the unique ID/division number of the divided image data ("*segment*") to be uploaded ("*applying a segment identifier to each data segment*") so that the divided image data can be easily identified when the data upload is stopped and resumed afterwards." TTI-1006, ¶¶[0012], [0022].

205. Additionally, as discussed in 1[c1]-[c2], Na discloses performing such an operation "in accordance with ... the **HTTP** 1.1 protocol," and a POSITA would have understood and found obvious that the division number included in the request message would have been applied to each divided image data "*at the application layer level*" in the multiple "request messages" pursuant to the HTTP protocol. TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

1[e]. applying a user identifier to each of said data segments for the destination web service at the application layer level, wherein said user identifier uniquely identifies a particular user of the web service;

206. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[e]. For example, Takahashi discloses that "a file name of the image data" contains at least "a user ID specific to a user from the server after logging-in" ("*user identifier... for each destination web service*" that "*uniquely identifies a particular user of the web service*"). TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0064), (0066), (0070), FIG. 5. The file name is "in the format of

user ID + YYYYMMDDHHMISS (year, month, day, hour, minute, and second).” TTI-1016, ¶¶(0056), (0084), (0087). Takahashi discloses that in “transmitting the divided image using HTTP, the generated file name” including the user ID “is also transmitted together” (“*applying a user identifier to each data segment*”) and “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file.” TTI-1016, ¶¶(0071), (0073).

207. Additionally, as discussed in 1[c1], Takahashi discloses each divided file is transmitted by the application of the mobile phone “as an HTTP request.” TTI-1016, ¶¶(0049), (0053) (“an application for mobile [devices] 1-3 for performing the transmission control of the image data”), (0055)-(0057), (0068), (0085), (0090), FIG. 3. A POSITA would have understood and found obvious that the file name including the user ID would have been applied to each divided file “*at the application layer level*” pursuant to the HTTP protocol. TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

208. A POSITA would have understood that there were multiple different ways to apply Takahashi’s file name including the user ID (“*user identifier*”) to each data segment. For example, Na discloses that when “uploading particular media data, the mobile communication terminal 10 issues an initial request message as shown in FIG. 2 comprising a command 100, a header 102, and a message body 104.”

TTI-1006, ¶[0062]. As shown in FIG. 2, the header 102 includes “a message ID assigned to the media data to be uploaded (X-Msg-id, e.g., **a terminal number** and upload time information **in the form of MIN_MMDDH HMMSS**).” TTI-1006, ¶¶[0063], [0078]. A POSITA would have understood or found it obvious that a terminal number in the form of MIN (mobile identification number) corresponds to the claimed “*user identifier*” that “*uniquely identifies a particular user of the web service*.” The POSITA’s knowledge is corroborated by multiple references. *Id.* For example, Chow describes “a mobile station with a mobile identification number (MIN)” and subscribers using their cellular phone and unique MIN for subscription to a service. TTI-1020, 4:7-10, 4:17-19, 2:60-63, 9:10-13, 12:46-49. As another example, Graves describes a “Mobile Identification Number (‘MIN’)” being used to identify a customer account. TTI-1021, 7:39-45. As yet another example, Hartmaier describes that the “unique Mobile Identification Number (MIN) assigned to each wireless telephone may be used to authenticate transactions for a particular account.” TTI-1022, 3:25-27.

209. Na further discloses that “the mobile communication terminal 10 issues an upload request message for each group of a predefined number (e.g., **K**) segments of the media data. In other words, the mobile communication terminal 10 sends an upload request message to the server 40 **each time** it uploads K segments (one segment is also called a sequence) of the media data.” TTI-1006, ¶[0053]. As

taught by this example of Na, a POSITA would have understood or found obvious that each of Takahashi-Na's and Takahashi-Na-Lind's mobile devices generates and sends to the server 40 an upload request message that includes the user ID for each divided file (*“applying a user identifier to each data segment for each destination web service”*).

210. As discussed in element 1[d], Na discloses another example of the “request message” destined to “the application server 40” including “header 202” that includes “user agent information (User-Agent, e.g., SKT/XXX200)” (*“applying a user identifier to each data segment for each destination web service”*). TTI-1006, ¶¶[0116]-[0117], FIG. 15. A POSITA would have understood and found obvious that the “user agent information (User-Agent, e.g., SKT/XXX200)” *“uniquely identifies a particular user of the web service”* because, per “the HTTP 1.1 (rfc 2616) protocol” used by Na, “[t]he **User-Agent** request-header field contains information about **the user agent originating the request,**” where the “**user agent**” is “[t]he client which initiates a request. These are often browsers, editors, spiders (web-traversing robots), or other end user tools,” which uniquely identifies a particular user that uploads data to the web service. TTI-1019, 8, 89.

211. As discussed in element 1[d], “*each*” request message would have included this “**user agent information,**” and thus “*each*” message body of each

request message (“*each data segment*”) is applied this “*user identifier*.” TTI-1006, ¶¶[0116]-[0117], FIG. 15; TTI-1019, 8, 89; *see above* 1[d]. As taught by this example of Na, a POSITA would have understood or found obvious that each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile devices generates and sends to the server a request message that includes the user ID for each divided file (“*applying a user identifier to each data segment for each destination web service*”).

212. Additionally, as discussed above in 1[c1]-1[d], a POSITA would have understood and found obvious that the identifier applied to the request message in accordance with HTTP 1.1 Protocol would have been “*at the application layer level*.” TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

1[f1]. transferring a data segment along with said user identifier and said segment identifier to the web service,

213. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[f1]. For example, as discussed above in 1[d]-1[e], Takahashi discloses that in “transmitting the divided image using HTTP, the generated file name” including the user ID “is also transmitted together” (“*transferring a data segment along with said user identifier*”) and “when carrying out division and transmission of the image file from the mobile phone 1, the information showing the order of the file such as division number (Ex. 5/10, or the like)” (“*a segment identifier*”) “may be attached and sent, if needed” “in the HPPT [*sic*] transmission” that includes a division of the data (“*transferring a data segment along with... said*

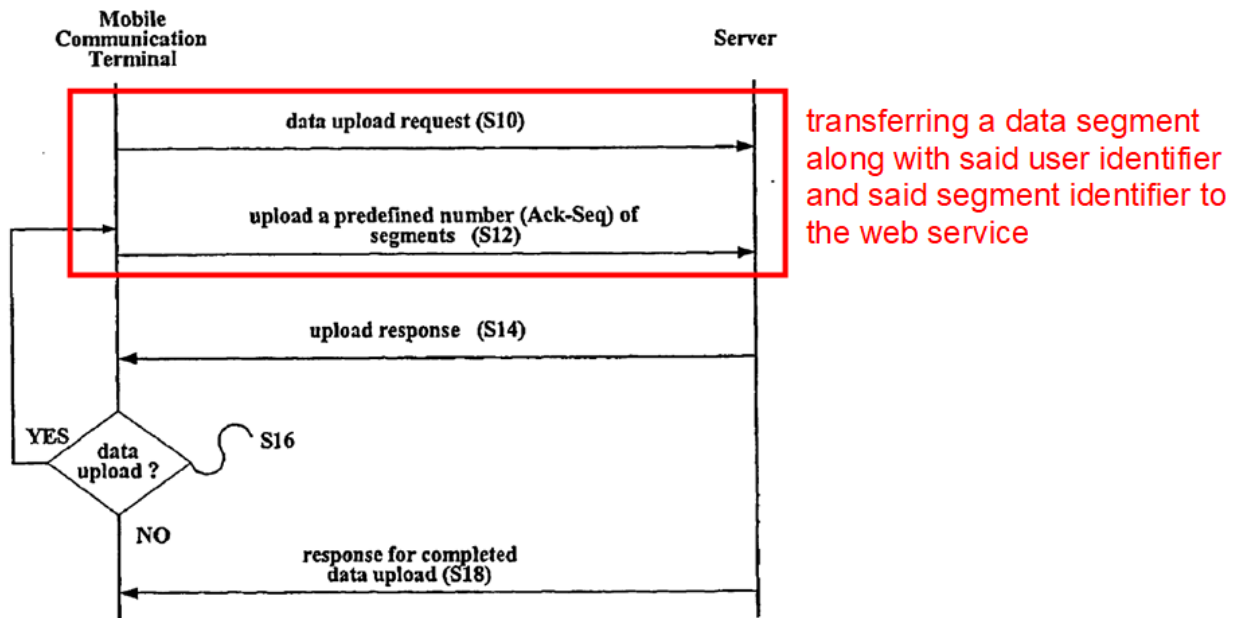
segment identifier”). TTI-1016, ¶¶(0071), (0073), (0058). Also as discussed in 1[b], Takahashi discloses “the mobile phone 1...sequentially transmits the divided image data to a preset transmission destination (image storage server...)” (“*transferring the data segment... to the web service*”) “as an HTTP request.” TTI-1016, ¶¶(0049), (0005)-(0008), (0020)-(0030), (0042)-(0043).

214. Based on Takahashi’s disclosure, a POSITA would have understood or found it obvious that the mobile phone application transmits the divided image data (“*transferring a data segment*”) together with the file name including the user ID (“*along with said user identifier*”) and a division number (“*said segment identifier*”) to the image storage server (“*web service*” per 1[pre]) so the divided image data “can be easily identified [by the server] when the data upload is stopped and resumed afterwards.” TTI-1006, ¶¶[0012]-[0013], [0022]. In this way, Takahashi’s mobile device need not restart upload of the whole media data when the data upload is stopped but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021]; *see above* §VIII.D.

215. A POSITA would have understood that there are multiple ways to transfer divided image data along with the file name including the user ID (“*user identifier*”) and division number “as an HTTP request.” For example, Na discloses generating and sending an upload request message, as shown in FIG. 2, that includes

the terminal number in the form of MIN (“*user identifier*”) and the sequence number of the segment to be uploaded (“*segment identifier*”) for each group of K segments. TTI-1006, ¶¶[0062]-[0063], [0078]. Na further discloses that “the mobile communication terminal 10 **issues an upload request message for each group of a predefined number (e.g., K) segments** of the media data,” and after “**sending an upload request** to the application server 40, the mobile communication terminal 10 **sends K segments** of the media data” to the application server 40. TTI-1006, ¶[0053]. Based on Na’s example of sending each group of K data segments (“*a data segment*”) immediately after sending a request message (“*a data segment*”), where the request message includes a user identifier and a segment identifier, a POSITA would have understood or found obvious that Takahashi-Na teaches transferring divided image data (“*data segment*”) along with a request message including a user ID (“*user identifier*”) and a division number (“*segment identifier*”) to the image storage server (“*web service*”).

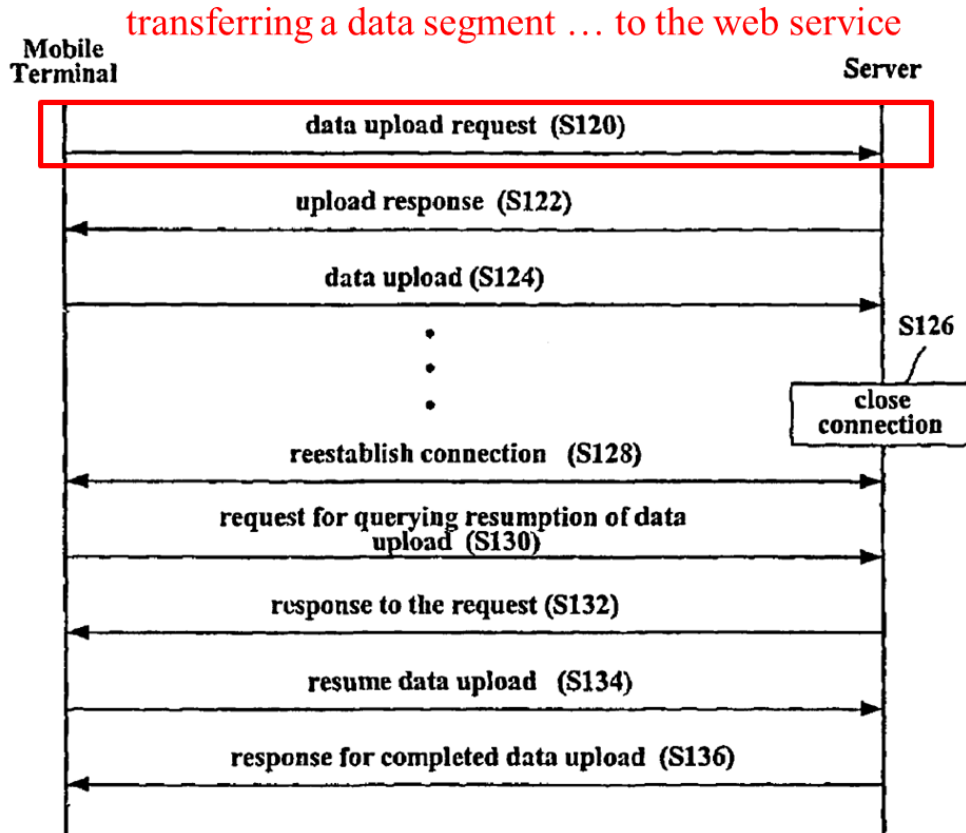
FIG. 10



BD-1006, FIG. 10.

216. As another example, Na discloses, “[w]hen uploading particular media data, the mobile communication terminal 10 **issues a request message** as shown in FIG. 15 comprising ... a message body 204” to the application server 40 (*“transferring a data segment ... to the web service”*). TTI-1006, ¶¶[0116]-[0117], [0112]. Na’s FIG. 25 depicts an example data upload using a request message in “(S120),” where “the mobile communication terminal 10 **sends a request message** for uploading particular media data to the application server 40, **the request message comprising** the command 200, **the header 202**, and the message body 206 as shown in FIG. 15 (S120).” TTI-1006, ¶¶[0148]-[0149].

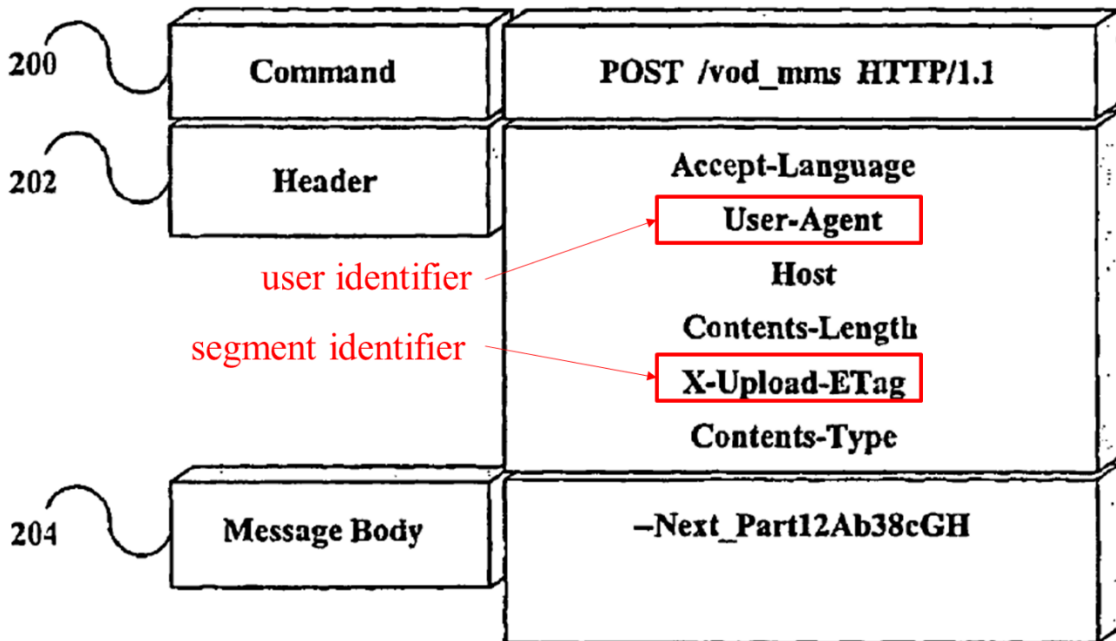
FIG. 25



TTI-1006, FIG. 25.

217. The “header 202” of the “request message as shown in FIG. 15” includes the “user agent information (User-Agent, e.g., SKT/XXX200)” (“*said user identifier*”) and the “unique ID of the message body” (e.g., “the ID of the message body to be uploaded (X-Upload-Etag,...)”) (“*said segment identifier*”). See above 1[d]-1[e]. TTI-1006, ¶¶[0116]-[0117], [0112], [0148]-[0149], FIG. 15.

FIG. 15



TTI-1006, FIG. 15.

218. As taught by this example of Na, a POSITA would have understood or found obvious that Takahashi-Na teaches transferring divided image data (“*data segment*”) as the message body of (“*along with*”) a request message including a user ID (“*user identifier*”) and a division number (“*segment identifier*”) to the image storage server (“*web service*”).

1[f2]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service;

219. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[f2]. For example, as discussed above in 1[d]-1[f1], Takahashi

discloses, at the mobile phone application, the data is divided and each data division is transmitted in a divided file with a division number and a file name including the user ID (“*configuring the data segment*”) so that “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file” (“*so that the data segment can be received by the web service*” and “*so that the data segments can be reassembled using segment identifiers by the web service*”). TTI-1016, ¶¶(0058), (0071), (0073).

220. A POSITA would have understood that there are multiple different ways to configure Takahashi’s division number and user ID for each data segment. For example, Na discloses that the mobile device generates and sends to the server 40 an upload request message, as shown in FIG. 2, that includes the terminal number in the form of MIN (“*user identifier*”) and the sequence number of the segment to be uploaded (“*segment identifier*”) for each group of K segments (“*configuring the data segment so that the data segment can be received by the web service*”). TTI-1006, ¶¶[0062]-[0063], [0078]. Na’s techniques make it possible “for the server to match the previously uploaded data and data to be newly uploaded” and allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred... by providing the sequence number of the segment to be uploaded” (“*so that the data segment can be received by the web service*” and “*so*

that the data segments can be reassembled using segment identifiers by the web service”). TTI-1006, ¶¶[0008], [0010]. A POSITA would have understood or found obvious that Takahashi-Na’s mobile device configures the request message for the media data to include the user identifier and the segment identifier so that the media data can be *reassembled* by the server 40 using the segment identifiers.

221. In another example discussed above in 1[d]-1[f1], Na discloses “[w]hen generating a request message for uploading the particular media data, the mobile communication terminal 10 **assigns a unique ID** of the message body to the header of the message,” (“*configuring the data segment*”), “thereby **allowing the application server 40 to match one portion of the message body that has been already uploaded and the other portion of the message body to be uploaded** in the case where the upload of the media data is resumed” (“*so that the data segment can be received by the web service*” and “*so that the data segments can be reassembled using segment identifiers by the web service*”). TTI-1006, ¶¶[0112], [0012], [0022]. For example, as discussed below in elements 1[k]-1[l], Na discloses that, by configuring the unique ID of the message body, the mobile communication terminal 10 can learn about “the range of the message body that has been uploaded successfully before the upload error or upload cancellation” (e.g., “1-1000/1500,” “where 1500 is the size of the whole data”) and resume the interrupted data upload from the point immediate following such a range (e.g., “1001-1500/1500”) so that

the application server 40 receives and reassembles the multiple portions of the data using the segment identifiers. *See e.g.*, TTI-1006, ¶¶ [0112], [0122], [0132].

222. Additionally, Na discloses that “the mobile communication terminal 10 connects to the application server 40 ... and performs an operation for uploading particular media data stored in a memory thereof to the application server 40 in accordance with the HTTP 1.1 protocol.” TTI-1006, ¶[0052]. More specifically, Na discloses the mobile communication terminal 10 issues request messages in accordance with the example format as shown in FIGS. 2, 15, and 17 (“**configuring the data**”). *See e.g.*, TTI-1006, ¶[0052], [0062], [0116], [0144]. A POSITA would have understood or found obvious that Na’s request messages are configured in such a way in accordance with the HTTP 1.1 protocol **so that the data segment can be received by** the application server 40 (“**the web service**”) to realize the disclosed method of “controlling a media message upload” to the application server. TTI-1006, Abstract. As taught by Na, a POSITA would have understood or found obvious that Takahashi-Na teaches configuring a request message for the data segment to include a user ID (“**user identifier**”) and a division number (“**segment identifier**”) so that the data segment can be received by the web service and reassembled using segment identifiers by the web service.

1[g]. receiving a reply message from the web service, corresponding to successful completion of said data segment transfer;

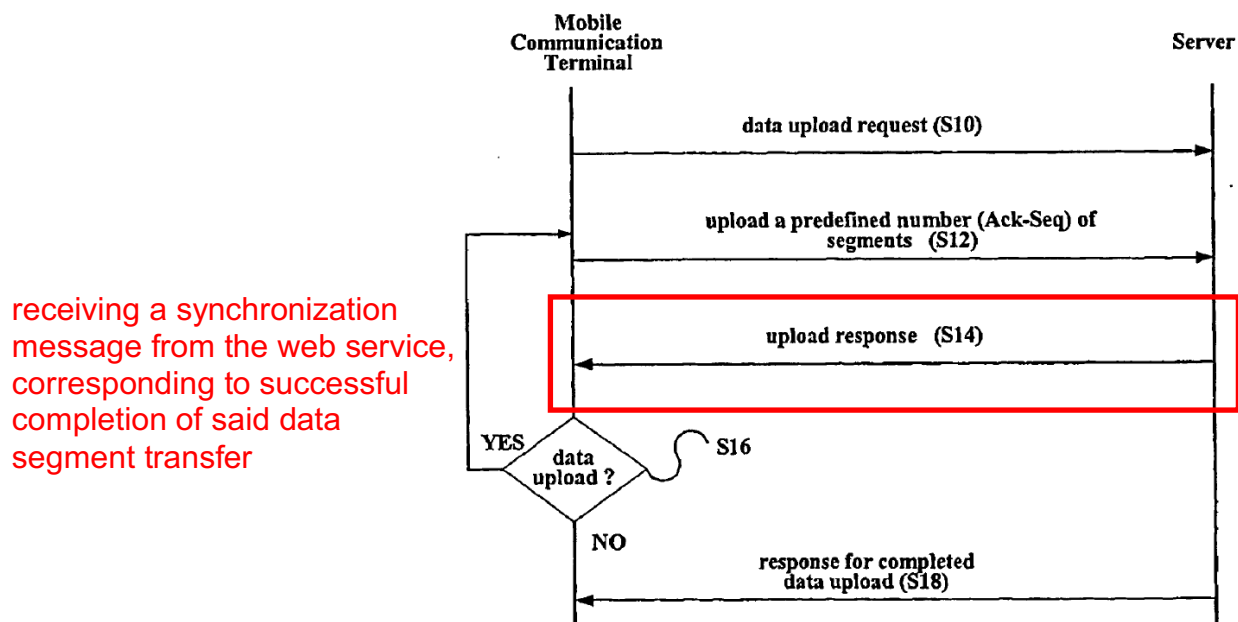
223. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[g]. For example, Takahashi discloses that “the mobile phone 1 adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side” (“***receiving a reply message from the web service, corresponding to successful completion of said data segment transfer***”). TTI-1016, ¶(0058).

224. Additionally, Na discloses that in “the case where a mobile communication terminal uploads particular media data to a server on the wireless network, the mobile communication terminal issues a request message each time it uploads a segment of the media data in accordance with the HTTP 1.1 protocol and **the server issues a response message for each request message**, thereby performing successive uploads of the segments.” TTI-1006, ¶[0004].

225. In a particular example, Na discloses that the application server 40 “issues one **response message** to the mobile communication terminal 10 in response to the upload request **each time it receives K segments** of data from the mobile communication terminal 10.” TTI-1006, ¶¶[0058], [0066] (“Receiving a specified number of data segments, the application server 40 sends **an acknowledgement message** to the mobile communication terminal 10.”). “In other words, after

receiving segments of a specified number (X-Ack-Seq), the application server 40 sends **an acknowledgement response** comprising a status line 110 including a message (e.g., HTTP 200 OK) and a connection status (e.g., Keep-Alive) as shown in FIG. 3 to the mobile communication terminal 10.” TTI-1006, ¶¶[0067], [0080] (“Receiving K segments as specified in the request message, the application server 40 sends **a response message** comprising the status line 110 as shown in FIG. 3 to the mobile communication terminal 10 (S14).”), FIG. 10 (step S14). Indeed, the HTTP 1.1 (rfc 2616) protocol corroborates this understanding and discloses that the status code “OK” indicates “the client’s request was **successfully received**[.]” TTI-1019, 38. A POSITA would have understood or found obvious that Takahashi-Na’s response message corresponds to *a reply message from the web service corresponding to successful completion of a data segment transfer*.

