

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION**

K.MIZRA LLC,

Plaintiff,

v.

GOOGLE LLC,

Defendant.

Civil Action No.: 1:26-cv-00754

Jury Trial Demanded

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff K.Mizra LLC ("K.Mizra") files this Complaint for Patent Infringement against Defendant Google LLC ("Google"), alleging as follows:

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	THE PARTIES.....	2
III.	JURISDICTION AND VENUE.....	2
IV.	GENERAL ALLEGATIONS	3
A.	The '120 Patent.....	3
1.	Prior Licensing and Litigation Concerning the '120 Patent	3
2.	Problems in 2007 Solved By the '120 Patent	4
3.	K.Mizra's Asserted '120 Patent Claims Are Presumed Valid and Patent-Eligible	8
4.	Google's Infringing Vertex AI Studio Product	10
B.	The '717 Patent.....	11
1.	Prior Licensing and Litigation Concerning the '717 Patent	12
2.	Problems in 2007 Solved By the '717 Patent	12
3.	K.Mizra's Asserted '717 Patent Claims Are Presumed Valid and Patent-Eligible	13
4.	Google's Infringing Nest Devices.....	19
C.	Google's Knowledge of the Asserted Patents.....	23
V.	FIRST CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '120 Patent)	24
VI.	SECOND CLAIM FOR RELIEF (Patent Infringement Under 35 U.S.C. § 271 of the '717 Patent)	33
VII.	REQUEST FOR RELIEF	48
VIII.	DEMAND FOR JURY TRIAL.....	49

I. INTRODUCTION

1. K.Mizra is a patent licensing company run by experienced management. The company focuses on high-value, high-quality patents and owns patent portfolios originating from a wide array of inventors, including patents and patent portfolios developed by well-known multinational corporations such as IBM, Intel, Rambus and others, as well as from research institutes such as Nederlandse Organisatie voor Toegespast Natuurwetenschappelijk Onderzoek (Netherlands Organization for Applied Scientific Research). By focusing on high-quality patents, K.Mizra provides a secondary market that enables inventors to recoup their research and development investments and to continue their innovations. K.Mizra offers licenses to its patents on reasonable terms and, in this way, plays an important role in the development of technologies that improve businesses and lives.

2. K.Mizra is the owner by assignment of United States Patent No. 8,438,120 ("the '120 Patent"). The '120 Patent was invented by Stephan Alexander Raaijmakers of the Nederlandse Organisatie Voor Toegepast-Natuurwetenschappelijk Onderzoek TNO, a well-known and widely respected technology research institution ("TNO").

3. K.Mizra is also the owner by assignment of United States Patent No. 8,144,717 ("the '717 Patent" or "the Mesh Networking Patent"). The '717 Patent was invented by Marinus Johannes Blange and Miodrag Djurica of the TNO.

4. The '120 Patent and '717 Patent are collectively referred herein as "the Asserted Patents." K.Mizra remains ready, willing, and able to provide commercially reasonable licenses for its various patented technologies, including its Asserted Patents, to all entities who wish or need to use its covered technologies internally or in connection with products or services offered to others. As outlined below, Google is one such entity.

II. THE PARTIES

5. K.Mizra is a Delaware limited liability company with a mailing address of 777 Brickell Avenue, #500-96031, Miami, Florida 33131, and operates in Florida.

6. K.Mizra is the owner by assignment of the Asserted Patents.

7. Upon information and belief, Google is a corporation organized and existing under the laws of the state of Delaware, with a principal place of business at 1600 Amphitheatre Parkway, Mountain View, California 94043.

8. Upon information and belief, Google maintains an office in this District at 500 West Second Street, Austin, Texas 78701.

9. Upon information and belief, Google employs hundreds of people in its Austin, Texas office, including persons with knowledge of the Google products and technology at issue in this case.

10. Upon information and belief, Google is also registered to do business in Texas and may be served with process by serving Corporation Service Company, Google's registered agent in Texas located at 211 East Seventh Street, Suite 620, Austin, Texas 78701-3218.

III. JURISDICTION AND VENUE

11. This is an action for patent infringement under the patent laws of the United States, 35 U.S.C. §§ 1 *et seq.*, including 35 U.S.C. §§ 271, 281, and 284, among others. The Court has subject-matter jurisdiction over the claims raised in this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

12. This Court has personal jurisdiction over Google by virtue of, *inter alia*, its conduct of business in this District, its registration to do business in Texas, its appointment of a registered agent in Texas, its employment of hundreds of persons in its Austin office located in this District,

and its substantial, continuous, and systematic contacts with the state of Texas and this District. Google intentionally markets and sells its infringing products directly and through agents to residents of Texas, enjoys substantial income from its business activities in the state of Texas and this District, and/or directly, by its own actions, and/or in combination with actions of customers and others under its control, has committed acts of patent infringement in this District at least by selling infringing products in this District.

IV. GENERAL ALLEGATIONS

A. The '120 Patent

13. K.Mizra is the sole owner by assignment of the '120 Patent with the full and exclusive right to bring suit to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the '120 Patent.

14. The '120 Patent, titled "Machine Learning Hyperparameter Estimation," was issued by the United States Patent and Trademark Office ("USPTO") to Inventor Stephen Alexander Raaijmakers on May 7, 2013. A true and correct copy of the '120 Patent is attached hereto as Exhibit 1 and incorporated by reference.

15. The '120 Patent is a U.S. National Phase filing of PCT/NL2008/050247 and claims priority to European Application Nos. 07106963.7 filed April 25, 2007 and 07112037.2 filed July 9, 2007.

1. Prior Licensing and Litigation Concerning the '120 Patent

16. The '120 Patent has previously been licensed to various industry participants.

17. Google is not and has never been a licensee of the '120 Patent nor has it had nor does it have any rights to use technologies covered by the Patent. Google thus has no ownership or other rights (and is entitled to no rights) relating to the '120 Patent.

18. No party licensed to the '120 Patent has used the technology claimed by the '120 Patent in a manner that requires marking under 35 U.S.C. § 287. K.Mizra is therefore not barred from seeking past damages for infringement of the '120 Patent under 35 U.S.C. § 287(a).

2. Problems in 2007 Solved By the '120 Patent

19. The technology described in the '120 Patent was invented by Stephen Raaijmakers and generally relates to improving the operation of Artificial Intelligence ("AI") through the improved automatic, systematic and computer implemented training techniques. More specifically, AI typically uses models to perform specific tasks, and, at its most basic level, an AI model is a program that has been "trained" on a set of data to recognize certain patterns or to make certain decisions without further human intervention. For example, an AI model could be used to automatically detect spam or other unwanted emails in a user's email inbox or distinguish between ripe and unripe fruit to be picked, etc. The '120 Patent uses the term "classifier" to refer to these types of AI models, as they are used to separate inputs (emails or fruit) into different classifications (spam or not spam/ripe or unripe).

20. AI models use variables called "parameters" to determine what result it should produce (e.g., to determine whether to classify an email as spam or not spam). These parameters are not set manually. Instead, they are learned automatically during the training process. During training, the model may be exposed to a labeled dataset and the parameters can be adjusted to minimize prediction errors. For instance, in the spam email example, the AI model may be trained by exposing the model to a large number of emails that have been labeled as either spam or not spam. The model identifies and thus learns the traits and characteristics found in these various emails that are indicative of a spam email or not spam email. Automatically adjusting the

parameters of the model to minimize the times when a spam email is marked as not spam, or vice versa, is the goal of parameter-based training.

21. AI models are generally trained for a specific purpose. However, training a full model is time consuming and, in some cases, has given way to the use of foundational models, which are models that are pre-trained on large, genericized datasets. These foundational models can later be fine-tuned for use in addressing a particular problem or issue by, for example, feeding it a set of additional relevant data and training upon that dataset. By way of example, a foundational AI model for diagnosing medical conditions could be fine-tuned by training it using various medical records. Similarly, a foundational large language model ("LLM") like ChatGPT could be fine-tuned to adjust its conversational tone by training it on sample texts representing a desired communication style. Using a foundational AI model is like getting a shirt and later tailoring it to fit your body, rather than starting with fabric, thread, and a needle.

22. As AI models become more advanced and versatile, they also demand more data and computing power to be properly trained. One of the main challenges in this exploding field of technology is efficiently training these models, which typically is done by configuring variables in a model's learning process. These variables are known as hyperparameters. The hyperparameters may, for example, define the architecture (and thus the size) of the model, as well as, how the model changes based on the training data it digests and that will affect how efficiently and effectively the model is trained. For example, a hyperparameter called the learning rate may be used to control how much a model adjusts its parameters during training. If the learning rate of the model is too low, it will take a long time for the model to reach an acceptable result. On the other hand, if the learning rate is too high, the model may overreact to each training round, bouncing around an optimum result instead of converging on that result. Different hyperparameters

may be used depending on whether a user is training a brand-new model or fine-tuning a foundational model. In either case, though, identifying the right hyperparameters and their values during the model training process improves the model's efficiency and thus the predictions or decisions to be output by the model.