FIG. 10



BD-1006, FIG. 10.

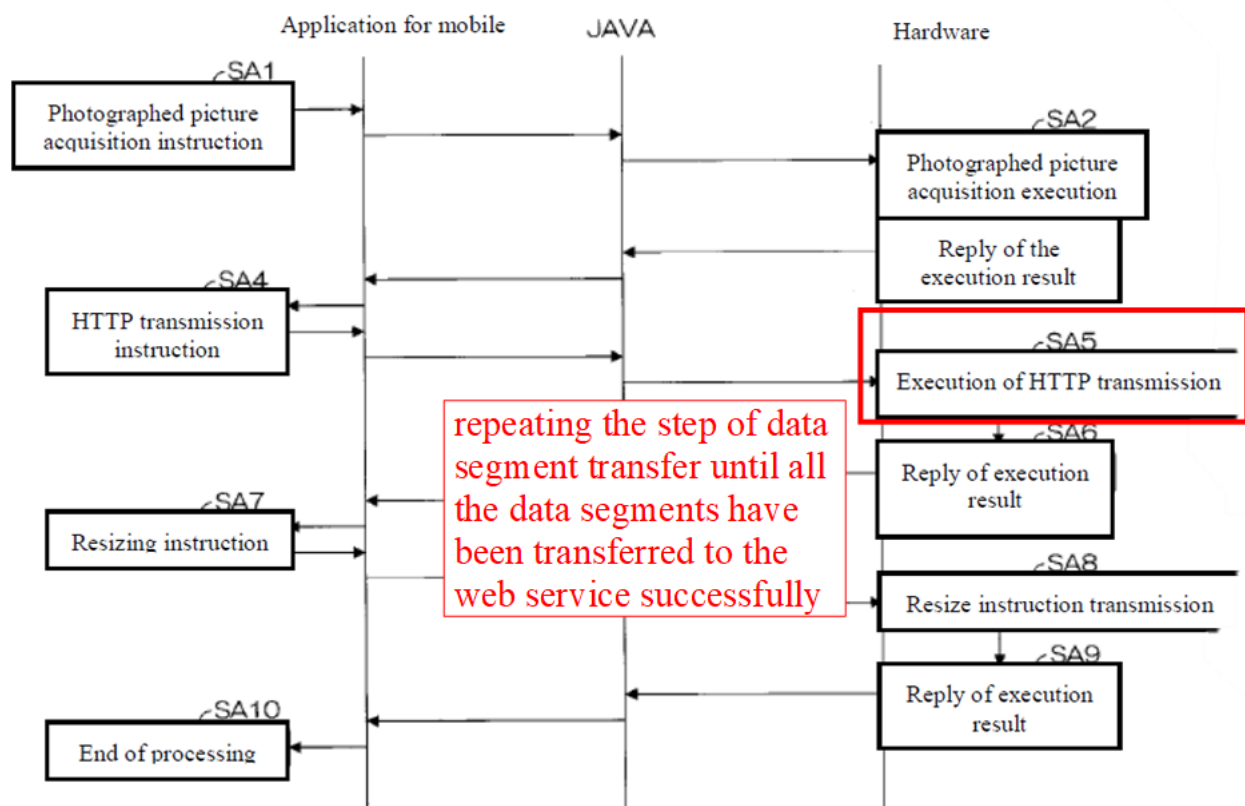
226. Based on Takahashi’s and Na’s disclosures, a POSITA would have understood or found obvious that Takahashi-Na’s response message corresponds to *a synchronization message from the web service corresponding to successful completion of a data segment transfer.*

1[h]. repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully,

227. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[h]. For example, Takahashi discloses that when “image data is divided into several data at this time, step SA5 is repeated, and once the transmission of all the divided image data is completed, the executed result is returned to the application for mobile 1-3” (“*repeating the step of data segment transfer... until*

all the data segments have been transferred to the web service successfully”), as shown in FIG. 6. TTI-1016, ¶(0068). As discussed above in 1[g], Takahashi also discloses that “the mobile phone 1 adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side” (*the step of receiving the synchronization message*”). TTI-1016, ¶(0058).

(FIG. 6)

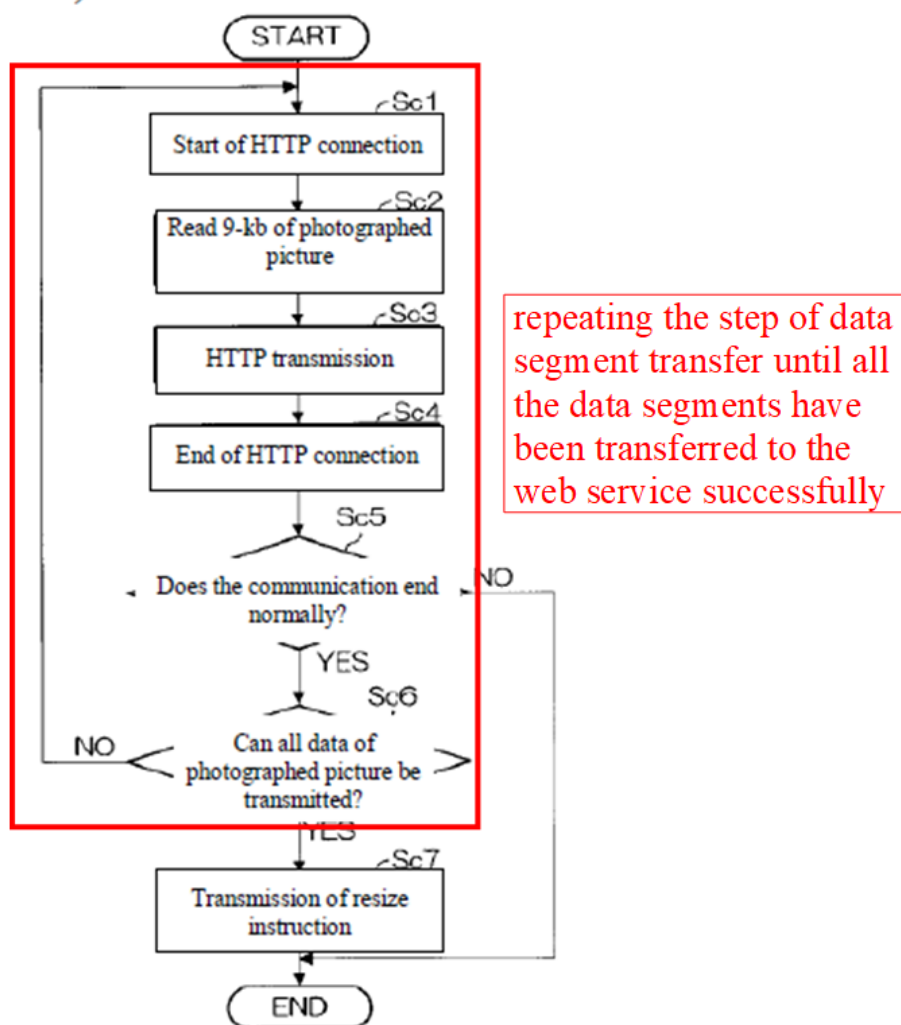


TTI-1016, FIG. 6.

228. Takahashi further explains that “a HTTP connection is initiated (Sc1), and 9-kb of photograph taken is read by the image dividing unit 17 (Sc2). Next,

once the read image data (9 kb) is transmitted to the image storage server 4 through HTTP (Sc3), and the transmission is completed, the HTTP connection is ended (Sc4).” TTI-1016, ¶(0071). If “all the data are not transmitted, the process returns to step Sc1, the processing described above is repeated, and the image data is transmitted 9 kb at a time to the image storage server 4 through HTTP.” TTI-1016, ¶(0072). As shown in FIG. 8, Takahashi repeats *the step of data segment transfer, until all the data segments have been transferred to the web service successfully*. As discussed above in 1[g], Takahashi also discloses that “the mobile phone 1 adopts a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side” (“*the step of receiving the synchronization message*”). TTI-1016, ¶(0058).

(FIG. 8)

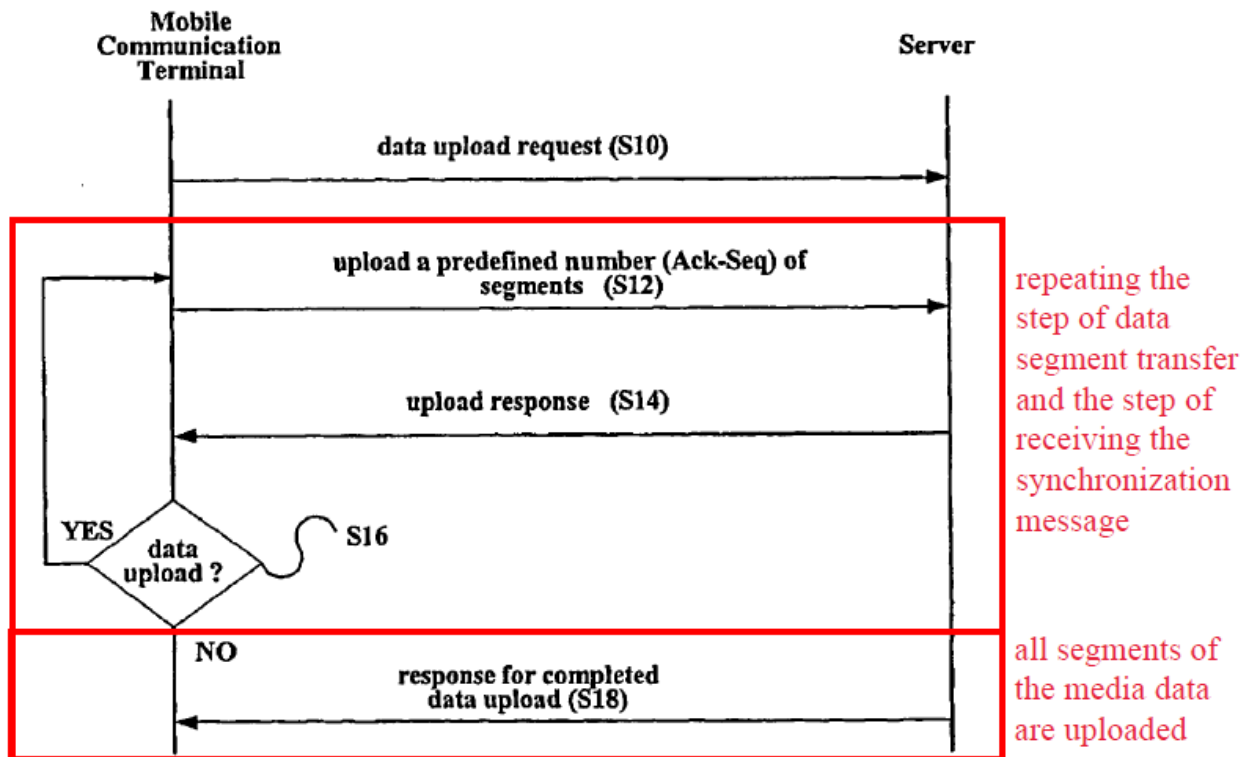


TTI-1016, FIG. 8.

229. Na confirms that transmitting data segments include repeating the step of data segment transfer and the step of receiving the synchronization message, until all the data segments have been transferred to the web service successfully. TTI-1006, ¶[0079]. Specifically, Na discloses that the “mobile communication terminal 10 sequentially uploads as many segments of the media data as is specified by the X-Ack-Seq field included in the request message (S12).” TTI-1006, ¶[0079].

“Receiving K segments as specified in the request message, the application server 40 sends a response message comprising the status line 110 as shown in FIG. 3 to the mobile communication terminal 10 (S14).” TTI-1006, ¶[0080]. “If it is necessary to upload the next K segments, the steps S12 through S16 are repeated until all segments of the media data are uploaded.” TTI-1006, ¶¶[0082]-[0083], [0097] (“If the resumed media data upload is completed normally, the application server 40 sends a response message comprising the status line ... (S52).”). Na’s FIG. 10 illustrates that step S12 for uploading a group of K segments (“*the step of data segment transfer*” per 1[f1]) and S14 for receiving an acknowledgement response message (“*the step of receiving the reply message*” per 1[g]) are repeated until all segments of the media data are uploaded.

FIG. 10



TTI-1006, FIG. 10.

230. Based on Takahashi’s and Na’s disclosures, a POSITA would have understood or found obvious that Takahashi-Na’s mobile phone application transmits the divided image data by repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully.

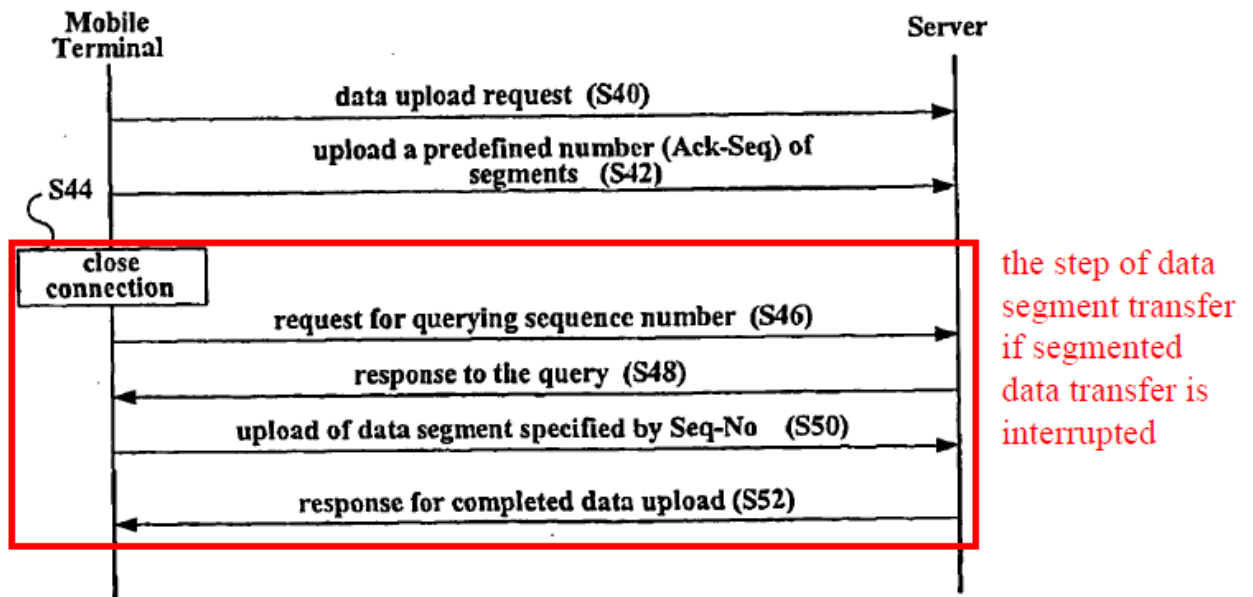
1[i]. wherein if segmented file transfer is interrupted, the software module further performs:

231. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[i]. As discussed above, Takahashi recognizes instances where a “communication is not completed normally.” TTI-1016, ¶(0071); *see above* §VIII.D.

Likewise, a POSITA would have recognized that, in mobile communications such as Takahashi's, "a data upload error caused by various communication problems may occur during an upload of media data," and "it is desirable to resume the upload in such a way that the upload is resumed from the point at which the error occurred." TTI-1006, ¶¶[0006]-[0008]. A POSITA would have been motivated and would have found it obvious and straightforward to implement Takahashi's mobile phone application software (*see* 1[a]) based on Na's teachings to beneficially allow "an upload of media data stopped by an upload error to be resumed from the point where the error occurred," as taught by Na. TTI-1006, ¶¶[0008], [0010].

232. Na discloses various techniques for "resuming an upload of media data" "stopped by an upload error," "when the communication connection is reestablished," or "stopped by a closed connection" ("*segmented file transfer is interrupted*"). TTI-1006, ¶¶[0084], [0091], [0139]-[0157], FIGS. 11-12, 24-25. For example, Na's "FIG. 12 illustrates a flowchart of the procedure for resuming an upload of media data when the communication connection is reestablished" if "the mobile communication terminal 10 gets disconnected from the application server 40 and as a result an upload error occurs (S44)" ("*if segmented file transfer is interrupted*"), which includes steps S46-S50 ("*the step of data segment transfer*") as discussed in more detail below in 1[j]-1[l]. TTI-1006, ¶¶[0036], [0091]-[0097].

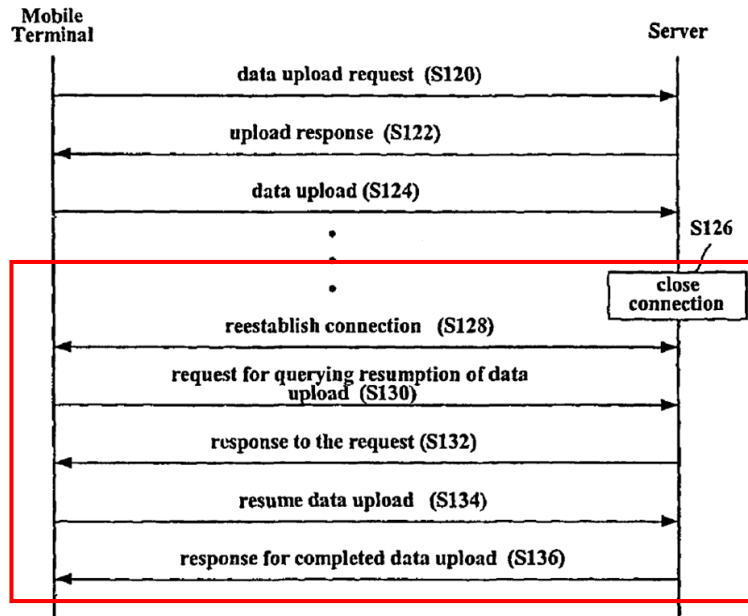
FIG. 12



TTI-1006, FIG. 12.

233. As another example, Na's FIG. 25 "illustrates a flowchart of the procedure for resuming an upload of media data stopped by a user" ("*segmented file transfer is interrupted*"), which includes steps S128-S136 ("*the step of data segment transfer*"), as discussed in more detail below in elements 1[j]-1[l]. TTI-1006, ¶¶ [0049], [0148]-[0157].

FIG. 25



Na discloses techniques for resuming an upload of media data stopped by a closed connection

TTI-1006, FIG. 25.

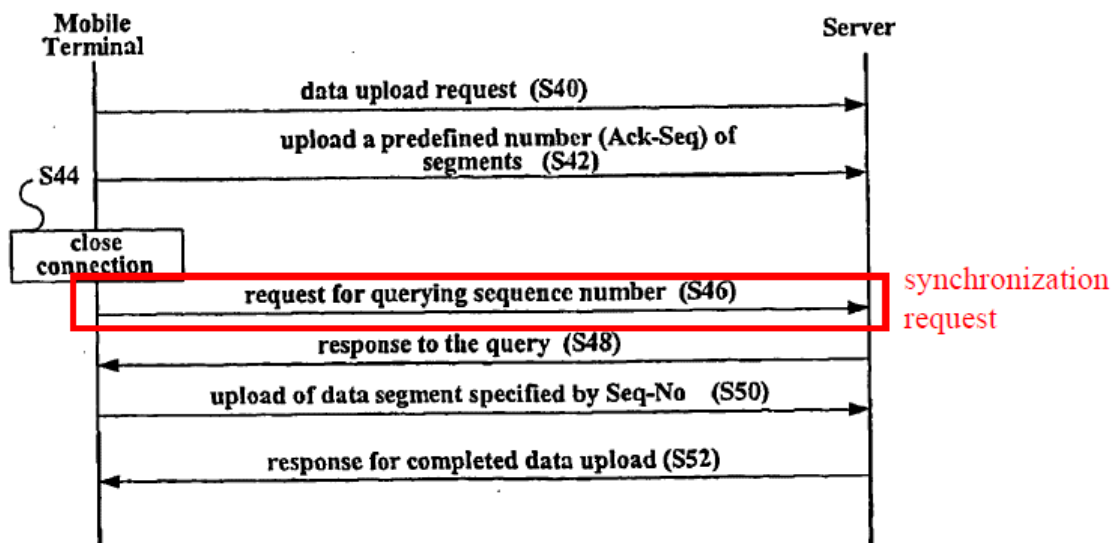
234. As discussed in 1[a], a POSITA would have understood and found obvious that the above operations would have been performed by a software application (“*software module*”) of the mobile phone. TTI-1006, 2:12-16; TTI-1016, ¶¶(0053), (0055)-(0056), FIGS. 2-3.

1[j]. sending, a synchronization request to the web service from the application layer level, corresponding to asking the web service about the segmented file data the web service has successfully received, for said user identifier, after a network connection to the web service is restored;

235. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[j]. For example, Na’s FIG. 12 illustrates a “procedure for resuming an upload of media data when the communication connection is reestablished” (“*after a network connection to the web service is restored*”). TTI-

1006, ¶¶[0036], [0091]. Na's FIG. 12 step S46 shows that "the mobile communication terminal 10 sends a request message querying the sequence number of the segment from which the upload should be resumed to the application terminal 40," (*"a synchronization request to the web service... corresponding to asking the web service about the segmented file data the web service has successfully received"*) "the request message comprising the command 140 and the header 142 including a content type and a **message ID** as shown in FIG. 6." TTI-1006, ¶[0094].

FIG. 12

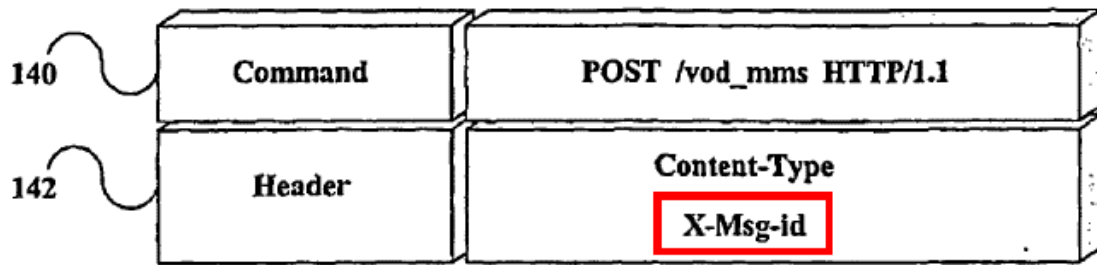


TTI-1006, FIG. 12.

236. As shown in FIG. 6, the message ID in the header 142 is "a message ID assigned to the media data to be uploaded (X-Msg-id, e.g., a **terminal number** and upload time information **in the form of MIN_MMDDH HMMSS**)," and is in a

similar format as Takahashi's file name that includes the user ID discussed above in 1[e], and a POSITA would have understood or found obvious that a terminal number in the form of MIN (mobile identification number) corresponds to the claimed "*user identifier*" that "*uniquely identifies a particular user of the web service.*" TTI-1006, ¶¶[0063], [0078]; TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0056), (0066), (0070)-(0071), (0073), (0084), (0087), FIG. 5; TTI-1020, 4:7-10, 4:17-19, 2:60-63, 9:10-13, 12:46-49; TTI-1021, 7:39-45; TTI-1022, 3:25-27. A POSITA would have recognized or found obvious that the request message would be "*for said user identifier*" because the request message is used to query the data previously uploaded that are associated with the "terminal number" or user ID ("*said user identifier*") to determine "the sequence number of the segment from which the upload should be resumed" for the same user identifier. TTI-1006, ¶[0094]. As such, Takahashi-Na teaches sending, a synchronization request to the web service, for said user identifier, after a network connection to the web service is restored. *Id.*

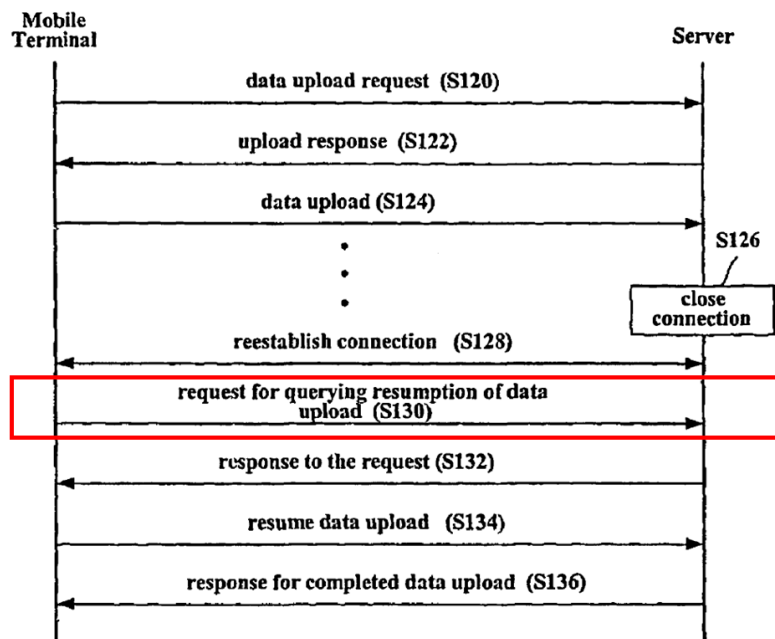
FIG. 6



TTI-1006, FIG. 6.

237. As another example shown in FIG. 25, Na discloses that, after “the connection between them gets reestablished” in S128 (*after a network connection to the web service is restored*), “the mobile communication terminal 10 issues a request message for querying where to resume the upload” to the application server 40 in S130 (“*sending, a synchronization request to the web service*”). TTI-1006, ¶¶[0152]-[0153]. The “request message comprises a header including information **querying whether a portion of the message body that was previously uploaded exists in the application server 40** (“*corresponding to asking the web service about the segmented file data the web service has successfully received*”). TTI-1006, ¶[0153].

FIG. 25

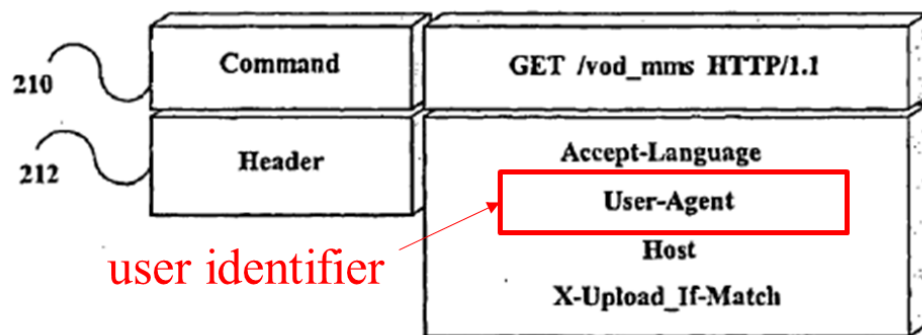


sending, a synchronization request to the web service, after a network connection to the web service is restored

TTI-1006, FIG. 25.

238. The “request message for querying where to resume the upload” is “*for said user identifier*” because, “[a]s shown in FIG. 16, the request message comprises a header including information querying whether a portion of the message body that was previously uploaded exists in the application server 40 (S130)” such as the “user agent information (User-Agent, e.g., SKT/XXX200)” (another example of “*said user identifier*”) as discussed in element 1[e]. TTI-1006, ¶¶[0153], [0116]-[0117], [0119], FIG. 16; TTI-1019, 8, 89.

FIG. 16



TTI-1006, FIG. 16.

239. A POSITA would have recognized or found obvious that the request message would be “*for said user identifier*” because the request message is used to query the data “previously uploaded” that are associated with the “user agent information” (“*said user identifier*”) to determine “where to resume the upload” the data for the same user identifier. TTI-1006, ¶¶[0153], [0116]-[0117], [0119]; *see above* 1[e].

240. Additionally, as discussed in 1[c1]-1[c2], Na discloses performing such an operation “in accordance with ... the **HTTP** 1.1 protocol,” and a POSITA would have understood and found obvious that the synchronization request would be sent “*from the application layer level*” pursuant to the HTTP protocol. TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

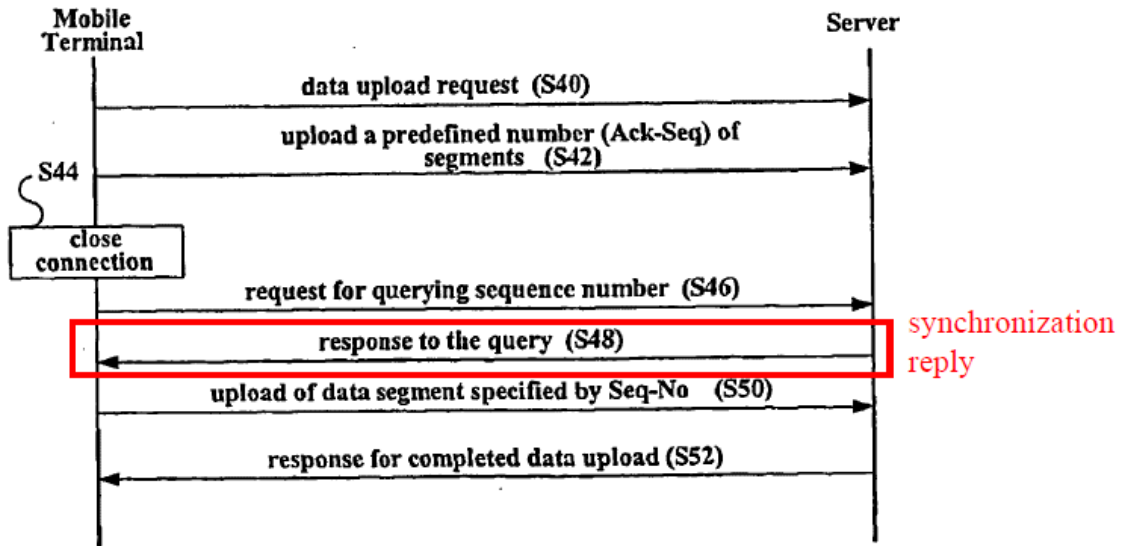
241. A POSITA would have been motivated and would have found it obvious and straightforward to incorporate Na’s teachings in Takahashi’s system to

beneficially allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0008], [0010]. In doing so, a POSITA also would have been motivated and would have found it obvious and straightforward to implement Takahashi’s mobile phone to send to the server a synchronization request querying “the sequence number of the segment from which the upload should be resumed” or “the range of the message body that has been successfully uploaded” so that the mobile phone “is capable of knowing what part of the message body is to be uploaded additionally,” as taught by Na. TTI-1006, ¶¶[0012]-[0013], [0022], [0094]. In this way, Takahashi’s mobile phone need not restart upload of the whole media data but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021].

1[k]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to indicate segmented file data said web service has successfully received, for said user identifier; and

242. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[k]. For example, Na discloses that receiving “the request message querying the sequence number” (“***synchronization request***” per 1[j]), “**the application server 40 issues a response message for the query**” (“***a synchronization reply back from the web service***”), shown as step S48 in Na’s FIG. 12. TTI-1006, ¶[0095].

FIG. 12



TTI-1006, FIG. 12.

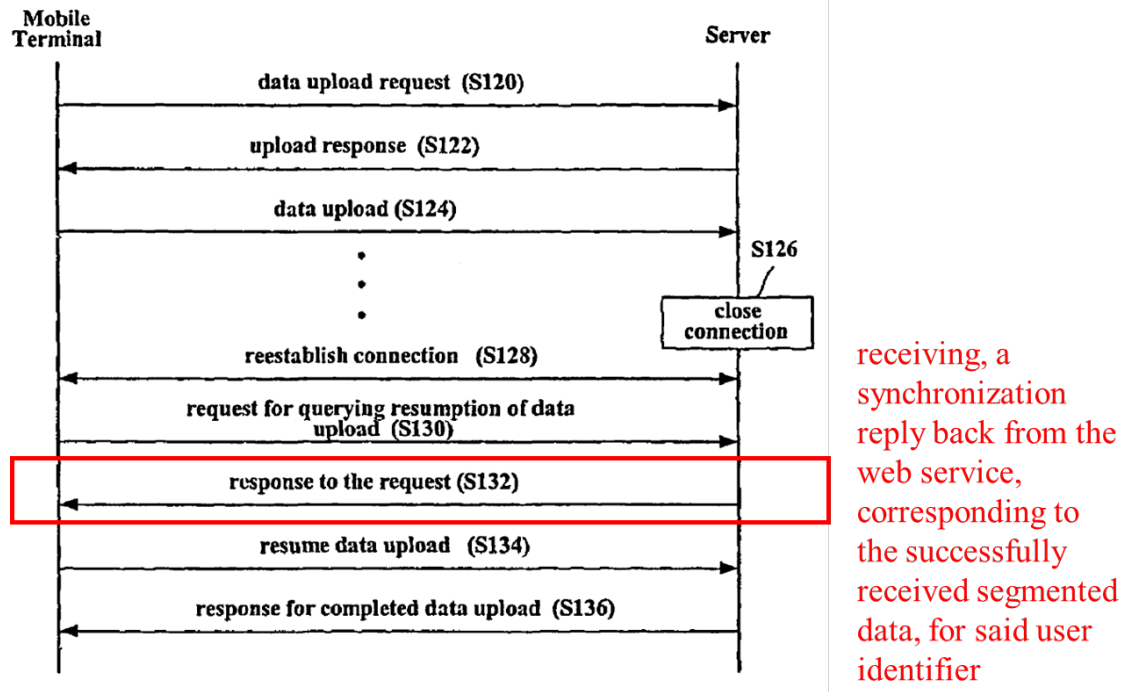
243. Na further discloses that “the response message comprises the status line 150 and the header 152 including a connection status, a cache control, and **the sequence number of the segment from which the upload should be resumed.**” TTI-1006, ¶[0095]. A POSITA would have understood or found obvious that “the sequence number of the segment from which the upload should be resumed” directly indicates (e.g., the last successfully received segmented data) or indirectly indicates (e.g., the next data segment following the last successfully received segmented data) and thus corresponds to the successfully received segmented file data.

244. Further, a POSITA would have recognized or found obvious that the response message would be “*for said user identifier,*” for example, because (1) the response message is directly in response to the request message that is “*for said user*

identifier” as discussed in element 1[j]; and (2) the response message itself identifies “the sequence number of the segment from which the upload should be resumed” that are associated with the “terminal number” or user ID (“*said user identifier*”) so as to inform the mobile communication terminal 10 “the sequence number of the segment from which the upload should be resumed” for the user identifier. *Id.*

245. As another example, Na discloses that the mobile communication terminal 10 receives “a response message ... as shown in FIG. 21 or FIG. 22 (S132)” from the application server 40 (“*receiving, a synchronization reply back from the web service*”). TTI-1006, ¶[0154]. The response message includes “a message header including the range of data that has been uploaded already,” such as “the range of the **content that has been uploaded successfully** before the upload error (X-Upload-Range, e.g., 1-1000/1500)” (“*corresponding to the indicated segmented file data said web service has successfully received*”). TTI-1006, ¶[0154], [0132], [0134], FIGS. 21-22, 25.

FIG. 25



TTI-1006, FIG. 25.

246. A POSITA would have recognized or found obvious that the response message would be “*for said user identifier*,” for example, because (1) the response message is directly in response to the request message that is “*for said user identifier*” as discussed in element 1[j]; and (2) the response message itself identifies “the range of data that has been uploaded already” that are associated with the “user agent information” (“*said user identifier*”) so as to inform the mobile communication terminal 10 “where to resume the upload” the data for the user identifier. TTI-1006, ¶¶[0153]-[0155],[0119]; *see above* 1[e], [1j].

247. Additionally, as discussed in 1[c1]-1[c2], Na discloses performing such an operation “in accordance with ... the **HTTP** 1.1 protocol,” and a POSITA would have understood and found obvious that the synchronization reply would be sent “***at the application layer level***” pursuant to the HTTP protocol. TTI-1006, ¶¶[0003]-[0004]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

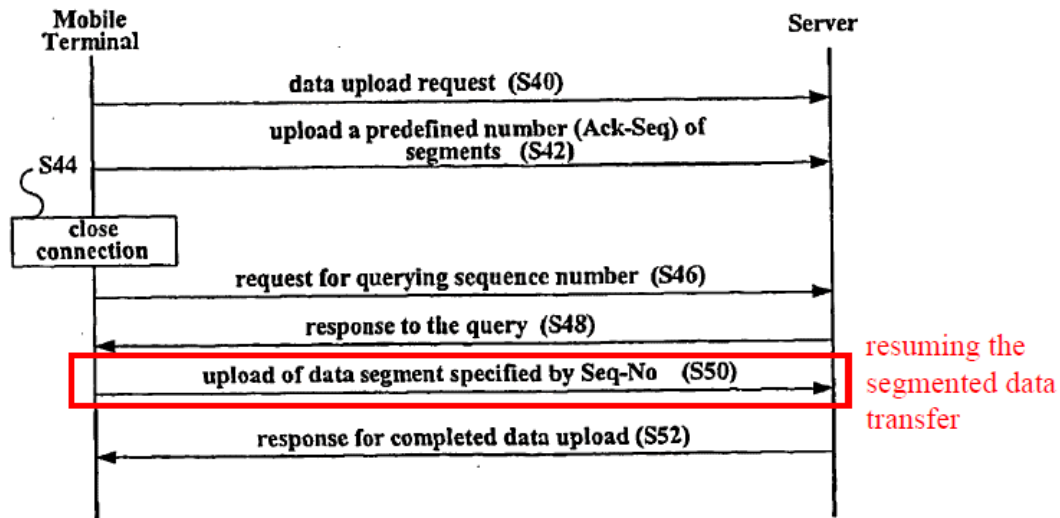
248. A POSITA would have been motivated and would have found it obvious and straightforward to implement Takahashi’s mobile phone to receive from the server a synchronization reply corresponding to “the sequence number of the segment from which the upload should be resumed” or “the range of the message body that has been successfully uploaded” so that the mobile phone “is capable of knowing what part of the message body is to be uploaded additionally,” as taught by Na. TTI-1006, ¶¶[0012]-[0013], [0022], [0094]; *see above* §VIII.D, 1[j].

1[l]. resuming, the segmented file transfer based on the synchronization reply received, for said user identifier.

249. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 1[l]. For example, Na discloses that the “mobile communication terminal 10 **resumes the upload with reference to** the sequence number included in **the response message** from the application server 40” (“***resuming, the segmented file transfer based on the synchronization reply received***”), as shown in step S50 of FIG. 12. TTI-1006, ¶[0096]. A POSITA would have understood or found obvious that the terminal 10 resumes the upload from “the sequence number

included in the response message from the application server 40” in order for the resumed media data upload to be completed normally. TTI-1006, ¶[0097]. Further, a POSITA would have recognized or found obvious that the mobile communication terminal 10 resuming the upload would be “*for said user identifier*,” for example, because the upload is associated with said user identifier, and the process described with respect to FIG. 12 is for resuming the upload of data for the user identifier. *Id.*

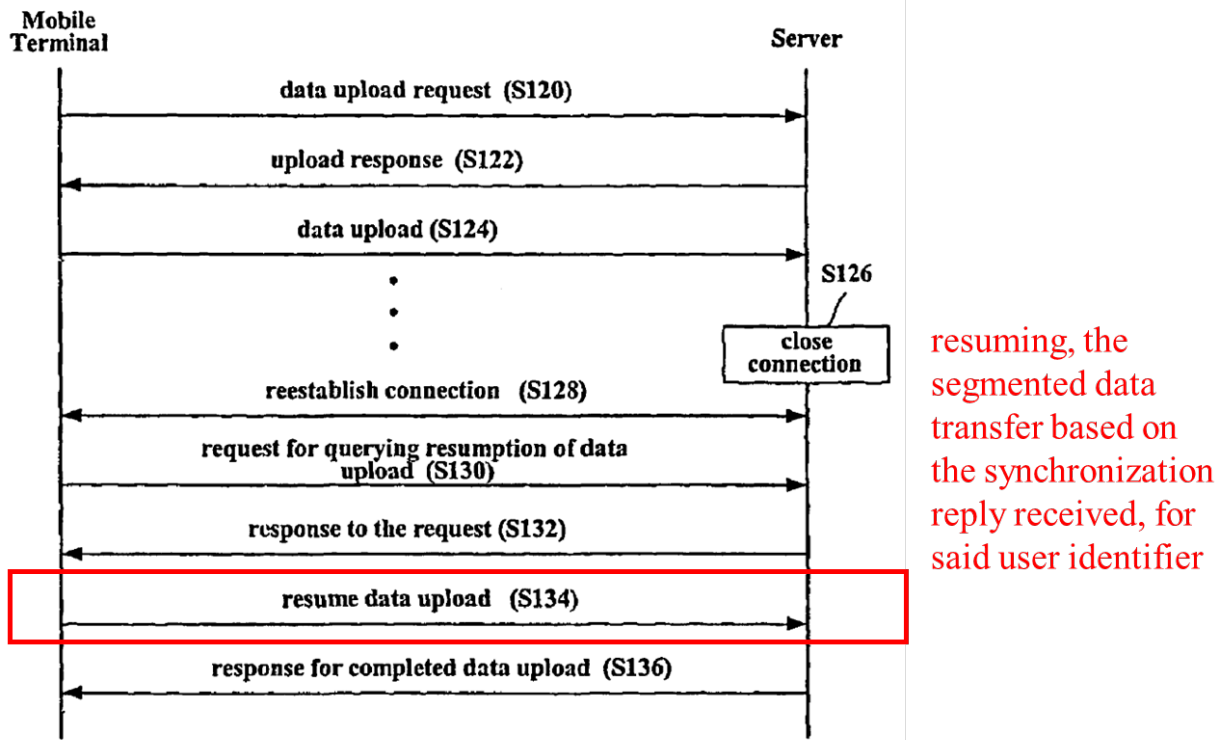
FIG. 12



TTI-1006, FIG. 12.

250. As another example, Na discloses that the mobile communication terminal 10 “begins uploading a portion of the media data that needs to be uploaded additionally (S134)” (“*resuming, the segmented file transfer*”) in response to “receiv[ing] the response message from the application server 40 (“*based on the synchronization reply received*”), as shown in FIG. 25. TTI-1006, ¶[0156].

FIG. 25



TTI-1006, FIG. 25.

251. A POSITA would have recognized or found obvious that the mobile communication terminal 10 resuming the upload would be “*for said user identifier*,” for example, because the upload is associated with said user identifier, and the process described with respect to FIG. 25 is for resuming the very upload of data for the user identifier. TTI-1006, ¶¶[0148], [0153]-[0155], [0119]; *see above* 1[e], [1j], 1[k].

252. As discussed above, a POSITA would have been motivated and would have found it obvious and straightforward to incorporate Na’s teachings into Takahashi’s system to beneficially allow “an upload of media data stopped by an

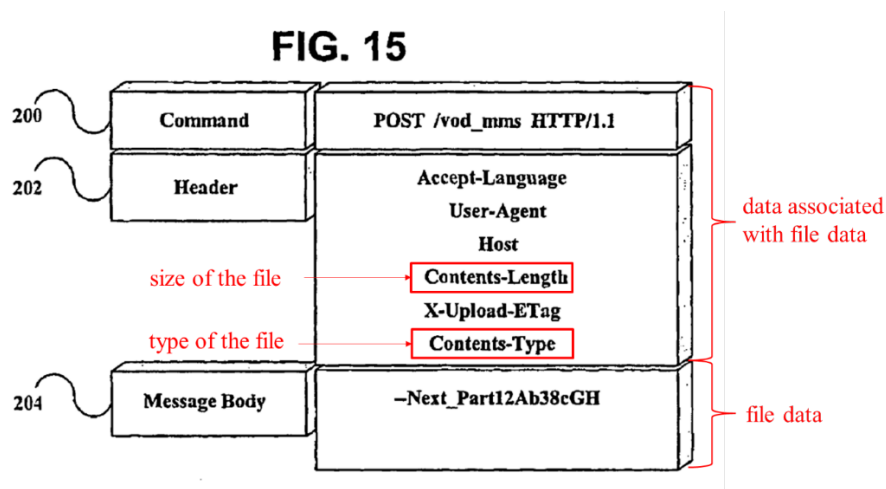
upload error to be resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0008], [0010]; *see above* §VIII.D. In this way, Takahashi’s mobile phone need not restart upload of the whole media data but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021].

2. The method of claim 1, wherein the data associated with file data is at least one of name of the file, size of the file, and type of file.

253. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 2. For example, as discussed above in 1[c3], Takahashi discloses that, at the mobile phone, the image data is divided and each data division (“**file data**”) is transmitted in a divided file with a file name (“**data associated with said file data is... name of the file**”) so that “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file.” TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070)-(0071), (0073), FIG. 5.

254. Additionally, Na discloses that the “header 102” (“**data associated with said file data**”) includes at least “content type information (Contents-Type, e.g., application/vnd.skt.message)” (“**type of file**”) and “the number of total segments in the media data (X-Total-Seq, e.g., 10)” (“**size of the file**”). TTI-1006, ¶[0063]; *see above* 1[c3].

255. As another example described in connection with FIG. 15, Na discloses that the “header 202” (“*data associated with said file data*”) includes “content length given in bytes (Contents-Length, e.g., 1500)” (“*size of the file*”) and content type information (Contents-Type, e.g., multipart/mixed;boundary=Next_Part12Ab38cGH)” (“*type of file*”). TTI-1006, ¶¶[0116]-[0117].



TTI-1006, FIG. 15.

256. A POSITA would have been motivated to implement Takahashi’s mobile phone to transfer image data and data (e.g., name of the file, size of the file, and type of file) associated with the image data, as suggested by Na, to facilitate the reception, storage, and publication of the image data at the server.

3. The method of claim 1, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

257. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 3. For example, Takahashi discloses transfer of “**image data**” (“*said*

file data comprises... image data”) by the mobile phone. See e.g., TTI-1016, Abstract, ¶(0005). Takahashi further discloses that the image data can be captured with a **digital** camera. TTI-1016, ¶¶(0002)-(0003). When describing image size and data size, Takahashi uses the unit “KB”/“kb” which stands for kilobytes. TTI-1016, ¶¶(0003), (0051), (0057), (0071)-(0073), (0080).

258. Additionally, Na discloses that uploaded data (“*file data*”) is “media data,” which in one example can be transmitted in a message body of a request message, with a “content length given in **bytes** (Contents-Length, e.g., **500**),” and can be “defined within 64 **bytes** in accordance with the message body format of the MMS 1.0 specification.” See e.g., TTI-1006, ¶¶[0001]-[0002], [0062]-[0063], [0116]-[0118], [0123].

259. A POSITA would have understood and found obvious that Takahashi-Na would have transferred “*digital data*” at least because it is common knowledge that “bytes” is a hallmark metric of “digital data,” and Takahashi and Na each discloses the media data size is measured in “bytes.”

4. The method of claim 1, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

260. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 4. For example, as discussed above in 1[e], Takahashi discloses “a user ID specific to a user from the server after logging-in” (“*user identifier*”). TTI-

1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070), FIG. 5. Takahashi further discloses that “the user ID 30 is stored in association with a device number specific to the mobile phone 1” (“*user-device-information*”) and “the image storage server 4 searches a user ID based on the device number (Sa3).” TTI-1016, ¶¶(0064), (0066). A POSITA would have understood or found it obvious that Takahashi’s user ID is either a “*user-name*” used to log in to the server or “*user-device-information*” stored in the image storage server.

261. Moreover, Na discloses that the header 102 includes “a message ID assigned to the media data to be uploaded (X-Msg-id, e.g., **a terminal number** and upload time information **in the form of MIN_MMDDH HMMSS**).” TTI-1006, ¶¶[0063], [0078]. A POSITA would have understood or found it obvious that a terminal number in the form of MIN (mobile identification number) (another example of “*user identifier*”) corresponds to “*user-device-information*.” TTI-1020, 4:7-10, 4:17-19, 2:60-63, 9:10-13, 12:46-49; TTI-1021, 7:39-45; TTI-1022, 3:25-27.

262. As another example discussed above in 1[e], Na discloses “**user agent information (User-Agent, e.g., SKT/XXX200)**,” (another example of “*a user identifier*”). TTI-1006, ¶¶[0116]-[0117], FIG. 15. A POSITA would have understood and found obvious that the “**user agent information (User-Agent, e.g., SKT/XXX200)**” “*comprises one or more elements from user information, and wherein the one or more elements of the user information comprise ... user-*

device-information” because, per “the HTTP 1.1 protocol” used by Na, “[t]he **User-Agent** request-header field **contains information about the user agent** originating the request,” where the “**user agent**” is “[t]he client which initiates a request. These are often **browsers, editors, spiders (web-traversing robots), or other end user tools,**” which is “*user device information.*” TTI-1019, 8, 89.