23. Claims of the '120 Patent are directed to technological solutions that address specific challenges grounded in the systematic and automatic tuning of hyperparameters so that efficient and effective AI models can be computer generated. Creating these models is a tremendous concern for modern enterprises, since they are now being used in all manner of human endeavor, from drug discovery, to providing banking efficiencies, etc.

24. The inventor of the '120 Patent understood that while basic models that trained on parameters existed prior to 2007, as the data to be analyzed become more and more voluminous and complex, the use of just parameters in developing a model would not be enough. Rather, the additional use of hyperparameters that could automatically and systematically be optimized by the computing system that was training the model was needed. The inventor also realized that the existing methods for estimating and selecting hyperparameters were unsuitable in many cases. For example, the '120 Patent discloses:

Hyperparameters may be determined using various methods, for example heuristic or statistical optimization methods. However, many heuristic methods are ad hoc methods, while not all statistical optimization methods are suitable for determining hyperparameters.

The inventor thus recognized that a new paradigm needed to be developed. That paradigm was then well before its time invented and is disclosed in the '120 Patent.

25. For example, the '120 Patent explains that a method of determining hyperparameters in a machine learning system is disclosed. The method is iterative, and with each computer implemented iteration, a random or pseudorandom sample of hyperparameter vectors

selected from a set of possible hyperparameter vectors must be considered, updating the estimate of the target hyperparameter vector by selecting for use the best vector in the present and previous iterations. The '120 Patent specification further explains that:

By selecting from the random sample a or the hyperparameter vector producing the best result in the present and any previous iterations, and using said selected hyperparameter vector to update the estimate of the target hyperparameter vector, the properties of the best performing hyperparameters are used to guide the next iteration [of the hyperparameter selection process]. In this way, a method similar to the cross-entropy method [used for selection of parameters] can be used effectively for determining hyperparameters, even when the hyperparameters are not continuous and their effects are not transparent. By using the method according to the present invention, an improved classification [i.e. model] accuracy is achieved.

...

In a preferred embodiment, the method according to the Present invention further comprises the steps of:

selecting from the random sample a further hyperparameter vector producing the best result in any previous iterations, and restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.

The step of restricting the random sample preferably involves the use of an interval surrounding the further selected hyperparameter vector which produces the best result in any previous iteration.

...

In a preferred embodiment, E^t is the hyperparameter vector X^t , at iteration t producing the best result $S(X^t)$ in the present iteration t and all previous iterations (if any). The random sample X^t , has elements X^t_{ij} , while the step of updating the hyperparameters comprises the step of determining the (target) hyperparameter v^t_j , where

$$v^t_j = \frac{\sum_{i=1}^n I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t) X^t_{ij}}{\sum_{i=1}^n I\{S(X^t_i) \geq \gamma^t\} W(X^t_i; E^t)},$$

wherein γ^t is a threshold value, W is a weighting function and $I\{S(X^t_i) \geq \gamma^t\}$ is an indicator function which is equal to 1 if $S(X^t_i)$ greater than or equal to the threshold value γ^t , and else is equal to 0. The weighting function W is preferably given by

$$W(X_i^t; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^m (X_{ij}^t - E_j^t)^2}}{\sqrt{\sum_{j=1}^m (X_{ij}^t)^2} \sqrt{\sum_{j=1}^m (E_j^t)^2}}.$$

Accordingly, the step of updating the hyperparameters may involve a weighting function based upon a distance function, preferably a Euclidean distance function. It is noted that when the Euclidean distance between E^t and X_{ij}^t , is zero, the weighting function W is equal to one.

The method according to the present invention is typically carried out by computer apparatus, for example a general purpose computer system comprising a processor and an associated memory, one part of the memory storing a soft ware program for instructing the processor to carry out the method steps of the present invention, and another part of the memory storing data, said data comprising the hyperparameter values referred to above.

See, e.g., Ex. 1 at 2:1–4:14.

26. The solution to the problems identified in the '120 Patent—as specified and claimed in the '120 Patent—was an advanced departure from the conventional tuning of hyperparameters in AI models in use in 2007 and was then, as it remains today, patent-eligible, highly valuable, novel, and non-obvious technology.

3. **K.Mizra's Asserted '120 Patent Claims Are Presumed Valid and Patent-Eligible**

27. The '120 Patent is directed to a critical advancement in machine learning, namely, an effective method of estimating hyperparameters given large potential datasets. The sole named inventor of the '120 Patent, Stephan Raaijmakers, is a pioneer in artificial intelligence, holding a Ph.D. in Machine Learning from Tilburg University and a chair on Communicative AI at Leiden University. Dr. Raaijmakers is also the author of the book *Deep Learning for Natural Language Processing*, distributed by Simon & Schuster.

28. K.Mizra asserts that at least, and without limitation, Claim 1 of the '120 Patent has been directly infringed, either literally or under the doctrine of equivalents. K.Mizra reserves the right to assert additional claims of the '120 Patent, including both independent and dependent claims, pursuant to the Court's (and other applicable) rules and procedures and as discovery progresses. These claims are referred to herein as the "Asserted '120 Claims."

29. None of the Asserted '120 Claims are directed to abstract ideas, and each employs inventive concepts and is directed to patent-eligible subject matter. All claims of the '120 Patent are also presumed to be valid and enforceable against Google and others.

30. Indeed, the '120 Patent's specification and claims demonstrate that the need satisfied by the inventions of the Asserted '120 Claims was long-felt in the industry and thus unconventional. The '120 Patent is directed to the use of "hyperparameters" to train machine learning systems. "Hyperparameters" are parameters that can be used to determine, for example, decision criteria to be used in classifying objects. (Ex. 1 at 1:20–23.) Hyperparameters in turn must be determined using particular methods. The '120 Patent specification references two papers discussing a method of estimating parameters called "cross-entropy" and explains why this approach is not suitable for determining hyperparameters in the machine learning context. (*Id.* at 3:44–59.) Rather, the '120 Patent describes an iterative method of determining hyperparameters by drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, selecting from the sample a hyperparameter vector producing the best result, updating the estimate using the selected hyperparameter vector, and repeating the process to determine the best hyperparameters. (*Id.* at 2:1–19.)

31. The '120 Patent thus identifies the conventional method of determining parameters and explains a specific technical improvement, namely:

The present invention is based upon the insight that methods similar to the cross-entropy method can be adapted for hyperparameter estimation by adding a best performance preservation feature. The present invention benefits from the additional insight that the best performance in a particular iteration may be preserved by restricting the sample using the best performance of the previous iterations.

(*Id.* at 6:61–67.) The '120 Patent thus specifically explains a technological solution to improving known methods of machine learning parameter estimation, and the Asserted '120 Claims directed to those improvements are patent eligible. *See, e.g., Aon Re, Inc. v. Zesty.AI, Inc.*, 791 F. Supp. 3d 531, 539 (D. Del. 2025) ("Although the patent does not purport to invent machine learning algorithms, computer hardware, or the general concept of aerial imagery analysis or risk assessment, it does specifically define how these known components must be structured to achieve a supposed technological improvement in accuracy and efficiency to the combination of those components."); *cf. Recentive Analytics, Inc. v. Fox Corp.*, 134 F.4th 1205, 1213 (Fed. Cir. 2025) (holding claims ineligible where "the claims do not delineate steps through which the machine learning technology achieves an improvement").

32. Because the Asserted '120 Claims are directed to a specific improvement in machine learning technology, the claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

4. Google's Infringing Vertex AI Studio Product

33. Google's Vertex AI Studio is part of the Google Cloud Platform and is a complete system for building, training, and using AI models. It offers a range of tools and APIs (Application Programming Interfaces) that let users either create and train their own models from scratch, or work with Google's foundation models, like Gemini, Imagen, Codey, and Chirp, and tailor them for specific needs. As is here relevant, Vertex AI provides several options for training all manner

of models. One of the most powerful of these provides a user with complete control over the model development process by allowing them access to powerful hyperparameter tuning and optimization options. These can be implemented using Google's Vertex AI Vizier. It is Google's provision of this computer implemented hyperparameter tuning ability that takes advantage of iterative, randomized/pseudo-randomized hyperparameter tuning techniques, that makes Google's Vertex AI so valuable to its customers. Google's usurpation of the critical '120 Patent's patented technology without K.Mizra's authorization allows Google to provide to its customers highly tuned and effective AI models, generating significant revenues and reputational prestige for Google, all to the injury of K.Mizra.