5. The method of claim 1, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

263. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 5. For example, as discussed above in 1[c2], a POSITA would have recognized or found obvious that the image data is divided (“*segmented*”) due in part to “a service limitation posed by a communication provider” to the mobile terminal (“*data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection*”) because Takahashi discloses that the image data is transmitted to the server via “a base station” and “the Internet” and that the image data is divided (“*segmented*”) “**in consideration of** such circumstances” where the data transfer capability of the mobile terminal is limited “by a communication provider.” TTI-1016, ¶¶(0003)-(0008), (0051), (0080). A POSITA would have understood or found obvious that the “base station is a part of a cellular network.” TTI-1005, ¶[0032]. Indeed, Takahashi discloses because of the

consideration of the limited data transfer capability of the mobile terminal, “the image data is transmitted after being divided,” and thus “high definition images can be transmitted and received without being influenced by **any restriction (10 KB or less) in terms of the service provided by a communication provider**” (“*data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection*”) and advantageously allow “[l]arge-capacity image data with high image quality” to be transmitted to a server via a network by an HTTP request “without being influenced by a service limitation posed by a communication provider.” TTI-1016, ¶¶(0004), (0051), (0080).

6. The method of claim 1, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

264. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 6. For example, as discussed above in 1[c1], Takahashi-Na uses HTTP, which a POSITA would have understood and found obvious is “***the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite***” because it was well known in the art before the Critical Date that the “***application layer***” was “***layer 7 in the OSI layer stack,***” TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049); TTI-1006, ¶[0052]; TTI-1015, 2:58-4:40 (describing the “OSI model”). Moreover, Gourley corroborates this understanding

and discloses that HTTP is the “application layer,” layered over TCP, “the popular Internet transport protocol” in the Internet protocol suite as shown in Figure 1-9. TTI-1007, 11-13, 76.

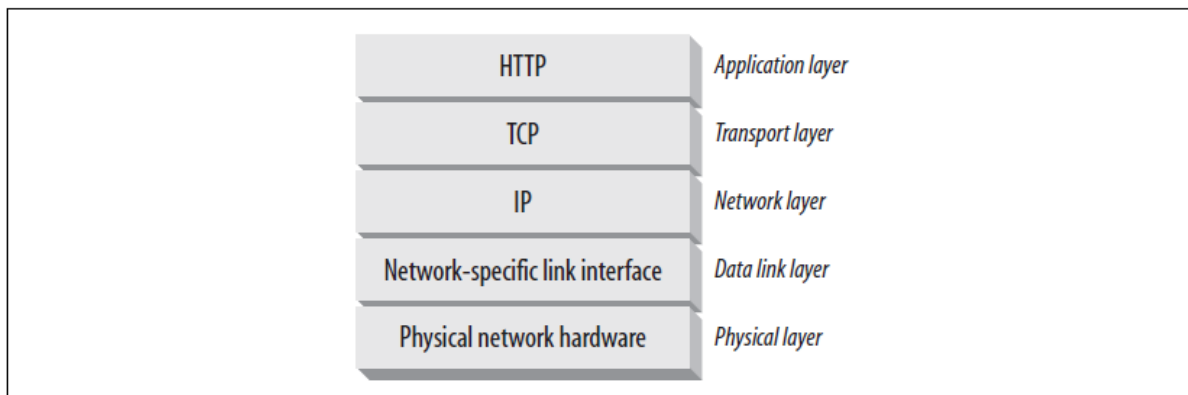


Figure 1-9. HTTP network protocol stack

TTI-1007, Figure 1-9.

7. The method of claim 1, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment.

265. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 7. For example, as discussed above in 1[d], Takahashi discloses “[w]hen carrying out division and transmission of the image file from the mobile phone 1, the information showing the order of the file such as division number (Ex. 5/10, or the like)” (“**the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment**”) “may be attached and sent, if needed” “in the HPPT [sic] transmission” that includes a division of the data (“**file data that is being transferred in a given data segment**”). TTI-1016, ¶(0058). A POSITA would have understood or found obvious that the division number

corresponds to the content range of the file data because it is “information showing the order of the file.” *Id.*

266. Further, with respect to Na’s sequence number discussed above in 1[d], a POSITA would have understood or found obvious that “the **sequence number** of the segment to be uploaded” (“*segment identifier*”) for each group of K segments *corresponds to the content range of said file data* (e.g., the group of K segments) *that is being transferred in a given data segment* because Na discloses “by providing the sequence number of the segment to be uploaded” (“*segment identifier*”), Na’s techniques make it possible “for the server to match the previously uploaded data and data to be newly uploaded” and allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred.” TTI-1006, ¶¶[0053], [0063], [0078], [0008], [0010].

267. As another example, a POSITA would also have understood or found obvious that Na’s “unique ID” (“*segment identifier*”) of a message body of a request message” indicates *the content range of said file data that is being transferred in a given data segment* (e.g., the message body in the request message) because Na discloses “the unique ID of the message body” (“*segment identifier*”) “allow[s] the application server 40 to match one portion of the message body that has been already uploaded and the other portion of the message body to be uploaded in the case where the upload of the media data is resumed” and allows “the mobile

communication terminal 10 [to examine] whether a portion of the media data that has been already uploaded exists in the application server 40 using and resumes the upload if available.” TTI-1006, ¶¶[0112]-[0113].

268. Moreover, as discussed in elements 1[k]-1[l], Na discloses that, by configuring the unique ID of the message body (“*segment identifier*”), the mobile communication terminal 10 can learn about “the **range of the message body that has been uploaded successfully** before the upload error or upload cancellation” (e.g., “1-1000/1500,” “where 1500 is the size of the whole data,”) (“*the content range of said file data that is being transferred in a given data segment*”) and resume the interrupted data upload from the point immediate following such a range (e.g., “1001-1500/1500”). *See e.g.*, TTI-1006, ¶¶ [0112], [0122], [0132].

269. A POSITA would have understood or found obvious that each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile devices generates and sends an upload request message that includes the sequence/division number or unique ID (“*segment identifier corresponds to the content range*”) of the divided image data to be uploaded (“*file data that is being transferred in a given data segment*”), so that Takahashi’s mobile device need not restart upload of the whole media data when the data upload is stopped but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021]; *see above* §VIII.D.

8[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprises:

270. To the extent the preamble is limiting, both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[pre] for similar reasons as discussed above for 1[pre].

8[a]. determining size of file to be transferred by the Bluetooth enabled mobile device;

271. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[a] in multiple ways for similar reasons as discussed above for 1[b].

8[b]. transferring said file by the Bluetooth enabled mobile device to the web service, wherein transferring said file comprises, the Bluetooth enabled mobile device, performing:

272. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[b]. For example, as discussed above in 1[pre], in Takahashi-Na and Takahashi-Na-Lind, a mobile phone (“*Bluetooth enabled mobile device*” per 1[pre]) transfers “image data” (“*transfer[s] said file*”) to “an image storage server” (“*the web service*”). TTI-1016, Abstract, ¶¶(0005)-(0008), (0020)-(0030), (0042)-(0043), (0049), FIG. 1; *see above* 1[pre].

8[c1]. segmenting said file into data segments at an application layer level;

273. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[c1] for similar reasons as discussed above for 1[c1].

8[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

274. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[c2] for similar reasons as discussed above for 1[c2].

8[c3]. wherein data in said data segments comprises one or more of said file data and data associated with said file data;

275. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[c3] for similar reasons as discussed above for 1[c3].

8[d]. applying a segment identifier to each of said data segments at the application layer level;

276. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[d] for similar reasons as discussed above for 1[d].

8[e]. applying a user identifier to each of said data segments for the destination web service at the application layer level, wherein said user identifier uniquely identifies a particular user of the web service;

277. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[e] for similar reasons as discussed above for 1[e].

8[f1]. transferring a data segment along with said user identifier and said segment identifier to the web service,

278. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[f1] for similar reasons as discussed above for 1[f1].

8[f2]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service;

279. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[f2] for similar reasons as discussed above for 1[f2].

8[g]. receiving a reply message from the web service, corresponding to successful completion of said data segment transfer;

280. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[g] for similar reasons as discussed above for 1[g].

8[h]. repeating the step of data segment transfer and the step of receiving the reply message, until all the data segments have been transferred to the web service successfully,

281. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[h] for similar reasons as discussed above for 1[h].

8[i]. wherein if segmented file transfer is interrupted, the Bluetooth enabled mobile device further performs:

282. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[i] for similar reasons as discussed above for 1[i].

8[j]. sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;

283. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[j] for similar reasons as discussed above for 1[j].

8[k]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said file by said web service, for said user identifier; and

284. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[k] for similar reasons as discussed above for 1[k].

8[l]. resuming, the segmented file transfer by transferring remaining data of the file based on the synchronization reply received, for said user identifier.

285. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 8[l] for similar reasons as discussed above for 1[l].

9. The method of claim 8, wherein the data associated with file data is at least one of name of the file, size of the file, and type of file.

286. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 9 for similar reasons as discussed above for claim 2.

10. The method of claim 8, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

287. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 10 for similar reasons as discussed above for claim 3.

11. The method of claim 8, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

288. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 11 for similar reasons as discussed above for claim 4.

12. The method of claim 8, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

289. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 12 for similar reasons as discussed above for claim 5.

13. The method of claim 8, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

290. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 13 for similar reasons as discussed above for claim 6.

14. The method of claim 8, wherein the segment identifier corresponds to the content range of said file data that is being transferred in a given data segment.

291. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 14 for similar reasons as discussed above for claim 7.

15[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:

292. To the extent the preamble is limiting, both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[pre] for similar reasons as discussed above for 1[pre].

15[a]. providing a software module on the Bluetooth enabled mobile device, the software module, performing;

293. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[a] for similar reasons as discussed above for 1[a].

15[b]. determining size of a file to be transferred;

294. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[b] for similar reasons as discussed above for 1[b].

15[c1]. segmenting said file into a first data segment and a second data segment at an application layer level;

295. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[c1]. For example, as discussed above in 1[c1], based on Takahashi's disclosure that "image data is divided into several data at this time, step SA5 is repeated" until "the transmission of all the divided image data is completed," and "a sequential transmission rule for transmitting a divided file one at a time and transmitting the next divided file after waiting for the response of the reception completion on the image storage server 4 side," a POSITA would have recognized or found it obvious that Takahashi's mobile phone application would have segmented the media data into multiple divided files ("***segmenting said file into a first data segment and a second data segment***") in order to transmit "all the divided image data." TTI-1016, ¶¶(0068), (0058).

296. As discussed in 1[c1], a POSITA would have understood that there were multiple different ways to format each "divided file" ("***data segment***") "as an HTTP request." As one example, based on Na's disclosure of uploading multiple groups of K segments (including "***a first data segment and a second data segment***") "until all segments of the media data are uploaded," a POSITA would have

recognized or found it obvious that each of Takahashi-Na's and Takahashi-Na-Lind's Bluetooth enabled mobile devices would send an HTTP upload request message to the server for each "divided image data" ("*data segment*"). TTI-1006, ¶¶[0052]-[0053], [0080]-[0082].

297. As another example, based on Na's disclosure of "one portion of the message body that has been already uploaded" ("*a first data segment*") "and the other portion of the message body to be uploaded" ("*a second data segment*"), a POSITA would have understood and found obvious that each of Takahashi-Na's and Takahashi-Na-Lind's Bluetooth enabled mobile devices would have segmented image data into divided image data ("*data segments*") to upload each divided image data in respective "message bodies" of different "request messages." TTI-1006, ¶¶[0112]-[0113], [0039], [0116]-[0118], [0151].

298. Additionally, as discussed in 1[c1], a POSITA would have understood and found obvious that the image data ("*file*") would be segmented "*at an application layer level*" to be uploaded along with or included in multiple "request messages" pursuant to the HTTP protocol. TTI-1006, ¶¶[0003]-[0004], [0052], [0111]; TTI-1019, 1; TTI-1007, 7, 76, 249-251; TTI-1002, 115.

15[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

299. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[c2] for similar reasons as discussed above for 1[c2].

15[d]. wherein the second data segment comprises file data and the first data segment comprises data associated with file data, and wherein the data associated with file data is at least one of name of the file, size of the file, and type of the file;

300. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[d]. For example, as discussed above in claim 2, Takahashi discloses that, at the mobile phone, the image data is divided and each data division (including “*the first data segment*” and “*the second data segment*”) is transmitted in a divided file (at least “*the second data segment comprises file data*”) with a file name (at least “*the first data segment comprises data associated with file data... is... name of the file*”) so that “when the divided file (image data) with the same file name has already been received [by the image storage server], the divided file is written as an addition (combination) to the existing file.” TTI-1016, ¶¶(0010), (0013), (0032), (0035), (0066), (0070)-(0071), (0073), FIG. 5.

301. Additionally, as discussed in claim 2, Na discloses that the “header 102” (“*data associated with said file data*”) includes at least “content type information (Contents-Type, e.g., application/vnd.skt.message)” (“*type of file*”) and “the number of total segments in the media data (X-Total-Seq, e.g., 10)” (“*size of the file*”). TTI-1006, ¶[0063]. Na further discloses that the “content type information” (“*type of file*”) and “the number of total segments” (“*size of the file*”) is included in the header 102 of at least the *first data segment*. TTI-1006, ¶¶[0063]-[0064] (“when the upload is initiated” or “in every request message”).

302. As also discussed above in 1[c1] and 1[f1], Na further discloses that “the mobile communication terminal 10 issues an upload request message for each group of a predefined number (e.g., K) segments of the media data,” and along with “sending an upload request to the application server 40, the mobile communication terminal 10 sends K segments of the media data” to the application server 40. TTI-1006, ¶[0053]. Accordingly, Na discloses multiple groups of K segments (“*second data segment[s]*”) that each includes media data (“*file data*”).

303. As another example discussed above in 1[c1] and 1[f1], Na discloses uploading multiple parts/portions of “particular media data” by sending multiple request messages (“*first data segment*” and “*second data segment*”) that each include a “header 202” (“*data associated with said file data*”) and a “message body 204” (“*file data*”). TTI-1006, ¶[0112]. As discussed in claim 2, the “header 202” includes “content length given in bytes (Contents-Length, e.g., 1500)” (“*size of the file*”) and content type information (Contents-Type, e.g., multipart/mixed;boundary=Next_Part12Ab38cGH)” (“*type of file*”). TTI-1006, ¶¶[0116]-[0117]; *see above* 1[c3]. The message body 204 includes a “*data segment*”: “-Next_Part12Ab38cGH” (“*file data*”). TTI-1006, ¶¶[0039], [0116]-[0118]. A POSITA would have been motivated to implement Takahashi’s mobile phone to transfer image data in at least the second data segment and data (e.g., name of the file, size of the file, and type of file) associated with the image data in at least the

first data segment, as suggested by Na, to facilitate the reception, storage, and publication of the image data at the server.

15[e1]. applying a user identifier to said first data segment at the application layer level, wherein the user identifier uniquely identifies a particular user of the web service;

304. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[e1] for similar reasons as discussed above for 1[c1], 1[e], and 15[c1]. For example, because each of Takahashi-Na's and Takahashi-Na-Lind's "user identifier" is applied to each data segment, it is applied to "said first data segment." See above 1[c1], 1[e], 15[c1].

15[e2]. transferring said first data segment along with said user identifier to the web service,

305. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[e2] for similar reasons as discussed above for 1[c1], 1[f1] and 15[c1]. For example, because each of Takahashi-Na's and Takahashi-Na-Lind's "user identifier" is transferred with each data segment, it is transferred with "said first data segment." See above 1[c1], 1[e], 15[c1].

15[e3]. receiving a reply message from the web service, corresponding to successful completion of said first data segment transfer;

306. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[e3] for similar reasons as discussed above for 1[c1], 1[g], and 15[c1]. For example, as discussed above, each of Takahashi-Na and Takahashi-Na-Lind renders obvious that an acknowledgement response message ("reply message")

is received for “*successful completion of [each] data segment transfer,*” it is received *corresponding to successful completion of said first data segment transfer.*

See above, 1[c1], 1[g], 15[c1].

15[f1]. applying said user identifier to said second data segment at the application layer level;

307. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[f1] for similar reasons as discussed above for 1[c1], 1[e], and 15[c1]. For example, because each of Takahashi-Na’s and Takahashi-Na-Lind’s “*user identifier*” is applied to each data segment, it is applied to “*said second data segment*” for the reasons discussed above. *See above 1[c1], 1[e], 15[c1].*

15[f2]. transferring said second data segment along with said user identifier to the web service;

308. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[f2] for similar reasons as discussed above for 1[c1], 1[f1], and 15[c1]. For example, because each of Takahashi-Na’s and Takahashi-Na-Lind’s “*user identifier*” is transferred with each data segment, it is transferred with “*said second data segment*” for the reasons discussed above. *See above 1[c1], 1[e], 15[c1].*

15[f3]. receiving a reply message from the web service, corresponding to successful completion of said second data segment transfer,

309. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[f3] for similar reasons as discussed above for 1[c1], 1[g]], and 15[c1]. For the reasons discussed above, both of Takahashi-Na and Takahashi-Na-

Lind render obvious that an acknowledgement response message (“*reply message*”) is received for “*successful completion of [each] data segment transfer,*” and thus it is received *corresponding to successful completion of said second data segment transfer*. See above, 1[c1], 1[g], 15[c1].

15[g]. wherein if said second data segment transfer is interrupted, the software module further performs:

310. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[g] for similar reasons as discussed above for 1[i] and 15[d]. For the reasons discussed above, a POSITA would have recognized or found obvious that “*if said second data segment transfer is interrupted,*” each of Takahashi-Na’s and Takahashi-Na-Lind’s mobile phone application software would have recovered from interruptions for the “*second*” data segment of the upload content, as discussed in 1[i]. See above 1[i].

15[h]. sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;

311. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[h] for similar reasons as discussed above for 1[j].

15[i]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said second data segment by said web service, for said user identifier; and

312. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[i] for similar reasons as discussed above for 1[j], 15[d], and

15[g]. For the reasons discussed above, “*a synchronization reply back from the web service*” would be corresponding to the successfully received data of the “*second*” data segment. See above 1[j], 15[g].

15[j]. resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received, wherein resuming the segmented file transfer of remaining data comprises, the software module, performing:

313. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[j]. For example, as discussed in 1[l] and 15[g], Na discloses that the “mobile communication terminal 10 **resumes the upload with reference to** the sequence number included in **the response message** from the application server 40” (“*resuming, the segmented data transfer... based on the synchronization reply received*”), as shown in step S50 of FIG. 12. TTI-1006, ¶[0096]. A POSITA would have recognized or found obvious that the mobile communication terminal 10 resuming the upload would be *by transferring remaining data of the second data segment* because Na discloses the “sequence number” indicates “from which the upload should be resumed,” therefore making it possible to “**resume the stopped upload from the point at which the previous upload was stopped**” and “the server to match the previously uploaded data and **data to be newly uploaded**” (“*remaining data of the second data segment*”), and preventing “the whole data to be uploaded from scratch” or “[t]he upload of the whole media [...] be conducted again despite

the fact that a portion of the media data has been already uploaded to the server.” TTI-1006, ¶¶[0095], [0007]-[0008].

314. As another example in connection with FIG. 25, Na explicitly discloses that the mobile communication terminal 10 “begins **uploading a portion of the media data that needs to be uploaded additionally (S134)**” (“*resuming, the segmented data transfer by transferring remaining data of the second data segment*”) in response to “receiv[ing] **the response message** from the application server 40” (“*based on the synchronization reply received*”). TTI-1006, ¶[0156].

315. As discussed above, a POSITA would have been motivated and would have found it obvious and straightforward to implement Takahashi’s mobile phone based on Na’s teachings to beneficially allow “an upload of media data stopped by an upload error to be resumed from the point where the error occurred,” as taught by Na. TTI-1006, ¶¶[0008], [0010]; *see above* §VIII.D. In this way, Takahashi’s mobile phone need not restart upload of the whole media data but rather “resumes the upload based on the sequence number of the segments, thereby improving upload efficiency and speed.” TTI-1006, ¶[0021].

15[k]. creating one or more data segments at the application layer level, wherein the one or more data segments comprise remaining data of the second data segment;

316. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[k]. For example, as discussed in 15[j], when resuming an

interrupted upload, Na's mobile communication terminal 10 sends "*remaining data of the second data segment*" (thus, *creating one or more data segments, wherein the one or more data segments comprise remaining data of the second data segment*). TTI-1006, ¶¶[0095], [0007]-[0008], [0156]. A POSITA would have recognized or found obvious that the *create[ed] one or more data segments* would have included the *remaining data of the second data segment* at least because retransmitting successfully received data would be inefficient. Further, because this claim limitation recites creating "one or more" segments, simply transmitting the entirety of the '*remaining data of the second data segment*' as one data segment would satisfy the claim.

317. Na discloses performing such an operation "in accordance with ... the **HTTP** 1.1 protocol," and a POSITA would have recognized or found obvious that the one or more data segments would have been created *at the application layer level* because "*the second data segment*" and thus "*the remaining data of the second data segment*" are created *at the application layer level* per 15[c1]. A POSITA would have recognized or found obvious to resume the upload and to transmit the remaining data of the second data segment by creating the one or more data segments "*at the application layer level*" "in accordance with ... the **HTTP** 1.1 protocol" disclosed in Na. TTI-1006, ¶[0052].

318. As discussed above, a POSITA would have been motivated and would have found it obvious and straightforward to implement Takahashi's mobile phone based on Na's teachings to achieve multiple benefits. TTI-1006, ¶¶[0008], [0010], [0021]; *see above* §VIII.D. In doing so, a POSITA also would have been motivated and would have found it obvious and straightforward to implement Takahashi's mobile phone to *create one or more data segments* that include the *remaining data of the second data segment* so as to resume the image data transmission from "the point where the error occurred," as taught by Na. TTI-1006, ¶¶[0008], [0010].

15[l]. applying a segment identifier to each of said one or more data segments at the application layer level;

319. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[l] for similar reasons as discussed above for 1[d].

15[m]. applying said user identifier to each of said one or more data segments at the application layer level; and

320. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[m] for similar reasons as discussed above for 1[e].

15[n]. transferring said one or more data segments along with said user identifier and said corresponding segment identifier to the web service,

321. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[n] for similar reasons as discussed above for 1[f1].

15[o]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service.

322. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 15[o] for similar reasons as discussed above for 1[f2].

16. The method of claim 15, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

323. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 16 for similar reasons as discussed above for claim 3.

17. The method of claim 15, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

324. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 17 for similar reasons as discussed above for claim 4.

18. The method of claim 15, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

325. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 18 for similar reasons as discussed above for claim 5.

19. The method of claim 15, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

326. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 19 for similar reasons as discussed above for claim 6.

20. The method of claim 15, wherein the segment identifier corresponds to the content range of the remaining data of the file.

327. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 14. for similar reasons as discussed above for claim 7.

21[pre]. A method of segmented file transfer from a Bluetooth enabled mobile device to a web service, the method comprising:

328. To the extent the preamble is limiting, both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[pre] for similar reasons as discussed above for 1[pre].

21[a]. determining size of file to be transferred by the Bluetooth enabled mobile device;

329. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[a] in multiple ways for similar reasons as discussed above for 1[b].

21[b]. transferring said file by the Bluetooth enabled mobile device to the web service, wherein transferring said file comprises, the Bluetooth enabled mobile device, performing:

330. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[b] for similar reasons as discussed above for 8[b].

21[c1]. segmenting said file into a first data segment and a second data segment at an application layer level,

331. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[c1] for similar reasons as discussed above for 15[c1].

21[c2]. wherein said file is segmented in part due to size of said file and data transfer capabilities of said Bluetooth enabled mobile device,

332. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[c2] for similar reasons as discussed above for 1[c2].

21[d]. wherein the second data segment comprises file data and the first data segment comprises data associated with file data, and wherein the data associated with file data is at least one of name of the file, size of the file, and type of the file;

333. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[d] for similar reasons as discussed above for 15[d].

21[e1]. applying a user identifier to said first data segment at the application layer level, wherein the user identifier uniquely identifies a particular user of the web service;

334. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[e] for similar reasons as discussed above for 15[e1].

21[e2]. transferring said first data segment along with said user identifier to the web service,

335. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[f] for similar reasons as discussed above for 15[e2].

21[e3]. receiving a reply message from the web service, corresponding to successful completion of said first data segment transfer;

336. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[g] for similar reasons as discussed above for 15[e3].

21[f1]. applying said user identifier to said second data segment at the application layer level;

337. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[h1] for similar reasons as discussed above for 15[f1].

21[f2]. transferring said second data segment along with said user identifier to the web service;

338. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[h2] for similar reasons as discussed above for 15[f2].

21[f3]. receiving a reply message from the web service, corresponding to successful completion of said second data segment transfer,

339. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[h3] for similar reasons as discussed above for 15[f3].

21[g]. wherein if said second data segment transfer is interrupted, the Bluetooth enabled mobile device performs:

340. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[i] for similar reasons as discussed above for 15[g].

21[h]. sending, a synchronization request to the web service from the application layer level, for said user identifier, after a network connection to the web service is restored;

341. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[j] for similar reasons as discussed above for 1[j].

21[i]. receiving, a synchronization reply back from the web service at the application layer level, corresponding to the successfully received data of said second data segment by said web service, for said user identifier; and

342. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[k] for similar reasons as discussed above for 15[i].

21[j]. resuming, the segmented file transfer by transferring remaining data of the second data segment based on the synchronization reply received, wherein resuming the segmented file transfer of remaining data comprises, the Bluetooth enabled mobile device, performing:

343. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[l] for similar reasons as discussed above for 15[j].

21[k]. creating one or more data segments at the application layer level, wherein the one or more data segments comprise remaining data of the second data segment;

344. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[m] for similar reasons as discussed above for 15[k].

21[l]. applying a segment identifier to each of said one or more data segments at the application layer level;

345. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[n] for similar reasons as discussed above for 1[d].

21[m]. applying said user identifier to each of said one or more data segments at the application layer level; and

346. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[o] for similar reasons as discussed above for 1[e].

21[n]. transferring said one or more data segments along with said user identifier and said corresponding segment identifier to the web service,

347. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[p] for similar reasons as discussed above for 1[f1].

21[o]. wherein transferring the data segment further comprises one or more of: configuring the data segment so that the data segment can be received by the web service; and configuring the data segment so that the data segments can be reassembled using segment identifiers by the web service.

348. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious element 21[q] for similar reasons as discussed above for 1[f2].

22. The method of claim 21, wherein said file data comprises one or more of audio data, video data, image data, text data, or digital data.

349. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 22 for similar reasons as discussed above for claim 3.

23. The method of claim 21, wherein the user identifier comprises one or more elements from user information, and wherein the one or more elements of the user information comprise user-name, user-password, user-device-information, user settings, or user preferences.

350. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 23 for similar reasons as discussed above for claim 4.

24. The method of claim 21, wherein said data transfer capabilities of said Bluetooth enabled mobile device are dependent on the limited network processing capability of the mobile device cellular network internet connection.

351. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 24 for similar reasons as discussed above for claim 5.

25. The method of claim 21, wherein the application layer is defined as one of following, the layer 7 in the OSI layer stack, or the application layer of the Internet protocol suite.

352. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 25 for similar reasons as discussed above for claim 6.

26. The method of claim 21, wherein the segment identifier corresponds to the content range of the remaining data of the file.

353. Both of the Takahashi-Na and Takahashi-Na-Lind combinations render obvious claim 26 for similar reasons as discussed above for claim 7.

IX. ADDITIONAL REMARKS

354. I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

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

Dated: 30 March 2024

By: 

APPENDIX A

Patrick Gerard Traynor

Professor

Associate Chair for Research in CISE

John and Mary Lou Dasburgh Preeminent Chair in Engineering

Department of Computer & Information Science & Engineering (CISE)

University of Florida

E301 CSE Building, PO Box 116120

Gainesville, FL 32611 USA

traynor@cise.ufl.edu

<http://www.cise.ufl.edu/~traynor>

Table of Contents

EDUCATIONAL BACKGROUND	4
EMPLOYMENT HISTORY	4
CURRENT FIELDS OF INTEREST	4
I. TEACHING	6
A. Courses Taught	6
B. Continuing Education	6
C. Curriculum Development	6
D. Individual Student Guidance	7
E. Teaching Honors and Awards	11
II. RESEARCH AND CREATIVE SCHOLARSHIP	12
A. Thesis	12
B. Published Journal Papers (Refereed)	12
C. Published Books and Parts of Books	14
D. Edited Proceedings	14
E. Conference Presentations	14
E.1. Conference Presentations with Proceedings (Refereed)	14
E.2. Conference Presentations with Proceedings (Non-Refereed)	20
E.3. Conference Presentations without Proceedings	20
F. Other	20
F.1. Submitted Journal Papers	20
F.2. Refereed Research Reports	20
F.3. Software	20
F.4. Published Papers (Non-Refereed)	21
F.5. Books in Preparation	21
F.6. Workshops and External Courses	21
G. Research Proposals and Grants (Principal Investigator)	22
H. Research Proposals and Grants (Contributor)	24
I. Research Honors and Awards	25
III. SERVICE	27
A. Professional Activities	27
A.1. Memberships and Activities in Professional Societies	27
A.2. Conference Committee Activities	27
B. On-Campus Committees	28
B.1. University of Florida	28
B.2. Georgia Tech	29
C. Special Assignments	29
D. Ph.D. Examining Committees	29
E. External Member of M.S. Examining Committee	33
F. Consulting and Advisory Appointments	33
G. Civic Activities	33
IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION	34
A. Honors and Awards	34
B. Invited Conference Session Chairmanships	34
C. Professional Registration	34
D. Patents	34
E. Editorial and Reviewer Work for Technical Journals and Publishers	35
F. Expert Witness Services	37
V. OTHER CONTRIBUTIONS	39
A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia)	39

B. Special Activities	43
---------------------------------	----

EDUCATIONAL BACKGROUND

Degree	Year	University	Field
Ph.D.	2008	Pennsylvania State University State College, PA <i>Dissertation:</i> Characterizing the Impact of Ridigity on the Security of Cellular Telecommunications Networks <i>Advisors:</i> Thomas F. La Porta and Patrick D. McDaniel	Computer Science & Engineering
M.S.	2004	Pennsylvania State University State College, PA	Computer Science & Engineering
B.S.	2002	University of Richmond Richmond, VA <i>Minors:</i> Biology, Business Admin	Computer Science

EMPLOYMENT HISTORY

Title	Organization	Years
Professor	University of Florida	August 2018–Present
Associate Professor	University of Florida	August 2014–July 2018
Associate Professor	Georgia Institute of Technology	March 2014–August 2014
Assistant Professor	Georgia Institute of Technology	2008–March 2014
Research Assistant	Pennsylvania State University	2004–2008
Teaching Assistant	Pennsylvania State University	2004

CURRENT FIELDS OF INTEREST

My research focuses on the security of cellular/telephony networks and mobile systems. The security of these systems generally relies on their closed nature and trust in the honest behavior of users. However, with the recent disintegration of these assumptions and with over than six billion subscribers around the world, cellular and mobile systems represent the next great expansion in global critical infrastructure and, because of their unique characteristics, require new and different approaches to security.

Recognizing this, my research focuses on three specific themes: (1) developing efficient techniques to allow telephony providers and customers to authenticate the origin of incoming calls; (2) measuring and improving the security of emerging mobile financial systems and (3) efficient and strong privacy-preserving techniques for mobile devices. Additionally, I have significant expertise in fraud detection, particularly for payment systems.

I have a strong interest in solutions that can be deployed in both the short and long terms, and am actively engaging both industry and government in this capacity. My research, if successful, will help to not only improve the general security of networked devices, but also to maintain the historical reliability of telephony networks as they become the dominant digital access technology.

I. TEACHING

A. Courses Taught

Semester/Year	Course Number & Title	Number of Students	Comments
Fall 2022	CNT 5410 Computer and Network Security	75	New Topics
Fall 2021	CNT 5410 Computer and Network Security	45	New Topics
Fall 2019	CNT 5410 Computer and Network Security	28	New Topics
Fall 2018	CIS 6930 Cellular and Mobile Network Security	16	New Course
Fall 2017	CNT 5410 Computer and Network Security	27	New Topics
Fall 2016	CNT 5410 Computer and Network Security	60	New Topics
Spring 2016	CNT 5410 Computer and Network Security	13	New Topics
Spring 2015	CNT 5410 Computer and Network Security	12	New Topics
Fall 2014	CNT 5410 Computer and Network Security	30	New Course
Spring 2014	CS 6262 Network Security	55	New Projects
Fall 2013	CS 3251 Computer Networks I	73	Expanded Syllabus
Spring 2013	CS 6262 Network Security	65	All New Projects
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2012	CS 8803 Cellular & Mobile Network Security	17	New Topics
	CS 8001 Information Security Seminar	20	New Speakers
Spring 2011	CS 8001 Information Security Seminar	20	New Speakers
Fall 2011	CS 6262 Network Security	27	Expanded Syllabus
	CS 8001 Information Security Seminar	35	New Speakers
Spring 2011	CS 3251 Computer Networks I	61	Expanded Syllabus
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2010	CS 8803/4803 Cellular & Mobile Network Security	16	New Course
	CS 8001 Information Security Seminar	31	New Speakers
Fall 2009	CS 6262 Network Security	55	Expanded Syllabus
Spring 2009	CS 3251 Computer Networks I	45	Expanded Syllabus
Fall 2008	CS 8003 Destructive Research	10	New Course

Guest lecturer for CS 4235 (Introduction to Information Security) and CS 8803 (e-Democracy) in Fall 2008.

Advised ECE 4811/CS 4802 (Vertically Integrated Project) with Ed Coyle

B. Continuing Education

None.

C. Curriculum Development

CS 8803 Cellular and Mobile Network Security: *Fall 2010.* Developed an entirely new course around security issues facing cellular and mobile networks. Students learned about wireless basics, spectrum issues, core network architectures (GSM, ISDN, IMS, SIP), air interfaces (GSM, 3G), mobility management, authentication, mobile phone operating systems (Android, iPhone), Android security, congestion and denial of service, privacy and eavesdropping. Students also complete a research project and aim towards publishing this work at a major venue. My aim is for this class to become part of the regular offering of security courses and receive a non-8803 number. Semester projects were also judged and encouraged using a “venture capital” model, in which students had to pretend as if they were pitching their ideas for a start-up

company to potential investors.

CS 6262 Network Security: *Fall 2009.* Totally rewrote the syllabus and slide material, giving the class its first major overhaul in a number of years. While many old themes remain, new lecture blocks including Web Security, Cellular Security and Social Engineering were developed from scratch. This new course material was made available to all other faculty members teaching this class, who have since used my slides and syllabus.

CS 3251 Computer Networks I: *Spring 2009.* Modified undergraduate networking course to include a persistent focus on security at all layers of the protocol stack. I have also created new lectures focusing on the physical layer and cellular networks and new exams to include all of the abovementioned changes.

CS 8803 Destructive Research: *Fall 2008.* Developed course based around understanding how so-called secure systems have been defeated by attackers. With such knowledge, students would have the context to develop the next generation of more secure systems. I delivered more than 1/3 of the lectures in this seminar course and paid special focus on vulnerabilities in cellular networks, analog telecommunications and electronic voting. Students were also instructed on techniques for performing research, writing technical papers and making conference and lecture-style presentations. I have offered these slides to future 7001 classes to help impact a wider audience.

D. Individual Student Guidance

1. Research Scientists Supervised

None.

2. Ph.D. Students Graduated

Chaitrali Amrutkar Georgia Institute of Technology

Fall 2009–Fall 2013

Her research discovered vulnerabilities in mobile web browsers and developed techniques to detect malicious mobile web pages. Joined Oracle in Spring 2014.

Jasmine Bowers University of Florida

Fall 2015–Summer 2020

Her research focuses on mobile applications, and the development of tools for building secure systems. Now: Research Scientist, MITRE

Henry “Hank” Carter Georgia Institute of Technology

Fall 2010–Spring 2016

Developing techniques for secure function evaluation for privacy-preserving applications on constrained mobile devices. Now: Assistant Professor, Villanova University

Italo Dacosta Georgia Institute of Technology

Fall 2008–Summer 2012

Co-advised with Mustaque Ahamad. Research on scaling performance of SIP network components. Graduated Summer 2012, currently research scientist at EPFL.

David Dewey Georgia Institute of Technology

Fall 2011–Summer 2015

Investigated compiler techniques to remove software vulnerabilities from code. Now CTO of MailChimp.

Brad Reaves University of Florida
Fall 2014–Spring 2017
Develop strong authentication techniques for cellular networks. Now: Assistant Professor at North Carolina State University.

Nolen Scaife University of Florida
Fall 2014–Spring 2019
Developed techniques to detect credit card skimming. First: Assistant Professor at the University of Colorado Boulder. Now: Director, Global Cyber Intelligence at Walmart

Imani Sherman University of Florida
Fall 2018–Summer 2021
Developing usable interfaces against robocalls. Co-advised with Juan Gilbert. Now: Assistant Professor at the University of California, San Diego

Luis Vargas University of Florida
Fall 2016–Summer 2021
Developing techniques for network-based detection and mitigation of malware in a healthcare environment. Now: Data Scientist at the Alethia Group

Hadi Abdullah University of Florida
Fall 2016–Summer 2022
Evaluating the security of ML-driven voice interfaces. Now: Research Scientist at Visa Research

Christian Peeters University of Florida
Fall 2016–Summer 2022
Develop techniques to detect and defend against call and message interception attacks in cellular networks. Now: Research Scientist at Harbor Labs

2. Ph.D. Students Supervised

Logan Blue University of Florida
Fall 2016–Present
Investigating facial feature reconstruction from voice recordings.

Nathaniel Bennett University of Florida
Fall 2022–Present
Finding vulnerabilities in cellular core networks via fuzzing.

Cassidy Gibson University of Florida
Fall 2019–Present
Investigating weaknesses in web software.

Ryon Kennedy University of Florida
Fall 2020–Present
Finding vulnerabilities in cellular core networks via fuzzing.

Seth Layton University of Florida
Fall 2020–Present
Detecting deepfakes in audio samples.

Allison Lu University of Florida
Fall 2022–Present
Measuring repeatability in computer security.

Daniel Olszewski University of Florida
Fall 2019–Present
Removing unwanted/insecure features from software.

Tyler Tucker University of Florida
Fall 2021–Present
Evaluating the security of Bluetooth/cellular radios.

Kevin Warren University of Florida
Fall 2019–Present
Detecting deepfake audio through linguistic information.

3. Ph.D. Students - Other

Saurabh Chakradeo Georgia Institute of Technology
Fall 2010–Spring 2013
Research exploring malicious mobile applications. Left to join Facebook.

Brendan Dolan-Gavitt Georgia Institute of Technology
Spring 2009
Research project on using kernel type graphs to detect dummy structures.

Eric (Yu) Liu Georgia Institute of Technology
Fall 2008
Research on the spread of malware through cellular infrastructure.

Chaz Lever Georgia Institute of Technology
Fall 2011–Spring 2014
Developing techniques to measure the spread of malware in cellular networks. Left Georgia Tech to create a startup.

Frank Park Georgia Institute of Technology
Fall 2008–Spring 2010
Research on multi-factor authentication using cellular phones. Left program after failing comprehensive exam to join startup.

Ferdinand Schober Georgia Institute of Technology
Fall 2009–Summer 2010
Developed mechanisms for smart networks and smart mobile devices to fight infection and provide remote remediation. Returned to Microsoft.

4. M.S. Students Supervised

Chaitrali Amrutkar Georgia Institute of Technology
Fall 2008–Spring 2009
Research on improving performance of security critical functions in IMS cellular core. Completed her Ph.D with me at GT.

Logan Blue University of Florida
Fall 2015–Spring 2016
Investigated problems of cellular and network security.

David Dewey Georgia Institute of Technology
Fall 2009–Spring 2010
Research on security issues caused by transitive trust assumptions in the Windows COM infrastructure. Completed his Ph.D. with me at GT.

Christopher Grayson Georgia Institute of Technology

Fall 2012–Fall 2013

Developed continuous authentication mechanisms using the multitude of sensors available on a mobile phone. Now at Bishop Fox Consulting (industry).

Young Seuk Kim Georgia Institute of Technology

Fall 2012–Fall 2013

Performed research that compared the security vulnerabilities found in the traditional and mobile web.

Daniel Komaromy Georgia Institute of Technology

Fall 2008–Summer 2009

Research on building a real-time streaming audio system using attribute-based crypto for broadcast encryption.

Nigel Lawrence Georgia Institute of Technology

Fall 2011–Spring 2012

Discovered hijacking attacks in SNMPv3, a widely used and thought to be secure network management protocol. Now at Solute (industry).

Philip Marquardt Georgia Institute of Technology

Fall 2009–Present

Research on developing an iPhone application to prevent individuals from being profiled by Shopper Loyalty Programs. First with MIT Lincoln Labs, now Raytheon

Rishikesh Naik Georgia Institute of Technology

Fall 2008–Spring 2010

Research on converting expensive cryptographic primitives (e.g., Secure Function Evaluation) into efficient applications for mobile phones. Now with Cisco Systems.

Ashish Nautiyal CISE

Fall 2015–Spring 2016

Research on connecting telephone calls to the larger authentication infrastructure.

Nilesh Nipane Georgia Institute of Technology

Fall 2008–Spring 2010

Research on creating provably anonymous networks on a base of secure function evaluation. Now with VMWare.

Walter “Nolen” Sciafe Georgia Institute of Technology

Spring 2012–Spring 2014

Developed the OnionDNS architecture, which prevents domain delisting attacks by leveraging a Tor hidden service. Joined Ph.D. program at UF.

Tyler Tucker University of Florida

Fall 2018–Spring 2021

Evaluating the security of Bluetooth radios.

5. M.S. Special Problems Students

Siddhant Deshmukh University of Florida

Fall 2016–Present

Developed tools for analysis of mobile digital financial services.

Chinmay Gangakhedkar Georgia Institute of Technology

Spring 2009

Research on multi-factor authentication using mobile phones.

Christopher Grayson Georgia Institute of Technology

Spring 2013

Research on continuous authentication using mobile phones.

Aarushi Karnany University of Florida

Fall 2016–Present

Developed tools for analysis of mobile digital financial services.

Rohit Matthews Georgia Institute of Technology

Spring 2011

Developed mobile phone-based tools for measuring performance and reachability throughout the Internet.

Ashwin Narasimhan Georgia Institute of Technology

Spring 2009

Research on developing efficient security mechanisms for the IMS cellular core.

Aamir Poonawalla Georgia Institute of Technology

Spring 2010

Helped develop a call provenance infrastructure, which included both networking and machine learning components.

Erin Reddick Georgia Institute of Technology

Fall 2008–Fall 2009

Research on IPTV security with GTRI.

Lalanthika Vasudevan Georgia Institute of Technology

Spring 2009

Research on developing efficient security mechanisms for the IMS cellular core.

6. Undergraduate Special Problems Students

Ethan Shernan Georgia Institute of Technology

Spring 2014

Developed an infrastructure for detecting billing bypass fraud attacks.

Young Seuk Kim Georgia Institute of Technology

Fall 2011–Spring 2012

Developed a mobile phone application for taking measurements of cellular networks.

Dane Van Dyck Georgia Institute of Technology

Summer 2009

Research on virtualization support for mobile phones.

E. Teaching Honors and Awards

1. Georgia Institute of Technology, CETL “Thanks For Being A Great Teacher” Award, Fall 2013.
2. Georgia Institute of Technology, CETL “Thanks For Being A Great Teacher” Award, Fall 2012.
3. United State Army Signal Corps, “Helmet” Award, 2010.
4. Georgia Institute of Technology, CETL “Thanks For Being A Great Teacher” Award, Spring 2009.
5. Pennsylvania State University CSE Graduate Student Teaching Award, 2005

II. RESEARCH AND CREATIVE SCHOLARSHIP

A. Thesis

1. Patrick Gerard Traynor. *Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks*. PhD thesis, The Pennsylvania State University, May 2008.

B. Published Journal Papers (Refereed)

1. Cassidy Gibson, Vanessa Frost, Katie Platt, Washington Garcia, Luis Vargas, Sara Rampazzi, Vincent Bindschaedler, Patrick Traynor, and Kevin Butler. Analyzing the Monetization Ecosystem of Stalkerware. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2022.
2. Bradley Reaves, Luis Vargas, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Characterizing the Security of the SMS Ecosystem with Public Gateways. *ACM Transactions on Privacy and Security (TOPS)*, 22(1), 2018.
3. Patrick Traynor, Kevin Butler, Jasmine Bowers, and Bradley Reaves. FinTechSec: Addressing the Security Challenges of Digital Financial Services. *IEEE S&P Magazine*, 15(5):85–89, 2017.
4. Nolen Scaife, Henry Carter, Rachel Jones, Lyrissa Lidsky, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. *International Journal of Information Security (IJIS)*, 2017.
5. Bradley Reaves, Jasmine Bowers, Nolen Scaife, Adam Bates, Arnav Bharatiya, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. *ACM Transactions on Privacy and Security (TOPS)*, 2017.
6. Henry Carter and Patrick Traynor. OPFE: Outsourcing Computation for Private Function Evaluation. *International Journal of Information and Computer Security (IJICS)*, 2017.
7. Stephan Heuser, Bradley Reaves, Praveen Kumar Pendyala, Henry Carter, Alexandra Dmitrienko, William Enck, Negar Kiyavash, Ahmad-Reza Sadeghi, and Patrick Traynor. Phonion: Practical Protection of Metadata in Telephony Networks. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2017.
8. Bradley Reaves, Jasmine Bowers, Sigmond A. Gorski III, Olabode Anise, Rahul Bobhate, Raymond Cho, Hiranava Das, Sharique Hussain, Hamza Karachiwala, Nolen Scaife, Byron Wright, Kevin Butler, William Enck, and Patrick Traynor. *droid: Assessment and Evaluation of Android Application Analysis Tools. *ACM Computing Surveys (CSUR)*, 2016.
9. Chaitrali Amrutkar, Young Seuk Kim, and Patrick Traynor. Detecting Mobile Malicious Webpages in Real Time. *IEEE Transactions on Mobile Computing (TMC)*, To Appear 2016.
10. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. *Journal of Security and Communication Networks (SCN)*, To Appear 2016.
11. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. *Journal of Computer Security (JCS)*, 24(2):137–180, 2016.
12. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. *Journal of Computer Security (JCS)*, 23(2):167–195, 2015.
13. Henry Carter, Chaitrali Amrutkar, Italo Dacosta, and Patrick Traynor. For Your Phone Only: Custom Protocols for Efficient Secure Function Evaluation on Mobile Devices. *Journal of Security and Communication Networks (SCN)*, 7(7):1165–1176, 2014.

14. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. An Empirical Evaluation of Security Indicators in Mobile Web Browsers. *IEEE Transactions on Mobile Computing (TMC)*, 14(5), 2015.
15. Andrew Harris, Seymour Goodman, and Patrick Traynor. Privacy and Security Concerns Associated with Mobile Money Applications in Africa. *Washington Journal of Law, Technology & Arts*, 8(3), 2013.
16. Italo Dacosta, Saurabh Chakradeo, Mustaque Ahamad, and Patrick Traynor. One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens. *ACM Transactions on Internet Technology (TOIT)*, 12(1), 2012.
17. Cong Shi, Xiapu Luo, Patrick Traynor, Mostafa Ammar, and Ellen Zegura. ARDEN: Anonymous netwoRking in Delay tolErant Networks. *Journal of Ad Hoc Networks*, 10(6):918–930, 2012.
18. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. *IEEE Transactions on Mobile Computing (TMC)*, 11(6):983–994, 2012.
19. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22(11):1804–1812, 2011.
20. Patrick Traynor, Chaitrali Amrutkar, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. From Mobile Phones to Responsible Devices. *Journal of Security and Communication Networks (SCN)*, 4(6):719 – 726, 2011.
21. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. *Journal of Computer Security (JCS)*, 18(5):799–837, 2010.
22. Patrick Traynor, Kevin Butler, William Enck, Kevin Borders, and Patrick McDaniel. malnets: Large-Scale Malicious Networks via Compromised Wireless Access Points. *Journal of Security and Communication Networks (SCN)*, 2(3):102–113, 2010.
23. Patrick Traynor. Securing Cellular Infrastructure: Challenges and Opportunities. *IEEE Security & Privacy Magazine*, 7(4), 2009.
24. Kevin Butler, Sunam Ryu, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(12):1803–1815, 2009.
25. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks On Open Functionality in SMS-Capable Cellular Networks. *IEEE/ACM Transactions on Networking (TON)*, 17(1), 2009.
26. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. *ACM Transactions on Information and System Security (TISSEC)*, 2008.
27. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. *Journal of Computer Security (JCS)*, 16(6):713–742, 2008.
28. Patrick Traynor, Raju Kumar, Heesook Choi, Sencun Zhu, Guohong Cao, and Thomas La Porta. Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing (TMC)*, 6(6), 2007.