34. Google has been making, selling, using and offering for sale AI products and services that infringe the '120 Patent in violation of 35 U.S.C. § 271 (collectively, "the Accused Hyperparameter Tuning Instrumentalities"). These Accused Hyperparameter Tuning Instrumentalities include, but are not limited to, Google's Vertex AI system, the sale, offer for sale, use, and/or manufacture in the United States of which constitutes infringement of at least Claim 1 of the '120 Patent, either literally or under the doctrine of equivalents.

B. The '717 Patent

35. K.Mizra is the sole owner by assignment of the '717 Patent with the full and exclusive right to enforce the Patent. K.Mizra is also entitled to sue to collect damages for all past infringement of the '717 Patent.

36. The '717 Patent, titled "Initialization of a Wireless Communication Network," was issued by the USPTO to Inventors Blange and Djurica on March 27, 2012. A true and correct copy of the '717 Patent is attached hereto as Exhibit 2 and incorporated by reference.

37. The '717 Patent is a U.S. National Phase filing of PCT/NL2007/0000001, filed on December 2, 2007, and is entitled to that priority date. The '717 Patent further claims priority to European Patent Application No. EP 05078044, filed December 30, 2005, and is entitled to that priority date.

1. Prior Licensing and Litigation Concerning the '717 Patent

38. The Mesh Networking Patent has previously been licensed to various industry participants.

39. Google is not and has never been a licensee of the '717 Patent nor has it had nor does it have any rights to use technologies covered by the Patent. Google thus has no ownership or other rights (and is entitled to no rights) relating to the '717 Patent.

40. No party licensed to the '717 Patent has used the technology claimed by the '717 Patent in a manner that requires marking under 35 U.S.C. § 287. K.Mizra is therefore not barred from seeking past damages for infringement of the '717 Patent under 35 U.S.C. § 287(a).

2. Problems in 2007 Solved By the '717 Patent

41. The technology described in the '717 Patent was invented by Marinus Johannes Blange and Miodrag Djurica and generally relates to creation of a flexible mesh wireless network.

42. Although wireless communications networks were known in the art prior to the invention, the known networks had several drawbacks. For example, there existed known techniques for routing messages in wireless networks. *See generally* Ex. 2 at 1:13–58. However, as explained in the '717 Patent, these techniques "require[d] transmission of information via sensor stations before the routing topology [of the network] has been defined." *Id.* at 1:59–62. "This complicates the (sensor) stations, because they will have to support different modes of

communication while operating in a low-power mode." *Id.* at 1:62–64. As further explained in the '717 Patent, the existing techniques also resulted in either slow or inefficient networks:

If forwarding of information is only done using the definite routing topology, a central approach is slow (each station has to wait until its upward station has been centrally assigned a route) and a decentral approach is inefficient (a routing topology with an unnecessary amount of branches has been created).

Id. at 1:64–2:2.

43. Another known protocol was a mobile communications protocol which uses two types of networks: a high power network and a low power network. *See generally* Ex. 2 at 2:9–57. However, the '717 Patent notes that "[t]he use of two types of network without central unit makes this system complex and inefficient in bandwidth use." *Id.* at 3:1–2.

44. The solution to these problems—as specified and claimed in the '717 Patent—was an advanced departure from the conventional networking solutions then in use and was then, as it remains today, patent eligible, highly valuable, novel, and non-obvious technology.

**3. K.Mizra's Asserted '717 Patent Claims
Are Presumed Valid and Patent-Eligible**

45. K.Mizra asserts that at least, and without limitation, Claim 11 of the '717 Patent has been directly infringed, either literally or under the doctrine of equivalents by Google. K.Mizra reserves the right to assert additional claims of the '717 Patent, including both independent and dependent claims, pursuant to the Court's (and other applicable) rules and procedures and as discovery progresses. These claims are referred to herein as the "Asserted '717 Claims."

46. None of the Asserted '717 Claims are directed to abstract ideas, and each employs inventive concepts and are directed to patent-eligible subject matter. All claims of the '717 Patent are also presumed to be valid and enforceable against Google and others.

47. Indeed, the '717 Patent's specification and claims demonstrate that the need satisfied by the inventions of the Asserted '717 Claims was long-felt in the industry and thus

unconventional. As one example, the '717 Patent explains that stations within wireless networks were required to transmit information before routing topology had been defined, *i.e.* when plural routes were still possible. The '717 Patent explains that this created complications because wireless stations would have to support different modes of communication while operating in low-power states, which was difficult to achieve. The '717 Patent further explains that wireless links between nodes were hindered by barriers and that using the same frequency for upward and downward link routes required synchronization, which led to further complications by requiring wireless stations to support synchronization and powerful radio frequency signals, again at low-power states. The '717 Patent thus provides "a wireless network wherein routes for communication can be determined without the need to forward information via stations for which the routes have not yet been defined," a method of operating such a network, and "stations for flexible use in wireless networks." (Ex. 2 at 3:6–15.)

48. The specification (including the provisions quoted above), the figures (including those included below), and the text related to the figures further illustrate the innovative network architecture captured by the Asserted '717 Claims. These figures include the following:

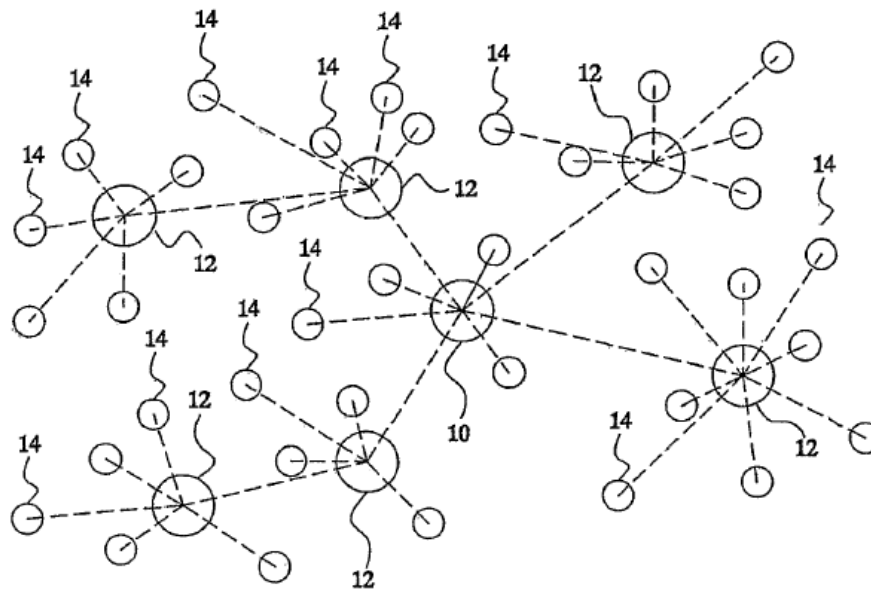


Fig.1

(See *id.* at Fig. 1.)

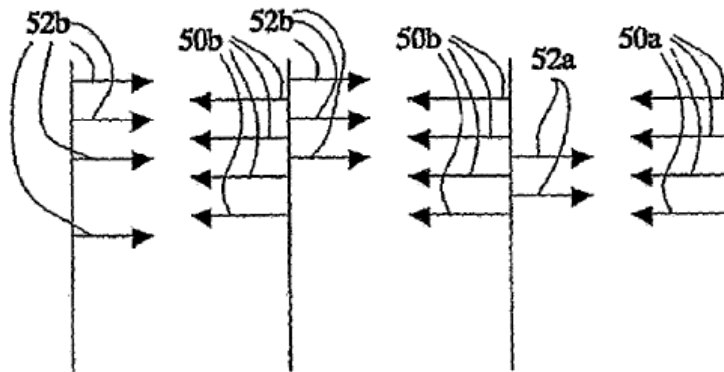


Fig.5a

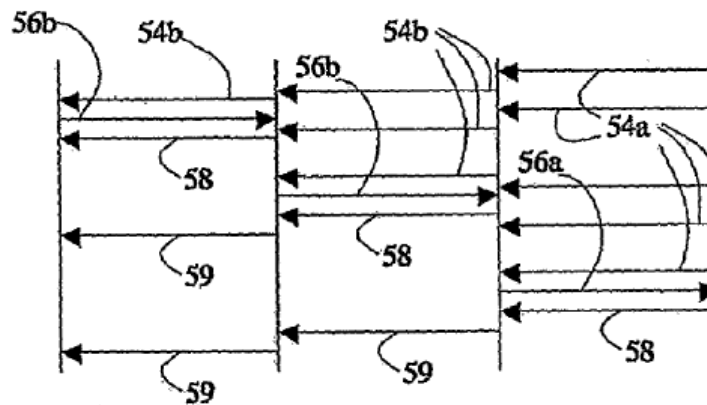


Fig.5b

(See *id.* at Fig. 5.)

49. The foregoing figures, which illustrate a wireless network and signaling protocol, demonstrate that the inventions of the Asserted '717 Claims focus on specific, highly efficient ways for a station to join a wireless network, thereby improving the operation and performance of the wireless network, as a whole. In particular, an unassociated station transmits an association request message and switches to an associated state upon reception of an association grant from

an association unit in response to one of its request messages. The association grant establishes an operating route associated with the newly associated station for exchanging operating messages with the association unit. Thereafter, the station can transmit association grants in response to received association request messages from additional stations seeking to join the network. This improves the efficiency and operability of the wireless network by facilitating the addition of new stations to the network. Thus, the Asserted '717 Claims are eligible as a matter of law for patent protection under step one of *Alice*.

50. All actions and steps recited in the Asserted '717 Claims, including the acts of transmitting association request messages, transmitting association grant messages, establishing operating routes and exchanging operating messages, and enabling stations to transmit association grant messages, require the involvement of various hardware components running dedicated software both before, during, and after the association of a station. Said another way, a claim directed to allowing a wireless network to automatically and dynamically associate a new station with the network and facilitate the addition of other stations is not simply adding a generic computer component to a fundamentally human process. Rather, it is an "improvement in the functioning of a" networked system that simply cannot be considered directed to an abstract concept. *Enfish LLC v. Microsoft Corp.*, 822 F.3d 1327, 1339 (Fed. Cir. 2016).

51. As the specification confirms, the improvement captured by the Asserted '717 Claims is not simply associating various devices into a wireless network, but is instead a specific network architecture whereby a particular unit operates to receive access requests and provide access grants, as well as providing operation information so that newly added wireless stations can transmit access grants to additional stations. Specifically, the specification of the '717 Patent discloses that to improve the efficiency of adding stations to a wireless network, a station seeking

to join the network operates initially in a non-associated state transmitting access requests. (*E.g.*, Ex. 2 at 8:25–32.) Subsequently, the station receives an access grant message and switches to an associated state. (*Id.* at 9:3–13.) In this state, the station establishes a dedicated communication route with the association unit or with a second station acting as an intermediary between the newly added station and the association unit, thereby ensuring efficient communication within the wireless network. (*Id.*) The newly added station can thereafter relay association grant messages to additional new stations, further enhancing the scope and efficiency of the wireless network. (*Id.* at 9:41–53.)

52. The '717 Patent specification further emphasizes that, once stations are added to the network via association grant messages, they have "well defined routes" to communicate within the network. (*Id.* at 9:63–10:6.) Thus, rather than requiring specific access routes in order to join the network initially, which creates needless complexity and power usage, the '717 Patent teaches that well defined routes are established after the station has been added to the network via an efficient method.

53. The Asserted '717 Claims are thus directed to a machine-implemented process for efficiently adding a new station to a wireless network and configuring that station to facilitate the addition of further stations to the network, improving the overall operation of the network itself. As such, the Asserted '717 Claims recite inventions with specific applications or improvements to technologies in the marketplace and cannot be considered abstract or patent ineligible under relevant law.

4. Google's Infringing Nest Devices

54. Google Nest is a line of products for home networking, allowing a user to control various Google and third-party smart devices throughout the Google Home app. On information and belief, Google Nest was previously known as Google Home.

55. Google offers a wide variety of home networking devices under the Google Nest brand. Many of these devices use a networking protocol called Thread.

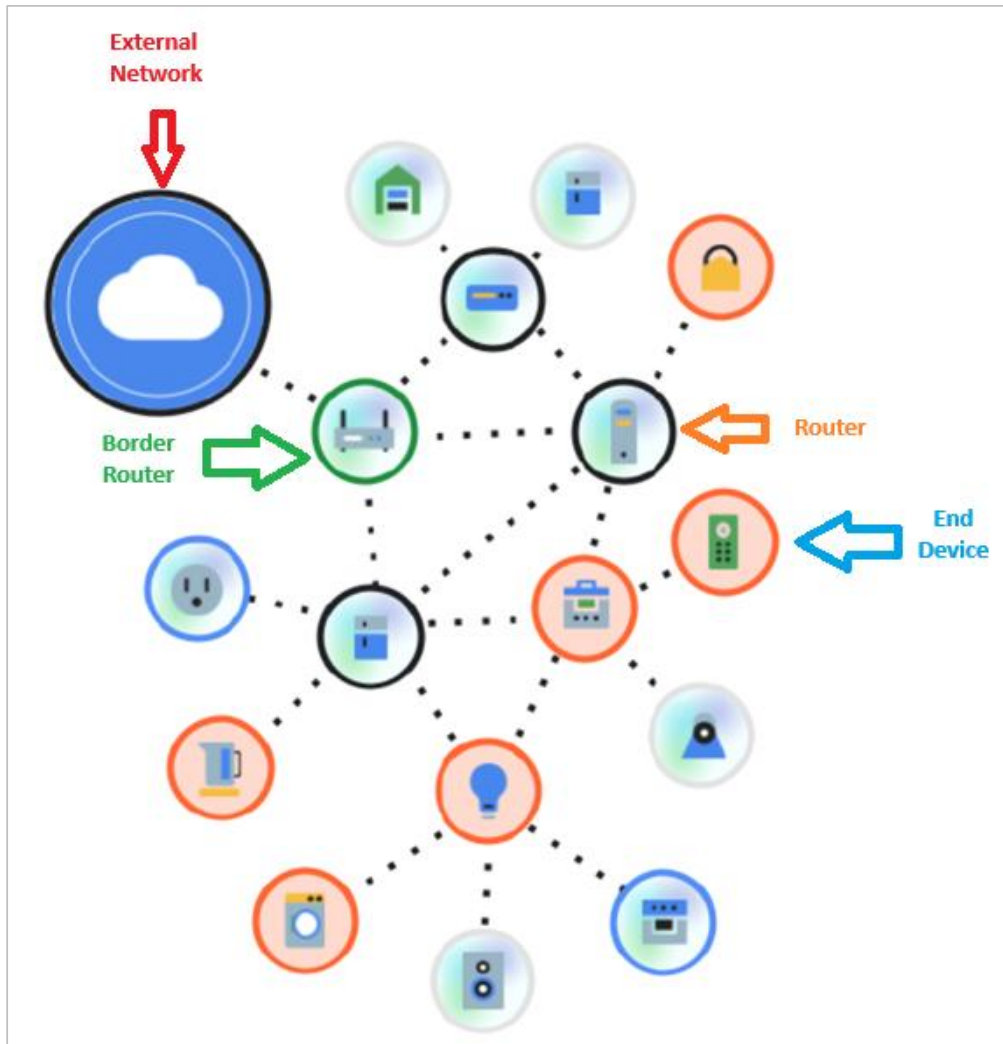
56. Thread is a networking protocol designed for low-power Internet of Things devices operating in an IEEE 802.11.15.4 wireless mesh network. It may be used alongside other networks, such as a Wi-Fi network, in a user's home. Thread was developed by the Thread group, which Google founded with other member companies in 2014.

57. Thread is built into many new smart devices such as smart bulbs, locks, sensors, and the like. Thread is used to create a mesh network in which each new smart device connects to the next nearest smart device, giving users more networking coverage and making the entire network more robust. If one device in the network fails, another can step in to maintain the signal, allowing the network to heal itself rapidly, effectively and efficiently.

58. A device in a Thread network may be either a router, which may forward data packets for other network devices and provides commissioning for devices trying to join the Thread network, or an end device, which communicates with a single router and does not forward packets for other network devices. An end device may convert itself into a router when a new device joins the network.

59. A special case of a router is a border router, which can forward information between a Thread network and a non-Thread network. For example, a border router may allow a user to control the devices on a Thread network through the Wi-Fi network via a border router.

60. An exemplary Thread network including a border router, routers, and end devices is shown below:



"Thread Play Services APIs," (available at <https://developers.home.google.com/matter/thread> (accessed March 2026 and incorporated by reference) (annotated)).