C. Published Books and Parts of Books

1. Andrew Harris, Frank S. Park, Seymour Goodman, and Patrick Traynor. *Emerging Privacy and Security Concerns for Digital Wallet Deployment*. Privacy in America: Interdisciplinary Perspectives. Scarecrow Press, July 2011.
2. Kevin Butler, William Enck, Patrick Traynor, Jennifer Plasterr, and Patrick McDaniel. *Privacy Preserving Web-Based Email*. Algorithms, Architectures and Information Systems Security, Statistical Science and Interdisciplinary Research. World Scientific Computing, November 2008.
3. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. *Security for Telecommunications Networks*. Number 978-0-387-72441-6 in Advances in Information Security Series. Springer, August 2008.

D. Edited Proceedings

None.

E. Conference Presentations

E.1. Conference Presentations with Proceedings (Refereed)

1. Christian Peeters, Tyler Tucker, Anushri Jain, Kevin Butler, and Patrick Traynor. LeopardSeal: Detecting Call Interception via Audio Rogue Base Stations. In *Proceedings of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2023.
2. Tyler Tucker, Hunter Searle, Kevin Butler, and Patrick Traynor. Blue's Clues: Practical Discovery of Non-Discoverable Bluetooth Devices. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2023.
3. Hadi Abdullah, Aditya Karlekar, Saurabh Prasad, Muhammad Sajidur Rahman, Logan Blue, Luke Bauer, Vincent Bindschaedler, and Patrick Traynor. Attacks as Defenses: Designing Robust Audio CAPTCHAs Using Attacks on Automatic Speech Recognition Systems. In *Symposium on Network and Distributed System Security (NDSS)*, 2023.
4. Daniel Olszewski, Sandeep Sathyanarayana, Weidong Zhu, Kevin Butler, and Patrick Traynor. HallMonitor: A Framework for Identifying Network Policy Violations in Software. In *IEEE Conference on Communications and Network Security (CNS)*, 2022.
5. Hadi Abdullah, Aditya Karlekar, Vincent Bindschaedler, and Patrick Traynor. Demystifying Limited Adversarial Transferability in Automatic Speech Recognition Systems. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. (Acceptance rate: 32%).
6. Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O'Dell, Kevin Butler, and Patrick Traynor. Who Are You (I Really Wanna Know)? Detecting Audio DeepFakes Through Vocal Tract Reconstruction. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2022. (Acceptance rate: 17.2%).
7. Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin R. B. Butler. FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware. In *Symposium on Network and Distributed System Security (NDSS)*, 2022. (Acceptance rate: 16.2%).
8. Christian Peeters, Christopher Patton, Imani N. Sherman, Daniel Olszewski, Thomas Shrimpton, and Patrick Traynor. SMS OTP Security (SOS): Hardening SMS-Based Two Factor Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2022. (Acceptance rate: 18.2%).

9. Hadi Abdullah, Muhammad Sajidur Rahman, Christian Peeters, Cassidy Gibson, Washington Garcia, Vincent Bindschaedler, Thomas Shrimpton, and Patrick Traynor. Beyond L_p Clipping: Equalization based Psychoacoustic Attacks against ASRs. In *The Asian Conference on Machine Learning (ACML)*, 2021.
10. Imani Sherman and Daniel Delgado and Juan Gilbert and Jaime Ruiz and Patrick Traynor. Characterizing User Comprehension in the STIR/SHAKEN Anti-Robocall Standard. In *Proceedings of the Annual Research Conference on Communications Information and Internet Policy (TPRC 49)*, 2021.
11. Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).
12. Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear “No Evil”, See “Kenansville”: Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).
13. Imani Sherman, Jasmine Bowers, Liz-Laure Laborde, Juan E. Gilbert, Jaime Ruiz, and Patrick Traynor. Truly Visual Caller ID? An Analysis of Anti-Robocall Applications and their Accessibility to Visually Impaired Users. In *IEEE International Symposium on Technology and Society (IEEE ISTAS)*, 2020.
14. Imani Sherman, Jasmine Bowers, Keith McNamara, Juan Gilbert, Jaime Ruiz, and Patrick Traynor. Are You Going to Answer That? Measuring User Responses to Anti-Robocall Application Indicators. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2020. (Acceptance rate: 17.4%).
15. Joseph Choi, Dave Tian, Grant Hernandez, Christopher Patton, Benjamin Mood, Thomas Shrimpton, Patrick Traynor, and Kevin Butler. A Hybrid Approach to Secure Function Evaluation Using SGX. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, 2019. (Acceptance Rate: 17.0% for full papers).
16. Vanessa Frost, Dave Tian, Christie Ruales, Patrick Traynor, and Kevin Butler. Examining DES-based Cipher Suite Support within the TLS Ecosystem. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, 2019. (Acceptance Rate: 22.0% for all papers).
17. Dave Tian, Joseph Choi, Grant Hernandez, Patrick Traynor, and Kevin Butler. A Practical Intel SGX Setting for Linux Containers in the Cloud. In *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY’19)*, 2019. (Acceptance rate: 23.5%).
18. Nolen Scaife, Jasmine Bowers, Christian Peeters, Grant Hernandez, Imani Sherman, Lisa Anthony, and Patrick Traynor. Kiss from a Rogue: Evaluating Detectability of Pay-at-the-Pump Card Skimmers. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2019. (Acceptance rate: 12.0%).
19. Jasmine Bowers, Imani Sherman, Kevin Butler, and Patrick Traynor. Characterizing Security and Privacy Practices in Emerging Digital Credit Applications. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019. (Acceptance rate: 25.6%).
20. Hadi Abdullah, Washington Garcia, Christian Peeters, P. Traynor, K. Butler, and J. Wilson. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).

21. Lius Vargas, Logan Blue, Vanessa Frost, Christopher Patton, N. Scaife, K. Butler, and P. Traynor. Digital Healthcare-Associated Infection Analysis of a Major Multi-Campus Hospital System. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).
22. Dominik Wermke, Nicolas Huaman, Yasemin Acar, Bradley Reaves, Patrick Traynor, and Sascha Fahl. A Large Scale Investigation of Obfuscation Use in Google Play. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2018. Acceptance Rate: 20.1%.
23. Nolen Scaife, Christian Peeters, and Patrick Traynor. Fear the Reaper: Characterization and Fast Detection of Card Skimmers. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.
24. Dave (Jing) Tian, Grant Hernandez, Joseph Choi, Vanessa Frost, Christie Raules, Kevin Butler, Patrick Traynor, Hayawardh Vijayakumar, Lee Harrison, Amir Rahmati, and Mike Grace. Attention Spanned: Comprehensive Vulnerability Analysis of AT Commands Within the Android Ecosystem. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.
25. Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2MA: Verifying Voice Commands via Two Microphone Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2018. (Acceptance Rate: 20.0%).
26. Nolen Scaife, Christian Peeters, Camilo Velez, Hanqing Zhao, Patrick Traynor, and David Arnold. The Cards Aren't Alright: Detecting Counterfeit Gift Cards Using Encoding Jitter. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).
27. Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).
28. Tyler Ward, Joseph Choi, Kevin Butler, John M. Shea, Patrick Traynor, and Tan Wong. Privacy Preserving Localization Using a Distributed Particle Filtering Protocol. In *IEEE MILCOM*, 2017. (Acceptance Rate: 56%).
29. Bradley Reaves and Logan Blue and Hadi Abdullah and Luis Vargas and Patrick Traynor and Thomas Shrimpton. AuthentiCall: Efficient Identity and Content Authentication for Phone Calls. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2017. (Acceptance Rate: 16.3%).
30. Jasmine Bowers and Bradley Reaves and Imani N. Sherman and Patrick Traynor and Kevin Butler. Regulators, Mount Up! Analysis of Privacy Policies for Mobile Money Applications. In *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2017. (Acceptance Rate: 26.5%).
31. Bradley Reaves, Logan Blue, and Patrick Traynor. AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).
32. Dave Tian, Nolen Scaife, Adam Bates, Kevin Butler, and Patrick Traynor. Making USB Great Again with USBFILTER. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).
33. Bradley Reaves, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Detecting SMS Spam in the Age of Legitimate Bulk Messaging. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2016. (Acceptance Rate: 35.0%).

34. Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin Butler. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016. (Acceptance Rate: 17.6%).
35. Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2016. (Acceptance Rate: 13.0%).
36. Benjamin Mood, Debayan Gupta, Henry Carter, Kevin Butler, and Patrick Traynor. Frigate: A Validated, Extensible, and Efficient Compiler and Interpreter for Secure Computation. In *Proceedings of the IEEE European Symposium on Security and Privacy*, 2016. (Acceptance Rate: 17.3%).
37. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. In *Proceedings of the International Conference on Cryptology and Network Security*, 2015. (Acceptance Rate: 52.9%).
38. Nolen Scaife, Henry Carter, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2015. (Acceptance Rate: 28.1%).
39. Bradley Reaves, Nolen Scaife, Adam Bates, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).
40. Bradley Reaves, Ethan Shernan, Adam Bates, Henry Carter, and Patrick Traynor. Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).
41. David Dewey, Bradley Reaves, and Patrick Traynor. Uncovering Use-After-Free Conditions In Compiled Code. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, 2015. (Acceptance Rate: 22%).
42. Ethan Shernan, Henry Carter, Dave Tian, Patrick Traynor, and Kevin Butler. More Guidelines Than Rules: CSRF Vulnerabilities from Noncompliant OAuth 2.0 Implementations. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2015. (Acceptance Rate: 22.7%).
43. Henry Carter, Charles Lever, and Patrick Traynor. Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2014. (Acceptance Rate: 19.9%).
44. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2013. (Acceptance Rate: 16.2%).
45. Chaitrali Amrutkar, Matti Hiltunen, Shobha Venkataraman, Kaustubh Joshi, Patrick Traynor, Trevor Jim, and Oliver Spatscheck. Why is My Smartphone Slow? On The Fly Diagnosis of Poor Performance on the Mobile Internet. In *Proceedings of The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2013. (Acceptance Rate: 19.6%).
46. Saurabh Chakradeo, Brad Reaves, Patrick Traynor, and William Enck. MAST: Triage for Market-scale Mobile Malware Analysis. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2013. (Acceptance Rate: 15.0%)(Best Paper).
47. Charles Lever, Manos Antonakakis, Brad Reaves, Patrick Traynor, and Wenke Lee. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2013. (Acceptance rate: 18.8%).

48. Chaitrali Amrutkar, Kapil Singh, Arunabh Verma, and Patrick Traynor. VulnerableMe: Measuring Systemic Weaknesses in Mobile Browser Security. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2012. (Acceptance rate: 25%) (Best Paper - SAIC Student Paper Competition (GT)) (Finalist - CSAW AT&T Applied Security Research Best Paper Competition 2012).
49. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. A Measurement Study of SSL Indicators on Mobile Browsers: Extended Life, or End of the Road? In *Proceedings of the Information Security Conference (ISC)*, 2012. (Acceptance rate: 32%) (Best Student Paper).
50. Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, 2012. (Acceptance Rate: 20.2%).
51. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2012. (Acceptance Rate: 17.8%).
52. Yacin Nadji, Jon Giffin, and Patrick Traynor. Automated Remote Repair for Mobile Malware. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).
53. Nilesh Nipane, Italo Dacosta, and Patrick Traynor. "Mix-In-Place" Anonymous Networking Using Secure Function Evaluation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).
54. Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2011. (Acceptance Rate: 13.9%).
55. Philip Marquardt, David Dagon, and Patrick Traynor. Impeding Individual User Profiling in Shopper Loyalty Programs. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*, 2011. (Acceptance Rate: 35.1%).
56. David Dewey and Patrick Traynor. No Loitering: Exploiting Lingering Vulnerabilities in Default COM Objects. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2011. (Acceptance Rate: 20.1%).
57. Vijay Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael Hunter, and Patrick Traynor. PinDrOp: Using Single-Ended Audio Features to Determine Call Provenance. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2010. (Acceptance Rate: 17.2%).
58. Patrick Traynor, Joshua Schiffman, Thomas La Porta, Patrick McDaniel, Abhrajit Ghosh, and Farooq Anjum. Constructing Secure Localization Systems with Adjustable Granularity. In *IEEE Global Communications Conference (GLOBECOM)*, 2010. (Acceptance Rate: 35.6%).
59. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2010. (Acceptance Rate: 25.0%).
60. Kapil Singh, Samrit Sangal, Nehil Jain, Patrick Traynor, and Wenke Lee. Evaluating Bluetooth as a Medium for Botnet Command and Control. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2010. (Acceptance Rate: 30.7%).

61. Italo Dacosta and Patrick Traynor. Proxychain: Developing a Robust and Efficient Authentication Infrastructure for Carrier-Scale VoIP Networks. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2010. (Acceptance Rate: 17.0%).
62. Frank S. Park, Chinmay Gangakhedkar, and Patrick Traynor. Leveraging Cellular Infrastructure to Improve Fraud Prevention. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2009. (Acceptance Rate: 19.0%).
63. Patrick Traynor, Michael Lin, Machigar Ongtang, Vikyath Rao, Trent Jaeger, Thomas La Porta, and Patrick McDaniel. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).
64. Brendan Dolan-Gavitt, Abhinav Srivastava, Patrick Traynor, and Jonathon Giffin. Robust Signatures for Kernel Data Structures. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).
65. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. In *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2009. (Acceptance Rate: 43.3%).
66. Patrick Traynor, Kevin Butler, William Enck, and Patrick McDaniel. Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2008. (Acceptance Rate: 17.7%).
67. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. On Attack Causality in Internet-Connected Cellular Networks. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2007. (Acceptance Rate: 12.3%).
68. Sunam Ryu, Kevin Butler, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. In *Proceedings of the IEEE International Symposium on Security in Networks and Distributed Systems (SSNDS)*, 2007. (Acceptance Rate: 40%).
69. Luke St. Clair, Lisa Johansen, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Trent Jaeger. Password Exhaustion: Predicting the End of Password Usefulness. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2006. (Invited Paper).
70. Kevin Butler, William Enck, Jennifer Plasterr, Patrick Traynor, and P. McDaniel. Privacy-Preserving Web-Based Email. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, December 2006. (Acceptance Rate: 30.4%).
71. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. In *Proceedings of the Thirteenth ACM Conference on Computer and Communications Security (CCS)*, November 2006. (Acceptance Rate: 14.8%).
72. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2006. (Acceptance Rate: 11.7%).
73. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, August 2006. (Acceptance Rate: 25.4%).

74. Patrick Traynor, JaeShung Shin, Barat Madan, Shashi Phoha, and Thomas La Porta. Efficient Group Mobility for Heterogeneous Sensor Networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*, September 2006. (Acceptance Rate: 58%).
75. Patrick Traynor, Raju Kumar, Hussain Bin Saad, Guohong Cao, and Thomas La Porta. LIGER: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. In *Proceedings of the 4th ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2006. (Acceptance Rate: 15.4%).
76. Patrick Traynor, Guohong Cao, and Thomas La Porta. The Effects of Probabilistic Key Management on Secure Routing in Sensor Networks. In *Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2006. (Acceptance Rate: 38.8%).
77. Patrick Traynor, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. Establishing Pair-Wise Keys In Heterogeneous Sensor Networks. In *Proceedings of the 25th Annual IEEE Conference on Computer Communications (INFOCOM)*, April 2006. (Acceptance Rate: 18%).
78. William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth ACM Conference on Computer and Communications Security (CCS)*, November 2005. (Acceptance Rate: 15%).

Removed for external version.

E.2. Conference Presentations with Proceedings (Non-Refereed)

None.

E.3. Conference Presentations without Proceedings

1. Patrick Traynor. Work in Progress Presentations: Fine-Grained Secure Localization for 802.11 Networks. 15th USENIX Security Symposium (SECURITY), August 2006.
2. Patrick Traynor. Work in Progress Presentations: Fundamental Limitations of Sensor Network Security. ACM/USENIX Fourth International Conference on Mobile Systems Applications and Services (MobiSys), June 2006. (Award: Most Entertaining WIP).
3. Patrick Traynor, Heesook Choi, Guohong Cao, and Thomas La Porta. Poster Session: Probabilistic Unbalanced Key Distribution and Its Effects on Distributed Sensor Networks. Workshop on Wireless Security (WiSe), October 2004.

F. Other

F.1. Submitted Journal Papers

None.

F.2. Refereed Research Reports

None.

F.3. Software

1. *GSM Air Interface Simulator*: Developed a full voice, data and SMS capable simulator for the wireless portion of a GSM network. Models communications down to the timeslot for highest possible accuracy. Used in the majority of our work on cellular security.

2. *Malicious Telephony Load Tester*: Built a system on top of the TM1 Telecom Database testing suite to allow for a comparison of malicious traffic of varying composition.

F.4. Published Papers (Non-Refereed)

1. Patrick Traynor. Characterizing the Limitations of Third-Party EAS Over Cellular Text Messaging Services. Technical report, 3G Americas Whitepaper, 2008.
2. Lisa Johansen, Kevin Butler, William Enck, Patrick Traynor, and Patrick McDaniel. Grains of SANs: Building Storage Area Networks from Memory Spots. Technical Report NAS-TR-0060-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, 2007.

F.5. Books in Preparation

None.

F.6. Workshops and External Courses

1. Luis Vargas, Patrick Emami, and Patrick Traynor. On the Detection of Disinformation Campaign Activity with Network Analysis. In *Proceedings of the 2020 ACM SIGSAC Cloud Computing Security Workshop, CCSW '20*, 2020.
2. Siddhant Deshmukh, Henry Carter, Grant Hernandez, Patrick Traynor, and Kevin Butler. Efficient and Secure Template Blinding for Biometric Authentication. In *IEEE Workshop on Security and Privacy in the Cloud (SPC)*, 2016.
3. Debayan Gupta, Benjamin Mood, Joan Feigenbaum, Kevin Butler, and Patrick Traynor. Using Intel Software Guard Extensions for Efficient Two-Party Secure Function Evaluation. In *Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC)*, 2016.
4. Chaitrali Amrutkar and Patrick Traynor. Rethinking Permissions for Mobile Web Apps: Barriers and the Road Ahead. In *Proceedings of the ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, 2012.
5. Nigel Lawrence and Patrick Traynor. Under New Management: Practical Attacks on SNMPv3. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, 2012.
6. Andrew Harris, Frank S. Park, Seymour Goodman, and Patrick Traynor. Emerging Privacy Concerns for Digital Wallet Deployment. In *Proceedings of the Workshop on Making Privacy in America*, 2009.
7. Patrick Traynor. Privacy and Security Concerns for Personal and Mobile Health Devices. In *Proceedings of the Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies*, 2009.
8. Kevin Butler, William Enck, Harri Hursti, Stephen McLaughlin, Patrick Traynor, and Patrick McDaniel. Systemic Issues in the Hart InterCivic and Premier Voting System: Reflections Following Project EVEREST. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology (EVT) Workshop*, 2008.

G. Research Proposals and Grants (Principal Investigator)

1. Approved and Funded

1. Testing Audio Deep Fake Detectors

Sponsor: Bank of America

Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$274,000 over 2 years

Awarded: August 2021

2. Deploying Defenses for Cellular Networks Using the AWARE Testbed

Sponsor: Department of Homeland Security: CISA:

Investigator(s): Patrick Traynor (PI), Kevin Butler, Guofei Gu, Radu Stoleru, Walter Magnussen, P. R. Kumar

Amount: \$3,100,000 over 4 years

Awarded: October 2019

3. SaTC: CORE: Medium: Securing the Voice Processing Pipeline Against Adversarial Audio

Sponsor: NSF Secure and Trustworthy Cyberspace

Investigator(s): Patrick Traynor (PI), Thomas Shrimpton, Vincent Bindschaedler

Amount: \$1,199,999 over 4 years

Awarded: October 2019

4. Artus Protocol STTR Phase II

Sponsor: Office of Naval Research

Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$800,000 over 4 years

Awarded: August 2019

5. Evaluating the Security of QR Code-Based Payments

Sponsor: Discover Financial

Investigator(s): Patrick Traynor (PI)

Amount: \$50,000 over 1 year

Awarded: September 2018

6. Workshop: Addressing the Technical Security Challenges of Emerging Digital Financial Services

Sponsor: NSF Secure and Trustworthy Cyberspace

Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$50,000 over 1 year

Awarded: September 2017

7. Designing Strong End-to-End Authentication Mechanisms for Modern Telephony Systems

Sponsor: NSF Secure and Trustworthy Cyberspace

Investigator(s): Patrick Traynor (PI)

Amount: \$500,000 over 3 years

Awarded: July 2016

8. Digital Healthcare-Associated Infection: Measurement, Defense and Prevention in a Modern Digital Healthcare Ecosystem

Sponsor: National Science Foundation

Investigator(s): Patrick Traynor (PI), Kevin Butler, Shigang Chen

Amount: \$1,200,000 over 4 years

Awarded: June 2016

9. **Evaluating and Improving Security in Emerging Branchless Banking Systems**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI)
Amount: \$500,000 over 3 years
Awarded July 2015
10. **Prevention and Detection of Disallowed Connections in Mobile and Pervasive Systems**
Sponsor: CISE-ECE Harris Endowed Seed Fund Program
Investigator(s): Patrick Traynor (PI), Renato Figueiredo (PI)
Amount: \$40,000 over 1 year
Awarded December 2014
11. **Mobile Excursion Study Support**
Sponsor: Hanscom AFB Electronic Systems Command Development Planning Division (ESC/XR)
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad, Jeff Evans, Chuck Bokath
Amount: \$280,000 over 3 months
Awarded July 2012
12. **Characterizing the Security Limitations of Accessing the Mobile Web**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI) and William Enck (NC State)
Amount: \$334,000 over 3 years
Awarded July 2012
13. **Mitigating Attacks on Mobile Devices and Critical Cellular Infrastructure**
Sponsor: US Department of Defense - Defense University Research Instrumentation Program (DURIP)
Investigator(s): Patrick Traynor (PI), Jon Giffin, Mustaque Ahamad
Amount: \$210,081 over 1 year
Awarded June 2011
14. **Characterizing and Implementing Efficient Primitives for Privacy-Preserving Computation**
Sponsor: DARPA PROgramming Computation on EncryptEd Data (PROCEED) – Broad Agency Announcement
Investigator(s): Patrick Traynor (PI) and Kevin Butler (UOregon)
Amount: \$580,000 over 4 years
Awarded May 2011
15. **Security for Converged IMS Networks**
Sponsor: US Department of Defense
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad and Russ Clark
Amount: \$242,401 over 1 year
Awarded August 2010
16. **CAREER: Protecting User Data on Lost, Stolen and Damaged Mobile Phones**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: \$400,000 over 5 years
Awarded: May 2010
17. **Provably Anonymous Networking Through Secure Function Evaluation**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: \$200,000 over 2 years
Awarded: July 2009

18. **Characterizing and Mitigating Device-Based Attacks in Cellular Telecommunications Networks**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI) and Jonathon Giffin
Amount: \$450,000 over 3 years
Awarded: July 2009

2. Pending

Removed for external version.

H. Research Proposals and Grants (Contributor)

1. Approved and Funded

1. **SaTC: Frontier: Securing the Future of Computing for Marginalized and Vulnerable Populations**
Sponsor: NSF SaTC
Investigator(s): Kevin Butler (PI), Patrick Traynor, Tadayoshi Kohno, Franzi Roesner, Apu Kapadia, Eakta Jain.
Amount: \$7,500,000 for 5 years
Awarded October 2022
2. **ROCKY: Reliable Obfuscated Communications Kit for everYone**
Sponsor: DARPA Resilient Anonymous Communication for Everyone (RACE) – Broad Agency Announcement
Investigator(s): Thomas Shrimpton (PI), Patrick Traynor, Kevin Butler, Vincent Bindschaedler, Nadia Heninger
Amount: \$1,600,000 over 4 years
Awarded May 2019
3. **WiFiUS: Collaborative Research: SELIOT: Securing Lifecycle of Internet-of-Things**
Sponsor: NSF CNS WiFiUS
Investigator(s): Gene Tsudik (PI), Patrick Traynor
Amount: \$300,000 for 2 years
Submitted December 2016
4. **Cloud-based Oblivious Spectrum Mapping and Allocation**
Sponsor: NSF CNS EARS
Investigator(s): John Shea (PI), Tan Wong, Patrick Traynor
Amount: \$532,952 for 2 years
Submitted May 2016
5. **DURIP: Developing Research Capability in Cyber-Physical Systems at the University of Florida**
Sponsor: Small
Investigator(s): Kevin Butler (PI), Patrick Traynor, My Thai
Amount: \$200,000 for 2 years
Submitted: June 2015
6. **Securing the New Converged Telephony Landscape**
Sponsor: NSF TWC: Small
Investigator(s): Mustaque Ahamad (PI) and Patrick Traynor
Amount: \$500,000 for 3 years
Submitted: December 2012

7. **Facilitating Free and Open Access to Information on the Internet**
 Sponsor: NSF Trustworthy Computing
 Investigator(s): Nick Feamster (PI), Wenke Lee, Patrick Traynor, Hans Klein, Roger Dingledine, Michael Freedman and Edward W. Felten
 Amount: \$1,500,000 for 4 years
 Awarded: June 2011
8. **Monitoring Free and Open Access to Information on the Internet**
 Sponsor: Google Focus Program
 Investigator(s): Nick Feamster (PI), Wenke Lee, Mustaque Ahamad, Patrick Traynor, Henry Owen, Ellen Zegura, Zvi Galil
 Amount: \$1,000,000 for 2 years
 Awarded: November 2011
9. **Dynamic-attribute-based Disclosure of Health Information in Emergency Care Scenarios**
 Sponsor: Health Systems Institute (HSI) Seed Grant Program
 Investigator(s): Doug Blough (PI), Mustaque Ahamad, Patrick Traynor and Jim Jose
 Amount: \$50,000 over 1 year
 Awarded: August 2009
10. **Federal Cyber Service Scholarships at Georgia Tech**
 Sponsor: NSF SFS Scholarships
 Investigator(s): Seymour Goodman (PI), Patrick Traynor
 Amount: \$1,250,682 over 5 years
 Awarded: June 2009
11. **Security for IMS-Enabled Converged Applications**
 Sponsor: US Department of Defense
 Investigator(s): Mustaque Ahamad (PI), Patrick Traynor (PI), Michael Hunter, Russ Clark
 Amount: \$146,121 for 1 year
 Awarded: August 2008

2. Pending

Removed for external version.

I. Research Honors and Awards

1. Fellow, Center for Financial Inclusion at Accion, 2017.
2. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.
3. Best Paper, The ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec); Budapest, Hungary, 2013.
4. Best Student Paper, The Information Security Conference (ISC); Passau, Germany, 2012
5. Lockheed Inspirational Young Faculty Award, 2012
6. Best Demo, "Is Browsing the Internet on Your Mobile Phone Secure?" Chaitrali Amrutkar (Ph.D Advisee), CoC Research Day, 2011
7. Best Poster, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers" Arunabh Verma, Henry Carter (MS, Ph.D Advisees), CoC Research Day, 2011
8. National Science Foundation CAREER Award, 2010

9. Pennsylvania State University Alumni Association Dissertation Award, 2007
10. Pennsylvania State University CSE Graduate Research Assistant Award, 2007
11. AT&T Wireless Fellowship, 2005

III. SERVICE

A. Professional Activities

A.1. Memberships and Activities in Professional Societies

1. Senior Member, Association for Computing Machinery (ACM)
2. Senior Member, Institute of Electrical and Electronics Engineers (IEEE)
3. Member, USENIX Advanced Computing Systems Association (USENIX)

A.2. Conference Committee Activities

1. Program co-Chair, *IEEE Symposium on Security and Privacy (OAKLAND)*: 2023, 2024
2. Program co-Chair, *USENIX Security Symposium (SECURITY)*: 2019
3. Program co-Chair, *Network and Distributed System Security Symposium (NDSS)*: 2017, 2018
4. Program Chair, *USENIX Workshop on Offensive Technologies (WOOT)*: 2016
5. Program Chair, *ACM Conference on Wireless Network Security (WiSec)*: 2014
6. Program Co-Chair, *Annual Computer Security Applications Conference (ACSAC)*: 2012, 2013
7. Program Chair, *USENIX Workshop on Hot Topics in Security (HotSec)*: 2012
8. Chair Invited Talks Committee, *USENIX Security Symposium (SECURITY)*: 2014
9. Workshops Chair, *IEEE Conference on Communications and Network Security (CNS)*: 2016
10. Program Committee, *USENIX Security Symposium (SECURITY)*: 2008, 2009, 2010, 2013, 2015-2018, 2020-2022
11. Program Committee, *IEEE Symposium on Security and Privacy (OAKLAND)*: 2009-2014, 2022.
12. Program Committee, *ACM Conference On Computer and Communications Security (CCS)*: 2009, 2013-2015, 2017
13. Program Committee, *Network and Distributed System Security Symposium (NDSS)*: 2010, 2013-2016, 2020-2021
14. Program Committee, *IEEE European Symposium on Security and Privacy (Euro S&P)*: 2016
15. Program Committee, *Annual Computer Security Applications Conference (ACSAC)*: 2008, 2009, 2010, 2011, 2015
16. Program Committee, *ACM Conference on Wireless Network Security (WiSec)*: 2009, 2010, 2013, 2015-2021
17. Program Committee, *International Conference on Financial Cryptography and Data Security (FC)*: 2010, 2013
18. Program Committee, *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*: 2016.
19. Program Committee, *ICST Conference on Security and Privacy in Communication Networks (SecureComm)*: 2009, 2010
20. Program Committee, *Privacy Enhancing Technologies Symposium (PETS)*: 2015, 2016

21. Program Committee, *International World Wide Web Conference (WWW)*: 2016
22. Program Committee, *USENIX Workshop on Hot Topics in Security (HotSec)*: 2011
23. Program Committee, *ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MOBIHELD)*: 2010
24. Program Committee, *International Workshop on Mobile Security (WMS)*: 2010
25. Program Committee, *European Symposium on Research in Computer Security (ESORICS)*: 2009, 2011
26. Program Committee, *IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS)*: 2009, 2010
27. Program Committee, *Information Security Conference (ISC)*: 2010
28. Program Committee, *IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT)*: 2009
29. Program Committee, *Computer Security Architecture Workshop (CSAW)*: 2008
30. Program Committee, *IWCMC Computer and Network Security Symposium*: 2009
31. Program Committee, *IARIA International Conference on Internet Monitoring and Protection (ICIMP)*: 2009
32. Program Committee, *IEEE Workshop on Network Security and Privacy (NSP)*: 2008
33. Program Committee, *IEEE International Workshop on Wireless and Sensor Networks Security (WSNS)*: 2008, 2009
34. Program Committee, *IEEE Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*: 2008
35. Program Committee, *ACM Conference on Computer and Communications Security, Industry and Government Track (CCS I&G)*: 2006, 2007
36. Program Committee, *Workshop on Secure Network Protocols (NPSec)*: 2006
37. Program Committee, *International Conference on Information Systems Security (ICISS)*: 2006, 2009, 2010
38. Program Committee, *IEEE LCN Workshop on Network Security (WNS)*: 2006, 2007, 2008

B. On-Campus Committees

B.1. University of Florida

1. Member, Computer and Information Science and Engineering Steering Committee, 2015-2017.
2. Member, Graduate Recruiting Committee, 2015-2017.
3. Chair, Computer and Information Science and Engineering Industrial Advisory Board, 2014-2015.

B.2. Georgia Tech

1. Member, Massive Open Online Master's (MOOMS) Investigation Committee, 2012-2013.
2. Chair, School of Computer Science Ph.D. Review Committee, 2012.
3. Member, School of Computer Science Ph.D Review Committee, 2011.
4. Faculty Advisor, Grey H@T - Georgia Tech Undergraduate Security Club, 2011-2014.
5. Member, School of Computer Science Ph.D. Review Committee, 2011.
6. Member, School Advisory Committee, School of Computer Science, 2011-2013.
7. Member, School of Computer Science Chair Recruiting Committee, 2011.
8. Member, School of Computer Science Faculty Recruiting Committee, 2010, 2011.
9. Chair, College of Computing Ph.D. Welcome Weekend Committee, 2009, 2010, 2011 (co-chair).
10. Member, College of Computing Ph.D. Recruiting Committee, 2009.
11. Member, Georgia Tech Computer and Network Usage Security Policy (CNUSP) Evaluation Group, 2009.

C. Special Assignments

None.

D. Ph.D. Examining Committees

Ph.D. Examining Committees

1. Bradley Reaves, Department of Computer and Information Science and Engineering, University of Florida, Summer 2017.
Advisor: Professor Patrick Traynor.
2. Adam Bates, Department of Computer and Information Science and Engineering, University of Florida, Spring 2016.
Advisor: Professor Kevin Butler.
3. Benjamin Mood, Department of Computer and Information Science and Engineering, University of Florida, Spring 2016.
Advisor: Professor Kevin Butler.
4. Henry Carter, College of Computing, Georgia Tech, Fall 2015.
Advisor: Professor Patrick Traynor.
5. David Dewey, College of Computing, Georgia Tech, Fall 2015.
Advisor: Professor Patrick Traynor.
6. Lateef Yusuf, College of Computing, Georgia Tech, Spring 2014.
Advisor: Professor Umakishore Ramachandran.
7. Chaitrali Amrutkar, College of Computing, Georgia Tech, Summer 2013.
Advisor: Professor Patrick Traynor.
8. Long Lu, College of Computing, Georgia Tech, Summer 2013.
Advisor: Professor Wenke Lee.

9. Manos Antonakakis, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Wenke Lee.
10. Junjie Zhang, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Wenke Lee.
11. Italo Dacosta, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Patrick Traynor.
12. Virendra Kumar, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Alexandra Boldyreva.
13. Anirudh Ramachandran, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Nick Feamster.
14. Vijay Balasubramaniyan, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Mustaque Ahamad.
15. Kapil Singh, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Wenke Lee.
16. Abhinav Srivastava, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Jon Giffin.
17. Adam O'Neill, College of Computing, Georgia Tech, Summer 2010.
Advisor: Professor Alexandra Boldyreva.
18. David Cash, College of Computing, Georgia Tech, Fall 2009.
Advisor: Professor Alexandra Boldyreva.

External Member of Ph.D. Research Committee

None.

External Member of Ph.D. Examining Committee

1. Shannon Eggers, Department of Materials Sciences and Engineering - Nuclear Engineering Program, University of Florida, Fall 2016.
Advisor: Professor Kelly Jordan.
2. Ed Carlisle, Department of Electrical and Computer Engineering, University of Florida, Summer 2016.
Advisor: Professor Alan George.
3. Claudio Marforio, Department of Computer Science, Swiss Federal Institute of Technology Zurich (ETH Zurich), Fall 2015.
Advisor: Professor Srdjan Capkun.
4. Nils Ole Tippenhauer, Department of Computer Science, Swiss Federal Institute of Technology Zurich (ETH Zurich), Spring 2012.
Advisor: Professor Srdjan Capkun.
5. Bongkyoung Kwon, School of Electrical and Computer Engineering, Georgia Tech, Summer 2009.
Advisor: Professor John Copeland.

Ph.D. Thesis Proposal Committees

1. Bradley Reaves, Department of Computer and Information Science and Engineering, Spring 2016.
Advisor: Professor Patrick Traynor.
2. Maliheh Shirvanian, University of Alabama, Birmingham, Spring 2016.
Advisor: Professor Nitesh Saxena.
3. Benjamin Mood, Department of Computer and Information Science and Engineering, Fall 2015.
Advisor: Professor Kevin Butler.
4. Adam Bates, Department of Computer and Information Science and Engineering, Fall 2015.
Advisor: Professor Kevin Butler.
5. Lateef Yusuf, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Umakishore Ramachandran.
6. Long Lu, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.
7. Chaitrali Amrutkar, College of Computing, Georgia Tech, Fall 2012.
Advisor: Professor Patrick Traynor.
8. Junjie Zhang, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Wenke Lee.
9. Italo Dacosta, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Patrick Traynor.
10. Manos Antonakakis, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Wenke Lee.
11. Abhinav Srivastava, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Jon Giffin.
12. Vijay Balasubramaniyan, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Mustaque Ahamad.
13. Kapil Singh, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Wenke Lee.
14. Anirudh Ramachandran, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Nick Feamster.
15. Adam O'Neill, College of Computing, Georgia Tech, Spring 2010.
Advisor: Professor Alexandra Boldyreva.
16. David Cash, College of Computing, Georgia Tech, Spring 2009.
Advisor: Professor Alexandra Boldyreva.

Ph.D. Qualifying Exam Committees—Georgia Tech

1. Byoungyoung Lee, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.
2. Yizheng Chen, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.

3. Xinyu Xing, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.
4. Brad Reaves, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Patrick Traynor.
5. Chaz Lever, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Patrick Traynor.
6. Terry Nelms, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professors Mustaque Ahamad and Roberto Perdesi.
7. Saurabh Chakradeo, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professor Patrick Traynor.
8. Henry Carter, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professor Patrick Traynor.
9. David Dewey, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professor Jon Giffin.
10. Chaitrali Amrutkar, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Patrick Traynor.
11. Yacin Nadji, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Wenke Lee.
12. Yogesh Mundada, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Nick Feamster.
13. Hyojoon Kim, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Nick Feamster.
14. Ikpeme Erete, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Alex Orso.
15. Chaitrali Amrutkar, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Patrick Traynor.
16. Brendan Dolan-Gavitt, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Wenke Lee and Professor Jon Giffin.
17. Sam Burnett, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Nick Feamster.
18. Cong Shi, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Mostafa Ammar and Professor Ellen Zegura.
19. Partha Kanuparth, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Constantine Dorvolis.
20. Long Lu, College of Computing, Georgia Tech, Spring 2010.
Advisor: Professor Wenke Lee.
21. Virendra Kumar, College of Computing, Georgia Tech, Spring 2009.
Advisor: Professor Alexandra Boldyreva.
22. Frank Park, College of Computing, Georgia Tech, Spring 2009.
Advisor: Professor Patrick Traynor.

23. Italo Dacosta, College of Computing, Georgia Tech, Fall 2008.
Advisor: Professor Mustaque Ahamad and Professor Patrick Traynor.
24. Adam O'Neill, College of Computing, Georgia Tech, Fall 2008.
Advisor: Professor Alexandra Boldyreva.

E. External Member of M.S. Examining Committee

M.S. Thesis Defense Committees None.

F. Consulting and Advisory Appointments

1. Skim Reaper, *Co-Founder and CEO*, 2019-Present.
2. CryptoDrop Anti-Ransomware, *Co-Founder and CEO*, 2017-2018.
3. Pindrop Security, *Research Fellow and Co-Founder*, Spring 2012 - Spring 2014.
4. United States Army (via US Falcon), *Information Assurance Officer Training Program*, Spring 2010.
5. 3G Americas, *Characterizing the Limitations of Third-Party EAS over Cellular Text Messaging Systems*, Fall 2008.

G. Civic Activities

None.

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION

A. Honors and Awards

1. Fellow, Kavli Foundation, 2017.
2. Fellow, Center for Financial Inclusion at Accion, 2016.
3. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.

B. Invited Conference Session Chairmanships

1. Session Chair, *Work-in-Progress* at the *USENIX Security Symposium (SECURITY)*, 2016.
2. Session Chair, *Mobile Security* at the *USENIX Security Symposium (SECURITY)*, 2013.
3. Poster Chair, *USENIX Security Symposium (SECURITY)*, 2010, 2011.
4. Session Chair, *Privacy and Anonymity* at the *USENIX Workshop on Hot Topics in Security (HotSec)*, 2011.
5. Session Chair, *Security of Authentication and Protection Mechanisms* at the *IEEE Symposium on Security & Privacy (OAKLAND)*, 2011.
6. Session Chair, *Information Abuse* at the *IEEE Symposium on Security & Privacy (OAKLAND)*, 2010.
7. Session Chair, *RFID Security* at the *ACM Conference on Computer and Communications Security (CCS)*, 2009.
8. Session Chair, *Browser Security Session* at the *USENIX Security Symposium (SECURITY)*, 2009.
9. Session Chair, *Information Security Session* at the *IEEE Symposium on Security and Privacy (OAKLAND)*, 2009.
10. Session Chair, *Work-in-Progress* at the *IEEE Symposium on Security and Privacy (OAKLAND)*, 2009.
11. Session Chair, *Work/Opinions-in-Progress* at the *ISOC Network and Distributed Systems Security (NDSS) Symposium*, 2009.
12. Session Chair, *Privacy Session* at the *USENIX Security Symposium (SECURITY)*, 2008.

C. Professional Registration

None.

D. Patents

1. Patrick G. Traynor, Christian Peeters, Bradley G. Reaves, Hadi Abdullah, Kevin Butler, Jasmine Bowers, Walter N. Scaife, "Detecting SS7 Redirection Attacks With Audio-Based Distance Bounding", United State Patent # 11,265,717, Filed March 2019, Issued March 2022.
2. Patrick G. Traynor, Logan E. Blue, Luis Vargas, "Method and Apparatus for Differentiating Between Human and Electronic Speaker for Voice Interface Security", United State Patent # 11,176,960, Filed June 2019, Issued November 2021.
3. Patrick G. Traynor, Bradley G. Reaves, Logan E. Blue Practical End-to-End Cryptographic Authentication for Telephony Over Voice Channels, United State Patent # 11,329,831, Filed November 2018, Issued May 2022.

4. Walter Nolen Scaife, Patrick G. Traynor and Christian Peeters, "Payment Card Overlay Skimmer Detection", United States Patent # 10,496,914, Filed October 2017, Issued December 2019. (See also # 10,936,928)
5. Patrick G. Traynor, David P. Arnold, Walter Nolen Scaife, Christian Peeters, and Camilo Valez Cuervo, "Detecting counterfeit magnetic stripe cards using encoding jitter", United States Patent # 10,803,261, Filed May 2017, Issued October 2020.
6. Patrick G. Traynor, Bradley Reaves, Logan Blue, Luis Vargas, Hadi Abdullah, and Thomas Shrimpton, "Identity and content authentication for phone calls", United States Patent # 10,764,043, Filed Apr 2017, Issued September 2020.
7. Walter Nolen Scaife, Henry Carter, Patrick G. Traynor and Kevin R. B. Butler. "Malware Detection Through User Data Transformation Monitoring", United States Patent # 10,685,114. Filed September 2015, Issued June 2020.
8. Vijay A. Balasubramaniyan, Mustaque Ahamad, Patrick G. Traynor. "Using Single-Ended Audio Features to Automatically Determine Voice Call Provenance", United States Patent, #9,037,113 June 2010, Issued May 2015. (See also #9,516,497 and #10,523,809)
9. Patrick G. Traynor, Byungsuk Kim and Farooq Anjum. "Secure Localization for 802.11 Networks with Fine Granularity", United States Patent, #8,107,400, Filed July 2008, Issued January 2012.

E. Editorial and Reviewer Work for Technical Journals and Publishers

Associate Editor:

- *ACM Transactions on Information and System Security (TISSEC)* 2015-present

Guest Editor:

Journals

- *IEEE Security and Privacy Magazine (S&P)* 2013

Reviewer for:

Journals

- *ACM Transactions on Information and System Security (TISSEC)* 2008, 2009, 2010, 2011, 2012, 2013
- *IEEE Transactions on Dependable and Secure Computing (TDSC)* 2012, 2013
- *IEEE Security and Privacy Magazine (S&P)* 2010, 2011
- *Communications of the ACM (CACM)* 2010
- *Journal of Anesthesia & Analgesia* 2009
- *IEEE Transactions on Mobile Computing (TMC)* 2008, 2010, 2011, 2012, 2013
- *IEEE Transactions on Internet Technology (TOIT)* 2009, 2010
- *ACM Mobile Computing and Communications Review (MC2R)* 2008
- *IEEE/ACM Transactions on Networking (TON)* 2007, 2008
- *Journal of Pervasive and Mobile Computing (PMC)* 2009, 2010

- *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 2005, 2009, 2010
- *IEEE Transactions on Computers (TOC)* 2010
- *Journal of Security and Communication Networks (SCN)* 2008
- *IEEE Communications Letters (CL)* 2007, 2009
- *IEEE Transactions on Wireless Communications (TWC)* 2007
- *Pervasive and Mobile Computing (PMC)* 2007
- *IEEE Transactions on Software Engineering (TSE)* 2007, 2008
- *Journal of Wireless Networks (WiNet)* 2006, 2007, 2008, 2009
- *Journal of Wireless Communications and Mobile Computing* 2006
- *ACM Computing Surveys (ACMCS)* 2006
- *Information Processing Letters (IPL)* 2006
- *IEEE Transactions on Very Large Scale Integration Systems (TVLSIS)* 2006

Conferences and Workshops

- *ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2011
- *ACM Conference on Computer and Communications Security (CCS)*, 2008, 2011
- *IEEE Symposium on Security and Privacy (OAKLAND)* 2007, 2008
- *Computer Security Foundations (CSF)*, 2011
- *IFIP Conference on Data and Applications Security (DBSec)* 2008
- *Financial Cryptography (FC)* 2007, 2008
- *International Conference on VLSI Design (VLSI)* 2007
- *Annual Computer Security Applications Conference (ACSAC)* 2005, 2006, 2007
- *USENIX Workshop on Hot Topics in Security (HotSec)* 2007
- *International Conference on Information Systems Security (ICISS)* 2007
- *IEEE International Conference on Computer Engineering & Systems (ICCES)* 2007
- *International Workshop on Security (IWSec)* 2006, 2007
- *USENIX Security Symposium (SECURITY)* 2006, 2007
- *IEEE Sarnoff Symposium (SARNOFF)* 2007
- *International Conference on New Technologies, Mobility and Security (NTMS)* 2007
- *IEEE Infocom (INFOCOM)* 2007
- *Network and Distributed System Security Symposium (NDSS)* 2007
- *International Workshop on Storage Security and Survivability (IWSSS)* 2006
- *ACM Conference on Computer and Communications Security (CCS)* 2006

- *IEEE GLOBECOM (GLOBECOM) 2006*
- *International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS) 2006*
- *IFIP Conference on Data and Applications Security (DBSec) 2006*
- *Emerging Trends in Information and Communications Security (ETRICS) 2006*
- *International Conference on Applied Cryptography and Network Security (ACNS) 2006*
- *ACM Symposium on Access Control Models and Technology (SACMAT) 2006*
- *IEEE Conference on Communication Systems Software & Middleware (COMSWARE) 2006*
- *International Conference on Cryptology in India (IndoCrypt) 2005*
- *IEEE Symposium on New Frontiers in Dynamic Spectrum Access (DySPAN) 2005*
- *European Symposium on Research in Computer Security (ESORICS) 2005*

F. Expert Witness Services

1. *Epic Games, Inc. & Anor v Google LLC & Ors - Federal Court of Australia Proceeding NSD 190 of 2021:* Expert witness for the Defense (via Corrs Chambers Westgarth). Status: Ongoing. *January 2023 - Present.*
2. *Telefonaktiebolaget LM Ericsson vs Apple, Inc:* Expert witness for the Defendant, Non-Infringement and Invalidity (via WilmerHale LLP). Status: Settled. *February 2022 - December 2022.*
3. *Wepay Global Payments, LLC v. Bank of America N. A.:* Expert Witness for the Defendant (via WilmerHale LLP) Status: Dismissed. *September 2022 - November 2022.*
4. *Apple vs. R.N Nehushtan Trust Ltd.:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Status: Submitted to PTAB. *August 2022 - Ongoing.*
5. *Apple/Microsoft vs. Zipit Wireless:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Status: Settled. *May 2021 - November 2022.*
6. *Blackberry Inc v MobileIron, Inc:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Verdict: Settled. *January 2021 - March 2021.*
7. *Apple Inc v Seven Networks, LLC:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Verdict: Three of four petitions instituted by PTAB. Fourth rejected via discretion, case settled. *August 2019 - November 2020.*
8. *mSIGNIA, Inc. v. InAuth, Inc.:* Expert Witness for the Defendant for Inter Partes Review, Non-Infringement and Invalidity (via Quinn Emanuel Urquhart and Sullivan, LLP). Verdict: Dismissed with prejudice. *October 2017 - December 2018.*
9. *Huawei v. T-Mobile:* Expert Witness for the Defendant for Non-Infringement (via WilmerHale LLP, Alston & Bird LLP) Verdict: Settled *June 2016 - December 2017.*
10. *Telefonaktiebolaget LM Ericsson v Apple:* Expert Witness for the Defendant for Non-Infringement, Invalidity (via WilmerHale LLP). Verdict: Settled *June 2015 - December 2015.*
11. *Mayfonk v Nike:* Expert Witness for the Plaintiff for Infringement (via Paul Hastings). Verdict: Settled. *June 2015 - November 2015.*

12. *Maxim Integrated Products v Bank of the West*: Expert Witness for the Defendant for Non-Infringement (via Paul Hastings LLP). Verdict: Dismissed with prejudice. *January 2014 - August 2014*.
13. *Maxim Integrated Products v Comerica Inc, et al*: Expert Witness for the Defendant for Non-Infringement (via McKenna, Long & Aldridge LLP). Verdict: Settled. *June 2014 - August 2014*.
14. *William Grecia v. Apple Inc. et al*: Expert Consultant for the Defendant for Invalidity (via Kirkland & Ellis LLP). Verdict: Closed in initial pleadings, dismissed with prejudice. *July 2014 - August 2014*.
15. *Intertrust Technologies Corp. v. Apple Inc.*: Expert Consultant for Defendant for Invalidity and Non-Infringement (via Kirkland & Ellis LLP). Verdict: Settled. *October 2013 - February 2014*.
16. *Maxim Integrated Products v KeyCorp Bank*: Expert Witness for the Defendant for Non-Infringement (via Calfee, Halter & Griswold LLP) Verdict: Settled. *April 2013 - June 2013*.
17. *Intellectual Ventures LLC vs. Check Point; et al.*: Expert Consultant for the Plaintiff for Infringement (via Susman Godfrey LLP), Verdict: Infringement on 2 of 4 patents. *October 2012 - February 2015*.