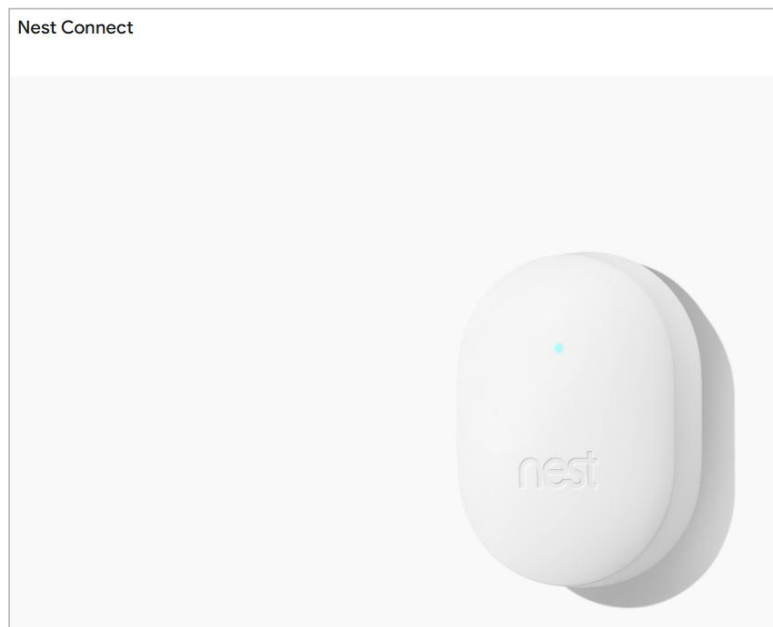
61. Google offers a wide range of products that use the Thread networking protocol. For example, Google offers several hubs for Google home that include a built-in Thread router, including the Nest Hub, Nest Hub Max, Nest Wifi Pro, and Google TV Streamer.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

Id.; see also "Connect Thread smart devices with Google TV Streamer (4K)," (available at <https://support.google.com/chromecast/answer/15178609> (accessed March 2026 and incorporated by reference)).

62. In addition, Google hub products such as the Google TV Streamer, Nest Hub (2nd Gen), Hub Nest Max Fuchsia, Nest Wi-Fi Point, and Nest Wi-Fi Router are certified as Built on Thread by the Thread Group.

63. As another example, Google offers a routing device called the Nest Connect.



Nest Connect – Range Extender for Detect & Guard – Google Store," (available at https://store.google.com/product/nest_connect (accessed March 2026 and incorporated by reference)). The Nest Connect includes a Thread radio that allows communication between a Thread border router and a peripheral device such as the Nest x Yale Lock.

In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

"Nest debuts Nest Secure home system and has a Thread router called Nest Connect," (available at <https://staceyoniot.com/nest-debuts-nest-secure-home-system-and-has-a-thread-router-called-nest-connect/> (accessed March 2026 and incorporated by reference)).

64. Yet another example, Google offers various peripheral devices that include a Thread radio such as the Nest x Yale Lock.



"Nest x Yale Lock – Key-Free Smart Deadbolt – Google Store," (available at https://store.google.com/product/nest_x_yale_lock (accessed March 2026 and incorporated by reference)).

65. Additional peripheral devices offered by Google that include a Thread radio include the Google Nest Learning Thermostat and the Google Nest Protect and CO alarm.

66. Google has been making, selling, using and offering for sale products including Thread radios that infringe the '717 Patent in violation of 35 U.S.C. § 271 (collectively, "the Accused Mesh Networking Instrumentalities"). These Accused Mesh Networking Instrumentalities include, but are not limited to, Google's Nest home networking products identified above, the sale, offer for sale, use, and/or manufacture in the United States of which constitutes infringement of at least Claim 11 of the Accused Mesh Networking Patent, either literally or under the doctrine of equivalents.

67. K.Mizra expects that discovery will reveal additional products that include Thread radios and infringe the '717 Patent, which may include additional Google Nest products and/or products that are not marketed under the Google Nest brand. For example, there are reports that at least some Google Pixel 9 devices may include a Thread radio. *See, e.g.,* "Pixel 9 series arrives at the FCC, includes Thread," (available at <https://9to5google.com/2024/07/12/pixel-9-fcc/> (accessed March 2026 and incorporated by reference)).

C. Google's Knowledge of the Asserted Patents

68. Google has been aware of K.Mizra's patent portfolio, including the Asserted Patents, based on discussions between the Parties in 2022 and early 2023.

69. Google has had actual knowledge of the Asserted Patents and its infringement of the Asserted Patents since at least May 23, 2025. On May 23, 2025, K.Mizra sent Google a

proposed amended complaint in *K.Mizra LLC v. Google LLC*, Case No. 1:25-cv-00236-ADA (W.D. Tex.). The proposed amended complaint included allegations of infringement of the Asserted Patents.

V. FIRST CLAIM FOR RELIEF
(Patent Infringement Under 35 U.S.C. § 271 of the '120 Patent)

70. K.Mizra incorporates paragraphs 1 through 69 as though fully set forth herein.

71. The '120 Patent includes 16 claims.

72. Google has directly infringed one or more claims of the '120 Patent by making, importing, using, offering for sale, and/or selling the Accused Hyperparameter Tuning Instrumentalities, in violation of 35 U.S.C. § 271(a).

73. Based on publicly available information, the Accused Hyperparameter Tuning Instrumentalities satisfy every limitation of at least Claim 1 of the '120 Patent.

74. Claim 1 of the '120 Patent states:

[preamble] A method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

[A] drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

[B] updating the estimate of the target hyperparameter vector by using the random sample, and

[C] selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

75. Regarding the preamble of Claim 1, and to the extent the preamble is determined to be limiting, the Accused Hyperparameter Tuning Instrumentalities provide the features described in the preamble, which recites "A method of determining hyperparameters." For

example, Google touts that its Vertex AI Platform is a "fully-managed, unified AI development platform for building and using generative AI."; "Custom training gives you complete control over the training process, including using your preferred ML framework, writing your own training code, and choosing hyperparameter tuning options."; and "Hyperparameter tuning jobs run multiple trials of your training application using different hyperparameter values."



Vertex AI Platform

Innovate faster with enterprise-ready AI, enhanced by Gemini models

Vertex AI is a fully-managed, unified AI development platform for building and using generative AI. Access and utilize [Vertex AI Studio](#), [Agent Builder](#), and 200+ foundation models.

- **Custom training gives you complete control over the training process, including using your preferred ML framework, writing your own training code, and choosing hyperparameter tuning options.**

"Innovate faster with enterprise-ready AI, enhanced by Gemini models," (available at <https://cloud.google.com/vertex-ai?hl=en> (accessed March 2026 and incorporated by reference)).

Hyperparameter optimization



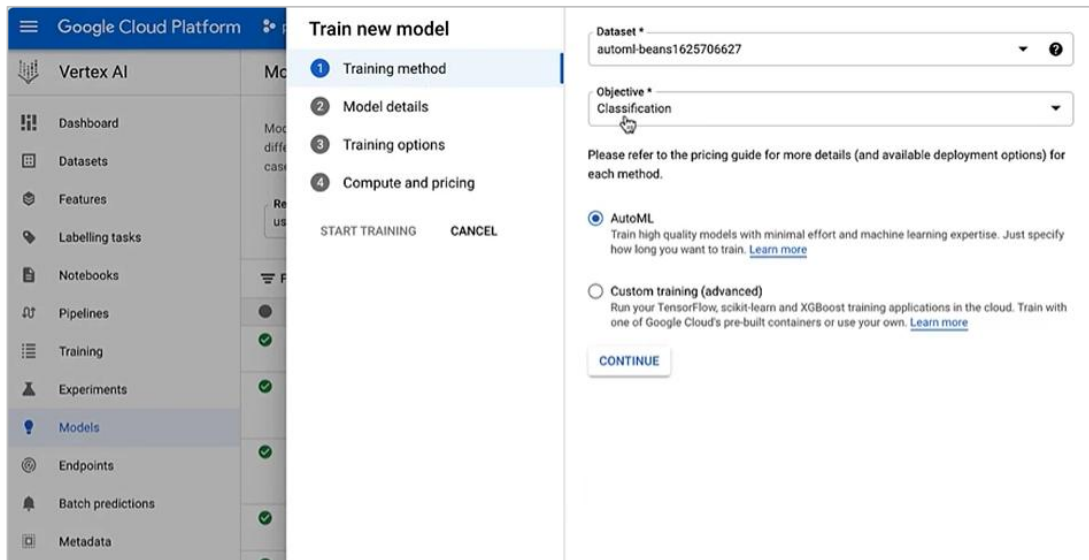
Hyperparameter tuning jobs run multiple trials of your training application using different hyperparameter values. You specify a range of values to test, and Vertex AI discovers the optimal values for your model within that range.

"Vertex AI custom training overview," (available at <https://cloud.google.com/vertex-ai/docs/training/overview> (accessed March 2026 and incorporated by reference)). Accordingly, and to the extent the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

76. The preamble also includes "of a classifier in a machine learning system".

Vertex AI is a machine learning (ML) platform that lets you train and deploy ML models and AI applications. Vertex AI combines data engineering, data science, and ML engineering workflows, which lets teams collaborate using a common toolset.

"Vertex AI documentation," (available at <https://docs.cloud.google.com/vertex-ai/docs> (accessed March 2026 and incorporated by reference)).



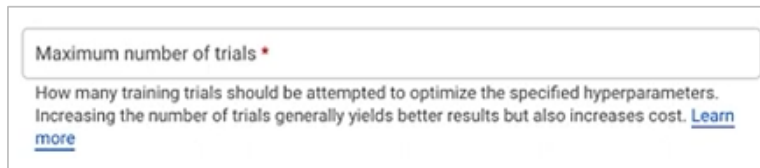
See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss at 6:30 (cropped) (accessed March 2026 and incorporated by reference). Thus, and again only to the extent that this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

77. The preamble further includes "by iteratively producing an estimate of a target hyperparameter vector":

How hyperparameter tuning works

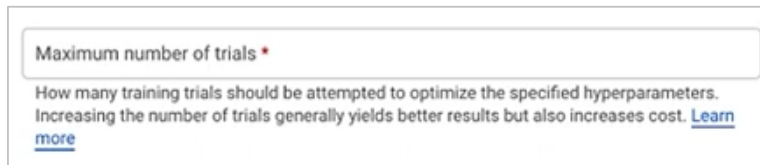
Hyperparameter tuning works by running multiple *trials* of your training application with values for your chosen hyperparameters, set within limits you specify. Vertex AI keeps track of the results of each trial and makes adjustments for subsequent trials. When the job is finished, you can get a summary of all the trials along with the most effective configuration of values according to the criteria you specify.

"Overview of hyperparameter tuning," (available at <https://docs.cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview> (accessed March 2026 and incorporated by reference)).



See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss at 7:13 (cropped) (accessed March 2026 and incorporated by reference). Accordingly, and to the extent this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

78. The preamble further also includes "each iteration comprising the steps of:"



Id. Again, then, to the extent this portion of the preamble of Claim 1 is deemed limiting, the Accused Hyperparameter Tuning Instrumentalities meet the limitation.

79. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet limitation of the preamble of Claim 1.

80. Limitations A and B of Claim 1 require:

[A] drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors,

[B] updating the estimate of the target hyperparameter vector by using the random sample, and

Google has disclosed:

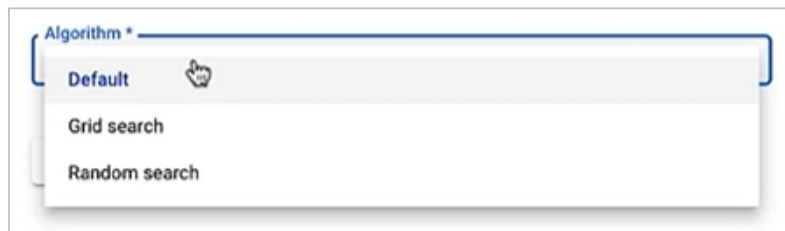
Search algorithms

You can specify a search algorithm in the `StudySpec` object. If you don't specify an algorithm, your job uses the default Vertex AI algorithm. The default algorithm applies Bayesian optimization to arrive at the optimal solution with a more effective search over the parameter space.

Available values:

- `ALGORITHM_UNSPECIFIED` : Same as not specifying an algorithm. Vertex AI chooses the best search algorithm between Gaussian process bandits, linear combination search, or their variants.
- `GRID_SEARCH` : A grid search within the feasible space. This option is particularly useful if you want to specify a quantity of trials that is greater than the number of points in the feasible space. In such cases, if you don't specify a grid search, the Vertex AI default algorithm may generate duplicate suggestions. To use grid search, all parameters must be of type `INTEGER`, `CATEGORICAL`, or `DISCRETE`.
- `RANDOM_SEARCH` : A random search within the feasible space.

"Overview of hyperparameter tuning," (available at <https://docs.cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview> (accessed March 2026 and incorporated by reference)).



See screenshot above of page at https://www.youtube.com/watch?v=8hZ_cBwNOss at 8:43 (cropped) (accessed March 2026 and incorporated by reference).

81. Limitation A requires "drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors" and the Google system meets this requirement:

```

StudySpec studySpec =
    StudySpec.newBuilder()
        .addMetrics(metric)
        .addParameters(parameter)
        .setAlgorithm(StudySpec.Algorithm.RANDOM_SEARCH)
        .build();
CustomJobSpec trialJobSpec =
    CustomJobSpec.newBuilder().addWorkerPoolSpecs(workerPoolSpec).build();
// hyperparameter_tuning_job
HyperparameterTuningJob hyperparameterTuningJob =
    HyperparameterTuningJob.newBuilder()
        .setDisplayName(displayName)
        .setMaxTrialCount(4)
        .setParallelTrialCount(2)
        .setStudySpec(studySpec)
        .setTrialJobSpec(trialJobSpec)
        .build();

```

"Create a hyperparameter tuning job for python package," (available at <https://docs.cloud.google.com/vertex-ai/docs/samples/aipatform-create-hyperparameter-tuning-job-python-package-sample?hl=en> (accessed March 2026 and incorporated by reference)).

82. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet Limitation A of Claim 1.

83. Limitation B requires "updating the estimate of the target hyperparameter vector by using the random sample". The use of random search meets this limitation:

Random search: set up a grid of hyperparameter values and select *random* combinations to train the model and score. The number of search iterations is set based on time/resources.

Random search is surprisingly efficient compared to grid search. Although grid search will find the optimal value of hyperparameters (assuming they are in your grid) eventually, random search will usually find a "close-enough" value in far fewer iterations. [This great paper explains why this is so](#): grid search spends too much time evaluating unpromising regions of the hyperparameter search space because it has to evaluate every single combination in the grid. Random search in contrast, does a better job of exploring the search space and therefore can usually find a good combination of hyperparameters in far fewer iterations.

"Intro to Model Tuning: Grid and Random Search," (available at <https://www.kaggle.com/code/willkoehrsen/intro-to-model-tuning-grid-and-random-search> (accessed March 2026 and incorporated by reference)).

Randomized Search for Hyperparameter Estimation

Randomized search is another method for hyperparameter optimization that can be more efficient than grid search in some cases. With randomized search, instead of specifying a list of values for each hyperparameter, you specify a distribution for each hyperparameter. The randomized search algorithm will then sample values for each hyperparameter from its corresponding distribution and train a model using the sampled values. This process is repeated a specified number of times, and the optimal values for the hyperparameters are chosen based on the performance of the models.

"Comparing Randomized Search and Grid Search for Hyperparameter Estimation in Scikit Learn," (available at <https://www.geeksforgeeks.org/comparing-randomized-search-and-grid-search-for-hyperparameter-estimation-in-scikit-learn/> (accessed March 2026 and incorporated by reference)).

84. Accordingly, the Accused Hyperparameter Tuning Instrumentalities also meet Limitation B of Claim 1.

85. Limitation C requires "selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result." Google describes:

How hyperparameter tuning works

Hyperparameter tuning works by running multiple *trials* of your training application with values for your chosen hyperparameters, set within limits you specify. Vertex AI keeps track of the results of each trial and makes adjustments for subsequent trials. When the job is finished, you can get a summary of all the trials along with the most effective configuration of values according to the criteria you specify.

"Overview of hyperparameter tuning," (available at <https://docs.cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview> (accessed March 2026 and incorporated by reference)).

86. Accordingly, the Accused Hyperparameter Tuning Instrumentalities meet Limitation C of Claim 1.

87. Additionally, and/or alternatively, Google has indirectly infringed and continues to indirectly infringe one or more claims of the '120 Patent, in violation of 35 U.S.C. § 271(b) by actively inducing users of the Accused Hyperparameter Tuning Instrumentalities to directly infringe one or more claims of the '120 Patent. For example, (a) Google had actual knowledge of or was willfully blind to the existence of the '120 Patent and its infringement thereof no later than its receipt of the proposed amended complaint alleging infringement of the '120 Patent in May 2025, and (b) Google intentionally causes, urges, or encourages users of the Accused Hyperparameter Tuning Instrumentalities to take action that, when taken, directly infringes one or more claims of the '120 Patent. Google's encouragement is accomplished by promoting, advertising, and instructing customers and potential customers to use the Accused Hyperparameter Tuning Instrumentalities, including infringing uses thereof. Google knows (based at least on the proposed amended complaint provided in May 2025 and/or after reading this Complaint) that its actions will induce users of the Accused Hyperparameter Tuning Instrumentalities to directly

infringe one or more claims of the '120 Patent, and users thereof directly infringe one or more claims of the '120 Patent. For instance, at a minimum, Google has supplied and continues to supply the Accused Hyperparameter Tuning Instrumentalities to customers while knowing that use thereof will infringe one or more claims of the '120 Patent.

88. Due to Google's knowledge of the '120 Patent and its infringement thereof no later than May 2025, Google's infringement of the '120 Patent has been and continues to be willful since no later than that date.

89. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

90. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

VI. SECOND CLAIM FOR RELIEF
(Patent Infringement Under 35 U.S.C. § 271 of the '717 Patent)

91. K.Mizra incorporates paragraphs 1 through 90 as though fully set forth herein.

92. The '717 Patent includes 20 claims.

93. Google has directly infringed one or more claims of the '717 Patent by making, importing, using, offering for sale, and/or selling the Accused Mesh Networking Instrumentalities, in violation of 35 U.S.C. § 271(a).

94. Based on publicly available information, the Accused Mesh Networking Instrumentalities satisfy every limitation of at least Claim 11 of the '717 Patent.

95. Open Thread is an open-source implementation of the Thread networking protocol released by Google. A description of Open Thread is provided at www.openthread.io. On information and belief, openthread.io is owned and operated by Google.

96. On information and belief, Open Thread is an implementation of the Thread networking technology used in Google Nest products.



OpenThread released by Google is an open-source implementation of Thread®. Google has released OpenThread to make the networking technology used in Google Nest products more broadly available to developers, in order to accelerate the development of products for the connected home and commercial buildings.

"OpenThread" (available at <https://openthread.io/> (accessed March 2026 and incorporated by reference)).

97. Claim 11 of the '717 Patent states:

[preamble] A station for use in a wireless communication network, the station comprising:

[A] a transmitter arrangement, a receiver arrangement, and a processor circuit coupled to the transmitter arrangement and the receiver arrangement,

[B] the processor configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state,

[C] the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages,

[D] the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit,

[E] the associated route including a source of the association grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or association grants for intermediate stations,

[F] the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

98. Regarding the preamble of Claim 11, and to the extent the preamble is determined to be limiting, the Accused Mesh Networking Instrumentalities provide the features described in the preamble, which recites "A station for use in a wireless communication network." For example, Google devices including the Nest WiFi, the Nest Hub Max, and the Nest Hub (2nd Gen) are stations configured for use in a wireless network implementing the Thread networking protocol.



A more reliable smart home network with Thread.

Thread is the latest technology for keeping all your smart home devices connected. Nest Hub Max and Nest Hub (2nd gen) have a built-in Thread border router to connect your Thread mesh network of smart devices to your Wi-Fi network. Your Thread-enabled network will be faster, use less energy, and reach more places throughout and around your home.

"Nest Hub Smart Displays for your Home – Google Store," (available at https://store.google.com/category/nest_hubs_displays (accessed March 2026 and incorporated by reference)).

Thread devices join users' existing home networks through a Thread Border Router. Just like a Wi-Fi router can bridge Wi-Fi and Ethernet devices into a single network, a Thread Border Router allows Thread devices to become part of users' networks.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

"Thread Play Services APIs," (available at <https://developers.home.google.com/matter/thread> (accessed March 2026 and incorporated by reference)).

99. Similarly, the Nest Connect is a station configured for use in a Thread wireless network.

In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

"Nest debuts Nest Secure home system and has a Thread router called Nest Connect," (available at <https://staceyoniot.com/nest-debuts-nest-secure-home-system-and-has-a-thread-router-called-nest-connect/> (accessed March 2026 and incorporated by reference)).

100. Accordingly, and to the extent the preamble of Claim 11 is deemed limiting, the Accused Mesh Networking Instrumentalities meet the limitation.

101. Limitation A of Claim 11 of the '717 Patent requires a transmitter arrangement, a receiver arrangement, and a processor circuit coupled to the transmitter arrangement and the receiver arrangement.

102. The Accused Mesh Networking Instrumentalities include a Thread radio with a transmitter arrangement and a receiver arrangement, as well as a processor circuit coupled to the transmitter arrangement and receiver arrangement that processes transmissions and receptions. For example, Google's documents confirm that each of the Nest WiFi, the Google Hub Max, and the Google Nest Hub (2nd Gen) include Thread radios.

Thread has the following key benefits:

- IPv6 based: Thread devices can join the same network as your other devices, and talk directly to each other and the cloud.
- Low-powered mesh: Built for IoT, Thread supports battery operated devices, with a mesh that delivers range and reliability.
- Speed: Low overhead, local connectivity, and mesh makes Thread devices extremely responsive.

Thread is developed in the Thread Group, which Google founded with other member companies in 2014.

Thread devices join users' existing home networks through a Thread Border Router. Just like a Wi-Fi router can bridge Wi-Fi and Ethernet devices into a single network, a Thread Border Router allows Thread devices to become part of users' networks.

Google devices like the Nest WiFi, Google Nest Hub Max and Google Nest Hub (2nd gen) have Thread radios built-into them and act as Thread Border Routers.

"Thread Play Services APIs," (available at <https://developers.home.google.com/matter/thread> (accessed March 2026 and incorporated by reference)). Similarly, a statement from the Thread Group (of which Google is a member) confirms that the Nest Connect is a Thread router.

In an emailed statement, Grant Erickson, president of the Thread Group said, "Nest Detect relies on Thread's low power architecture to last for long periods of time on one battery. Nest Connect is a Thread router, which seamlessly extends both the reach and connectivity of the Thread network. Nest Guard provides seamless IP-based routing between Thread, Wi-Fi, and cellular, and their interoperability with the Yale Linus Lock is the posterchild of Thread's benefits in action."

"Nest debuts Nest Secure home system and has a Thread router called Nest Connect," (available at <https://staceyoniot.com/nest-debuts-nest-secure-home-system-and-has-a-thread-router-called-nest-connect/> (accessed March 2026 and incorporated by reference)). Indeed, in order to communicate with other devices, each Thread device (including the Accused Mesh Networking Instrumentalities) must necessarily include a transmitter arrangement (to transmit messages from

the Thread device to other devices in the Thread network) and a receiver arrangement (to receive communications, including responses to its transmitted messages, from other devices in the Thread network). Thread devices, including the Accused Mesh Networking Instrumentalities, must also include a processor coupled to both the transmitter arrangement and the receiver arrangement to, among other things, initiate transmissions, respond to messages sent by other devices in the Thread network, and process received messages. The operations of the processor are described in further detail in connection with Limitations B through F.

103. Accordingly, the Accused Mesh Instrumentalities meet Limitation A of Claim 11.

104. Limitation B of Claim 11 of the '717 Patent requires the processor is configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state.

105. The Accused Mesh Networking Instrumentalities include this feature. For example, Thread devices, such as the Accused Mesh Networking Devices, have a Detached state when the Thread device has not yet attached to a Thread network.

Note: The Detached state is observed when Thread is enabled but the device is not attached to a Thread network. The Disabled state is observed when Thread is disabled on the device.

"Developing with OpenThread APIs, (available at <https://openthread.io/codelabs/openthread-apis#1> (accessed March 2026 and incorporated by reference)).

106. In order to attach to a Thread network, a Thread device such as the Accused Mesh Networking Instrumentalities goes through the Attach process.

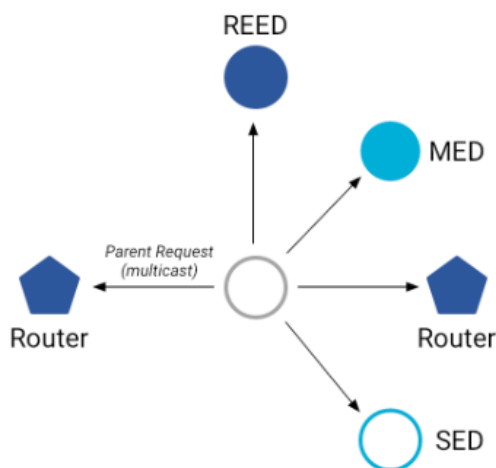
Join an existing network

If the device elects to join an existing network, it configures its Channel, PAN ID, XPAN ID, and Network Name to match that of the target network via Thread Commissioning, then goes through the MLE Attach process to attach as a Child (End Device). This process is used for Child-Parent links.

"Network Discovery and Formation," (available at <https://openthread.io/guides/thread-primer/network-discovery> (accessed March 2026 and incorporated by reference)). As the first step in this process, an attaching device sends a Parent Request message to find neighboring devices that are part of the Thread network.

1. Parent Request

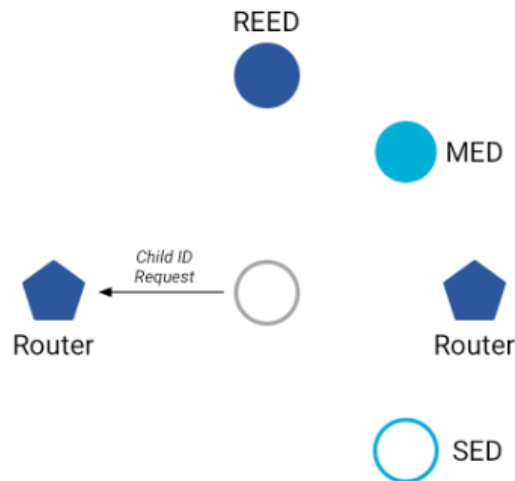
A Parent Request is a multicast request from the attaching device that is used to discover neighboring Routers and Router Eligible End Devices (REEDs) in the target network.