V. OTHER CONTRIBUTIONS

A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia)

1. Keynote: Well, It Worked on My Computer: Reproducibility, Tech Transfer, and Computer Security Research. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) Vision 2.0 Workshop, March 2023. University of Texas at Dallas.
2. Humans vs The Computer Interfaces: Separating Deepfakes/Bots from People Using Psychoacoustics. UCLA Electrical and Computer Engineering Distinguished Seminar, February 2023. University of California, Los Angeles.
3. Keynote: Exploiting the Gaps Between Human and Machine Understanding of Audio: Frameworks, Attacks, and Defenses. ISCA Symposium on Security and Privacy in Speech Communication (SPSC), November 2021. Virtual.
4. The State of Voice Cloning Technology. Federal Trade Commission (FTC) Workshop on Voice Cloning Technologies, January 2020. Washington, DC.
5. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. North Carolina State University Department of Computer Science Colloquium, January 2020. Raleigh, NC.
6. Moving from research to practice: How to maximize the impact of SaTC projects. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) PI Meeting, October 2019. Alexandria, VA.
7. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Purdue University Computer Science Excellence Lecture Series, October 2019. West Lafayette, IN.
8. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Bank of America - Colloquium Series, March 2019. Charlotte, NC.
9. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. CISPA – Helmholtz Center for Information Security, Saarland University, February 2019. Saarbrücken, Germany.
10. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. University of Maryland - Distinguished Colloquium, February 2019. College Park, MD.
11. Responsible Finance for the Digital Client. Foromic Conference, October 2018. Barranquilla, Colombia.
12. Panel: Authentication Challenges for New Interfaces, Devices, and Wireless Networks. ACM Conference on Security and Privacy in Wireless and Mobile Networks, June 2018. Stockholm, Sweden.
13. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. CyberSecurity@KAIST Workshop - KAIST, June 2018. Daejeon, South Korea.
14. Why Caller-ID Spoofing Is So Easy (and Why End-To-End Solutions Are the Way Forward). IEEE Workshop on Technology and Consumer Protection (ConPro'18), May 2018. San Francisco, CA.
15. Panel: The Future of Cybersecurity. SEC Academic Conference - Auburn University, May 2018. Auburn, AL.
16. Sound Principles: Verifying Voice Commands in an IoT World. IoT Security Workshop - Aalto University, September 2017. Helsinki, Finland.

17. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Eurecom Institute, September 2017. Sophia Antipolis, France.
18. Panel: Infrastructure Stability. ITU-T Focus Group Digital Financial Services, December 2016. Geneva, Switzerland.
19. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. ETH Zurich, December 2016. Zurich, Switzerland.
20. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. University of Richmond, October 2016. Richmond, Virginia.
21. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Indiana University, September 2016. Bloomington, Indiana.
22. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Aalto University Computer Science Department Forum, August 2016. Helsinki, Finland.
23. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. KAIST Information Security Seminar - Korean Advanced Institute of Science and Technology, June 2016. Daejeon, South Korea.
24. Updated Mobile Money Vulnerability Report. International Telecommunications Union Digital Financial Services Working Group Workshop, May 2016. Washington, DC.
25. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. UF Eye Opener Discovery Breakfast - University of Florida, May 2016. Gainesville, FL.
26. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Illinois Science of Security (SoS) Lablet Speaker Series - University of Illinois, Urbana-Champaign, April 2016. Urbana-Champaign, Illinois.
27. New Trends in Cybersecurity: Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - Cybersecurity and Cybercrime Workshop for Lusophone Africa, September 2015. Maputo, Mozambique.
28. New Trends in Cybersecurity: Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - ECCAS Cybersecurity and Cybercrime Workshop, August 2015. Kinshasa, Democratic Republic of Congo.
29. Chasing Telephony Security: Where the Wild Things... Are? University of Florida - Department Colloquium, January 2014. Gainesville, FL.
30. Chasing Telephony Security: Where the Wild Things... Are? Verizon Wireless RNC/Data Center, October 2013. Alpharetta, GA.
31. Chasing Telephony Security: Where the Wild Things... Are? University of Waterloo - CrySP Speaker Series on Privacy, October 2013. Waterloo, ON, Canada.
32. Analyzing Malicious Traffic in Cellular Networks. GSM Association's (GSMA) Mobile Malware Community Workshop, July 2013. Mountain View, CA.
33. Threats to Mobile Devices. US Federal Trade Commission (FTC) Public Forum - Invited Speaker, June 2013. Washington, D.C.
34. Chasing Telephony Security: Where the Wild Things... Are? University of Wisconsin - Madison, Security Seminar, March 2013. Madison, WI.

35. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2013. Belfast, Northern Ireland.
36. Chasing Telephony Security: Where the Wild Things... Are? Stanford Security Seminar, March 2013. Stanford, CA.
37. Chasing Telephony Security: Where the Wild Things... Are? University of California, Berkeley, Security Group, March 2013. Berkeley, CA.
38. Chasing Telephony Security: Where the Wild Things... Are? Carnegie Mellon University CyLab Seminar, February 2013. Pittsburgh, PA.
39. Chasing Telephony Security: Where the Wild Things... Are? University of Oregon Department of Computer Science Colloquium, November 2012. Eugene, OR.
40. Chasing Telephony Security: Where the Wild Things... Are? University of Washington Department of Electrical Engineering, Network Security Lab (NSL): Invited Talk, November 2012. Seattle, WA.
41. Needles and Haystacks: Digging for Ground Truth on Mobile Malware. ZISC Workshop on Secure Mobile and Cloud Computing, ETH Zurich, June 2012. Zurich, Switzerland.
42. Panel: Advice for Early Career Faculty. CRA Career Mentoring Workshop, February 2012. Washington, D.C.
43. Research Challenges in Cellular and Mobile Network Security. US-China Software Workshop (Co-Sponsored by NSF and NSFC), September 2011. Beijing, China.
44. Mobile Security: Understanding Risks to Critical Infrastructure. Invited Talk: US Department of State East African Workshop on Cyberspace Security, July 2011. Nairobi, Kenya.
45. Tomorrow's Issues: Solving the Mobile Security Threat. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2011. Belfast, Northern Ireland.
46. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. Invited Talk: MITRE Corporation, March 2011. Burlington, MA.
47. Defeating Session Hijacking Attacks with Disposable Web Credentials. Invited Talk: Facebook, February 2011. Palo Alto, CA.
48. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: RSA Conference, February 2011. San Francisco, CA.
49. Panel: Voice Security – Now Just a False Sense of Security and Privacy. Invited Panelist: Mobile Security Symposium, February 2011. San Francisco, CA.
50. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: Concordia University, May 2010. Montreal, QC, Canada.
51. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Qualcomm Research, March 2010. San Diego, CA.
52. Privacy and Security Concerns for Personal and Mobile Health Devices. Invited Talk: Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies, October 2009. Indianapolis, IN.
53. Considerations for EAS Over Cellular Text Messaging Services. 3G Americas Webinar, July 2009.

54. University Telephony Research Panel. Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM), July 2009.
55. The Evolving Mobile Landscape: Emerging Security Threats. Mobile Security eConference, SC Magazine, June 2008.
56. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Washington, February 2009. Seattle, WA.
57. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Microsoft Research, February 2009. Redmond, WA.
58. Next Year's Problems. Secure Computing (SC) Magazine Webinar, November 2008.
59. Panel: Embedded Systems and their Increasing Impact on Infrastructure Security. Workshop on Embedded Systems Security (WESS), October 2008.
60. Can you DoS me now? Security Issues in Cellular Networks. Georgia Institute of Technology, September 2008. Atlanta, GA.
61. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Georgia Institute of Technology, April 2008. Atlanta, GA.
62. Characterizing the Impact of Rigidity on the Security of Cellular Networks. AT&T Research Labs, April 2008. Florham Park, NJ.
63. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Arizona, March 2008. Tucson, AZ.
64. Cellular Networks Security Panel. USENIX Security Symposium, August 2007. Boston, MA.
65. malnets:Large-Scale Malicious Networks via Compromised Access Points. The Pennsylvania State University - ACM Club Invited Speaker, October 2006. State College, PA.
66. malnets:Large-Scale Malicious Networks via Compromised Access Points. The University of Michigan, October 2006. Ann Arbor, MI.
67. Exploiting Open Functionality in SMS-Capable Cellular Networks. The University of Michigan, October 2006. Ann Arbor, MI.
68. Exploiting Open Functionality in SMS-Capable Cellular Networks. High Technology Crime Investigation Association (HTCIA), September 2006. Pittsburgh, PA.
69. Trends in Security: Critical Engineering in the Large. Schlumberger Innovate IT! Workshop, May 2006. Cambridge, MA.
70. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.
71. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.

B. Special Activities

Presentations to Lay Media

1. How technology can fight digital fakery. The Babbage Podcast/The Economist <https://shows.acast.com/theeconomistbabbage/episodes/babbage-how-to-detect-a-deepfake>, January 2023.
2. Deepfake audio has a tell and researchers can spot it. Ars Technica <https://arstechnica.com/information-technology/2022/09/researchers-use-fluid-dynamics-to-spot-deepfake-voices/>, September 2022.
3. This security tool could help stop the problem of ransomware in its tracks. TheJournal.ie <https://www.thejournal.ie/ransomware-researchers-stop-2875032-Jul2016/>, July 2016.
4. Researchers Unleash Ransomware Annihilation. BankInfoSecurity - <http://www.bankinfosecurity.com/researchers-unleash-ransomware-annihilation-a-9255>, July 2016.
5. CryptoDrop Stops Ransomware by Stopping its Encryption. Security Intelligence - https://securityintelligence.com/news/cryptodrop-stops-ransomware-by-stopping-its-encryption/?utm_source=tfeed&utm_medium=twitter, July 2016.
6. Ransomware 'stopped' by new software. BBC - <http://www.bbc.com/news/technology-36772461>, July 2016.
7. Researchers create effective anti-ransomware solution. Help Net Security - <https://www.helpnetsecurity.com/2016/07/12/anti-ransomware-solution/>, July 2016.
8. Florida U boffins think they've defeated all ransomware. http://www.theregister.co.uk/2016/07/12/ransomware_defeated/, July 2016.
9. This Anti-Ransomware Tool Could Save You Hundreds of Pounds. Huffington Post - http://www.huffingtonpost.co.uk/entry/anti-ransomware-tool-save-hundreds-pounds_uk_57838beee4b0935d4b4b30ba, July 2016.
10. Researchers develop method to stop 100% of ransomware before it encrypts all files. Myce - <http://www.myce.com/news/researchers-develop-method-stop-100-ransomware-encrypts-files-79873/>, July 2016.
11. Desarrollan una solución para detener el ransomware. ComputerHoy - <http://computerhoy.com/noticias/software/desarrollan-solucion-detener-ransomware-47972>, July 2016.
12. Why your antivirus software can't stop ransomware. Futurity - <http://www.futurity.org/ransomware-computer-files-1198242-2/>, July 2016.
13. CryptoDrop Gives Users Hope to Prevent Ransomware Infections in the Future. Softpedia - <http://news.softpedia.com/news/cryptodrop-gives-users-hope-to-prevent-ransomware-infections-in-the-future-506187.shtml>, July 2016.

14. Could this be the answer to the ransomware threat?, Consumer Affairs. Consumer Affairs - <https://www.consumeraffairs.com/news/could-this-be-the-answer-to-the-ransomware-threat-071116.html>, July 2016.
15. Extortion extinction: Researchers develop a way to stop ransomware. Phys.org - <http://phys.org/news/2016-07-extortion-extinction-ransomware.html>, July 2016.
16. Researchers Develop A Way To Stop Ransomware By Watching The Filesystem. Slashdot - <https://yro.slashdot.org/story/16/07/08/2242244/researchers-develop-a-way-to-stop-ransomware-by-watching-the-filesystem>, July 2016.
17. Mohul Ghosh. Trak.in - Digital Money Apps In India Are Unsafe and Unsecured - Researchers. <http://trak.in/tags/business/2015/08/17/digital-money-apps-india-unsafe-unsecured/>, August 2015.
18. Richard Handford. Mobile World Live - Survey finds security holes in mobile money apps. <http://www.mobileworldlive.com/money/news-money/survey-finds-security-holes-in-mobile-money-apps/#.Vc27Y-QTmSQ.twitter>, August 2015.
19. JENNIFER VALENTINO-DEVRIES. Wall Street Journal - Researchers Find Security Flaws in Developing-World Money Apps. <http://blogs.wsj.com/digits/2015/08/11/researchers-find-security-flaws-in-developing-world-money-apps/>, August 2015.
20. Jonathon Cheng. Wall Street Journal - Samsung Phone Studied for Possible Security Gap. <http://online.wsj.com/news/articles/SB10001424052702304244904579276191788427198>, December 2013.
21. N. V. The Economist - The Threat in the Pocket. <http://www.economist.com/blogs/babbage/2013/10/difference-engine-0?fsrc=scn/fb/wl/bl/thethreatinthepocket>, October 2013.
22. Antone Gonsalves. ComputerWorld - Let's Dump Anti-Virus and Move On:. <http://blogs.computerworld.com/mobile-security/22969/lets-dump-av-and-move>, October 2013.
23. Mathew J. Schwartz. InformationWeek - Google: Don't Fear Android Malware. <http://www.informationweek.com/security/mobile/google-dont-fear-android-malware/240162399>, October 2013.
24. Kirsten Doyle. ITWeb - Android Threat Exaggerated, or is it? http://www.itweb.co.za/index.php?option=com_content&view=article&id=68055, October 2013.
25. Danielle Walker. SC Magazine - Mobile malware prevalence expands, but privacy-abusing apps should be top of mind. <http://www.scmagazine.com/mobile-malware-prevalence-expands-but-privacy-abusing-apps-should-be-top-of-mind/article/300597/>, June 2013.
26. Jim Burrell. WABE NPR - Mobile Web Browsers Full of Security Risks, Tech Professor Finds. <http://wabe.org/post/mobile-web-browsers-full-security-risks-tech-professor-finds>, December 2012.

27. Mark Huffman. Consumer Affairs - Georgia Tech: mobile browsers fail safety test. <http://www.consumeraffairs.com/news/georgia-tech-mobile-browsers-fail-safety-test-120612.html>, December 2012.
28. Matthew J. Schwartz. Information Week - Blame Screen Size: Mobile Browsers Flunk Security Tests. <http://www.informationweek.com/security/mobile/blame-screen-size-mobile-browsers-flunk/240143999>, December 2012.
29. Jon Gold. Network World - Ga. Tech researchers: Mobile Browsers need better HTTPS indicators. <http://www.networkworld.com/news/2012/120512-mobile-browsers-264846.html>, December 2012.
30. United Press International. Study: Most mobile Web browsers unsafe. http://www.upi.com/Science_News/Technology/2012/12/05/Study-Most-mobile-Web-browsers-unsafe/UPI-73431354743353/#ixzz2EGtQsuLd, December 2012.
31. Suzanne Choney. Mobile browser woes can fool even experts: report. <http://www.nbcnews.com/technology/mobile-browser-woes-can-fool-even-experts-report-1C7451203>, December 2012.
32. Meghan Kelly. VentureBeat - 3 hot security startups to watch. <http://venturebeat.com/2012/02/27/3-security-startups-to-watch-at-the-2012-rsa-conference/>, February 2012.
33. Jacob Goodwin. Government Security News - RSA 2012 – Pindrop Security can distinguish a fraudulent phone call from a real one. <http://www.gsnmagazine.com/node/25721?c=communications>, February 2012.
34. Matt Liebowitz. Phone hack logs keystrokes from nearby computers. MSNBC.com - http://www.msnbc.msn.com/id/44993238/ns/technology_and_science-security/#.TqU5MNSjPh4, October 2011.
35. Jacob Aron. iPhone keylogger can snoop on desktop typing. New Scientist - <http://www.newscientist.com/article/dn21059-iphone-keylogger-can-snoop-on-desktop-typing.html>, October 2011.
36. iPhone Keylogger Can Snoop on Desktop Typing. Slashdot - <http://mobile.slashdot.org/story/11/10/18/2346222/iphone-keylogger-can-snoop-on-desktop-typing>, October 2011.
37. Robert Lemos. Smart Phones Could Hear Your Password. Technology Review - <http://www.technologyreview.com/computing/38913/?p1=A2>, October 2011.
38. Kevin McCaney. Bad vibrations: How smart phones could steal PC passwords. Government Computer News - <http://gcn.com/articles/2011/10/18/smart-phone-sensors-steal-keystrokes.aspx>, October 2011.
39. PhysOrg. Turning iPhone into spiPhone: Smartphones' accelerometer can track strokes on nearby keyboards. PhysOrg.com - <http://www.physorg.com/news/2011-10-iphone-spiphone-smartphones-accelerometer-track.html>, October 2011.
40. Brid-Aine Parnell. Securo-boffins call for 'self-aware' defensive technologies. The Register - http://www.theregister.co.uk/2011/09/14/self_aware_cyber_security_technologies_should_be_a_top_priority/, September 2011.

41. Clay Dillow. 'PinDr0p' Tech Uses Unique Noise Fingerprints to Trace Calls. Popular Science - <http://www.popsci.com/technology/article/2010-10/pindr0p-tech-tags-phone-calls-unique-fingerprints-trace-call-paths-across-networks>, October 2010.
42. Lewis Page. Voice-routing call fingerprint system fights vishing. The Register - http://www.theregister.co.uk/2010/10/06/voice_fingerprints, October 2010.
43. Science Daily. Voice Phishing: System to Trace Telephone Call Paths Across Multiple Networks Developed. <http://www.sciencedaily.com/releases/2010/10/101005121820.htm>, October 2010.
44. Brian Kalish. To Text or Not to Text During Emergencies. NextGov.com - http://www.nextgov.com/nextgov/ng_20100914_5986.php?oref=topnews, September 2010.
45. Ki Mae Heussner. 'Operation Chokehold': Fake Steve Jobs Rallies iPhone Users to Cripple AT&T Network. ABC News - <http://abcnews.go.com/Technology/GadgetGuide/fake-steve-jobs-rallies-iphone-users-cripple-att/story?id=9355447>, December 2009.
46. Bob Brown. Researchers Set Their Sights on iPhones, Mobile Malware. PC World Magazine - http://www.pcworld.com/article/182005/iphone_worms_mobile_malware.html?tk=rss, November 2009.
47. MacGregor Campbell. Botnets show their disruptive potential. New Scientist Magazine - <http://www.newscientist.com/article/mg20427347.000-mobile-botnets-show-their-disruptive-potential.html>, November 2009.
48. Angela Moscaritolo. Remote repair for infected phones in development. SC Magazine - <http://www.scmagazineus.com/remote-repair-for-infected-phones-in-development/article/157504/>, November 2009.
49. Bob Brown. iPhone worms, other smartphone malware in researchers' sights. Network World - <http://www.networkworld.com/news/2009/111109-smartphone-security-georgia-tech.html?hpg1=bn>, November 2009.
50. Urvaksh Karkaria. GT researchers work to secure cellphones. Atlanta Business Chronicle - <http://atlanta.bizjournals.com/atlanta/blog/atlantech/2009/11/cellphone.html>, November 2009.
51. Making Carriers Shoulder Smartphone Security. http://mobile.slashdot.org/story/09/11/11/2318247/Making-Carriers-Shoulder-Smartphone-Security?art_pos=31, November 2009.
52. Ben Meyer. Georgia Tech to Lead Fight Against Cell Phone Hackers. NBC 11 Atlanta - <http://www.11alive.com/news/local/story.aspx?storyid=132505&catid=3>, July 2009.
53. Illena Armstrong. Safeguarding your mobile networks. SC Magazine - <http://www.scmagazineus.com/Safeguarding-your-mobile-networks/article/138289/>, June 2009.
54. Kelli B. Grant. Four Free Cellphone Apps to Help Manage Your Money. SmartMoney Magazine - <http://www.smartmoney.com/Spending/Deals/4-Great-Free-Finance-Apps-for-Cellphones/>, June 2009.
55. Amanda Hoffstrom. Technology's limitations in alerting campus danger. UWire Magazine - <http://www.uwire.com/Article.aspx?id=3738798>, February 2009.

56. Laura Sydell. Compromise Allows Obama To Keep BlackBerry. National Public Radio - <http://www.npr.org/templates/story/story.php?storyId=99790788>, January 2009.
57. Dennis Carter. Questions abound as emergency alert flops Virginia Tech's text-message alert system failed when the sound of gunfire was heard on campus; officials scramble to understand why. eSchool News - http://www.eschoolnews.com/iphone/top-story/index.cfm?i=56122#_56122, November 2008.
58. Jessica Bauer. Study: Text alerts may fail in real emergency. Diamondback Online - <http://media.www.diamondbackonline.com/media/storage/paper873/news/2008/10/14/News/Study.Text.Alerts.May.Fail.In.Real.Emergency-3485509.shtml>, October 2008.
59. Associated Press. Hackers Expected To Start Targeting Cell Phones. <http://cbs5.com/watercooler/Cell.Phones.Hackers.2.840909.html>, 2008.
60. Associated Press. College alert systems unreliable, study says. http://www.ajc.com/search/content/metro/stories/2008/09/25/college_campus_alerts.html, 2008.
61. Lee Shearer. Study: Campus alerts unreliable. Athens Banner Herald http://www.onlineathens.com/stories/092508/uga_336494829.shtml, 2008.
62. Bill Ray. 3G Americas warns against text warning systems. The Register - http://www.theregister.co.uk/2008/09/18/emergency_text/, 2008.
63. 3G Americas. 3G Americas Highlights New Research Report on Use of Cellular Text Messaging for Emergency Alert Services. 3G Americas http://www.3gamericas.org/English/news_room/DisplayPressRelease.cfm?id=3400&s=ENG, 2008.
64. Evan Koblentz. Web Exclusive: From Messaging to Management Duty. Wireless Week - <http://www.wirelessweek.com/Messaging-to-Management-Duty.aspx>, 2008.
65. Christopher Beam. How Do You Intercept a Text Message? Turn your cell phone into a spy gadget. Slate Magazine <http://www.slate.com/id/2161402/>, 2007.
66. Jamming Cellphones with Text Messages. Slashdot <http://it.slashdot.org/it/05/10/05/1839217.shtml?tid=215&tid=172>, 2005.
67. Cell phone networks at risk? CNN http://money.cnn.com/2005/10/05/technology/hacker_cellphones/, 2005.
68. John Schwartz. Text Hackers Could Jam Cellphones, a Paper Says. The New York Times <http://www.nytimes.com/2005/10/05/technology/05phone.html?ex=1286164800&en=d917b9cd43dfaa31&ei=5090&partner=rssuserland&emc=rss>, 2005.