Id. Upon receiving a response, the Thread device sends a message called the Child ID Request to a particular device in the Thread network for the purpose of establishing a link with that device to join the network.

3. Child ID Request

A Child ID Request is a unicast request from the attaching device (Child) that is sent to the Router or REED (Parent) for the purpose of establishing a Child-Parent link. If the request is sent to a REED, it [upgrades itself to a Router](#) before accepting the request.



Id. The attaching device may also be referred to as the Child, and the router with which the attaching device is seeking to establish a link may be referred to as the Parent.

107. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation B of Claim 11.

108. Limitation C of Claim 11 of the '717 Patent requires the processor circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages.

109. Thread devices, such as the Accused Mesh Networking Instrumentalities, become attached to a Thread network through the MLE Attach process. As described above in connection with Limitation B, the attaching device begins the process by sending a Parent Request and then exchanges further messages in order to establish a link with a device in the existing Thread network.

Join an existing network

If the device elects to join an existing network, it configures its Channel, PAN ID, XPAN ID, and Network Name to match that of the target network via Thread Commissioning, then goes through the MLE Attach process to attach as a Child (End Device). This process is used for Child-Parent links.



Key Point: Every device, router-capable or not, initially attaches to a Thread network as a Child (End Device).

1. The Child sends a multicast **Parent Request** to all neighboring Routers and REEDs in the target network.
2. All neighboring Routers and REEDs (if the Parent Request Scan Mask includes REEDs) send **Parent Responses** with information about themselves.
3. The Child chooses a Parent device and sends a **Child ID Request** to it.
4. The Parent sends a **Child ID Response** to confirm link establishment.

Id. When the link has been established, the Thread device has attached to the network and thus is in an attached state. The establishment of the link between the attaching device and the Thread device in the existing Thread network is confirmed by the Child ID Response.

4. Child ID Response

A Child ID Response is a unicast response from the Parent that is sent to the Child to confirm that a Child-Parent link has been established.

Id. Notably, Thread devices such as the Accused Mesh Networking Instrumentalities may exchange information about how long they have been in certain states or roles, suggesting that these devices track their current state/role and when each state/role is entered.

<code>mChildRole</code>	<code>uint16_t</code> Number of times device entered child role.
<code>mChildTime</code>	<code>uint64_t</code> Milliseconds device has been in child role.
<code>mDetachedRole</code>	<code>uint16_t</code> Number of times device entered detached role.
<code>mDetachedTime</code>	<code>uint64_t</code> Milliseconds device has been in detached role.

"otNetworkDiagMleCounters," (available at <https://openthread.io/reference/struct/ot-network-diag-mle-counters> (accessed March 2026 and incorporated by reference)).

110. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation C of Claim 11.

111. Limitation D of Claim 11 of the Accused Mesh Networking Instrumentalities requires the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit.

112. Each of the Accused Mesh Networking Instrumentalities is configured to operate in a Thread network. A Thread network includes a device designated as the Leader.

Thread Leader

The Thread Leader is a Router that is responsible for managing the set of Routers in a Thread network. It is dynamically self-elected for fault tolerance, and aggregates and distributes network-wide configuration information.

"Node Roles and Types," (available at <https://openthread.io/guides/thread-primer/node-roles-and-types> (accessed March 2026 and incorporated by reference)). The processors of the Accused Mesh

Networking Devices are configured to store information about the route between the attaching device and the Leader as further discussed below in connection with Limitation E of Claim 11.

113. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation D of Claim 11.

114. Limitation E of Claim 11 requires the associated route includes a source of the association grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or association grants for intermediate stations.

115. Upon joining a Thread network, Thread devices such as the Accused Mesh Networking Devices receive a Child ID Response which provides information about the route between the Leader and the Thread device. For example, the Child ID Response includes the Source Address (i.e., the address for the Parent Thread device), the Leader RLOC (i.e., the address for the Leader of the Thread network), Network Data such as "more-specific routes," and Route information.

Child ID Response Message Contents	
Source Address	Parent's RLOC16
Address16	Child's RLOC16
Leader Data	Information about the Parent's Leader (RLOC, Partition ID, Partition weight)
Network Data	Information about the Thread network (on-mesh prefixes, address autoconfiguration, more-specific routes)
Route (REED only)	Route propagation
Timeout	Inactivity duration before the Parent removes the Child
Address Registration (MEDs and SEDs only)	Confirm registered addresses

"Network Discovery and Formation," (available at <https://openthread.io/guides/thread-primer/network-discovery> (accessed March 2026 and incorporated by reference)). K.Mizra further expects that discovery, including review of source code, will provide further information and confirm this understanding.

116. Accordingly, the Accused Mesh Networking Instrumentalities meet Limitation D of Claim 11.

117. Limitation E of Claim 11 of the '717 Patent requires the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

118. Thread devices such as the Accused Mesh Networking Instrumentalities can send Parent Response and Child ID Response communications once attached to the Thread network, but will not send such responses if the device is disconnected from the network. For example, upon joining a Thread network, Router Eligible End Devices (or REEDs) can upgrade to become routers.

Upgrade to a Router

After attaching to a Thread network, the Child device may elect to become a Router. Before initiating the MLE Link Request process, the Child sends an Address Solicit message to the Leader, asking for a Router ID. If the Leader accepts, it responds with a Router ID and the Child upgrades itself to a Router.

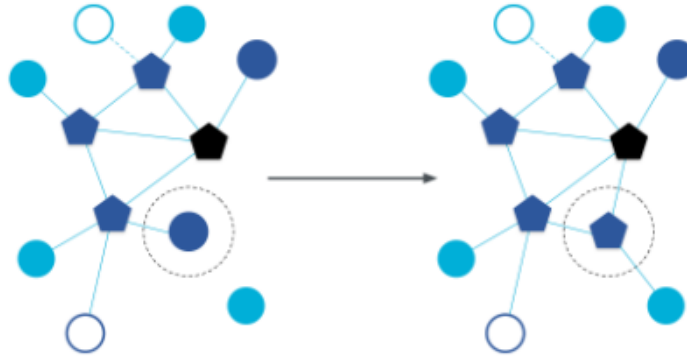
The MLE Link Request process is then used to establish bi-directional Router-Router links with neighboring Routers.

1. The new Router sends a multicast [Link Request](#) to neighboring Routers.
2. Routers respond with [Link Accept and Request](#) messages.
3. The new Router responds to each Router with a unicast [Link Accept](#) to establish the Router-Router link.

"Router Selection," (available at <https://openthread.io/guides/thread-primer/router-selection> (accessed March 2026 and incorporated by reference)).

Upgrading and downgrading

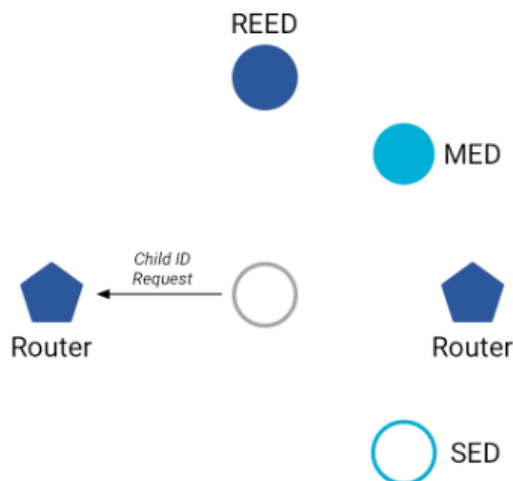
When a REED is the only node in reach of a new End Device wishing to join the Thread network, it can upgrade itself and operate as a Router:



"Node Roles and Types," (available at <https://openthread.io/guides/thread-primer/node-roles-and-types> (accessed March 2026 and incorporated by reference)). Upon upgrading to a router, the Thread device is capable of accepting a request from an attaching device to join the Thread network.

3. Child ID Request

A Child ID Request is a unicast request from the attaching device (Child) that is sent to the Router or REED (Parent) for the purpose of establishing a Child-Parent link. If the request is sent to a REED, it [upgrades itself to a Router](#) before accepting the request.



"Network Discovery and Formation," (available at <https://openthread.io/guides/thread-primer/network-discovery> (accessed March 2026 and incorporated by reference)).

119. Accordingly, the Accused Mesh Networking Instrumentalities meet limitation of the Limitation E of Claim 11.

120. Additionally, and/or alternatively, Google has indirectly infringed and continues to indirectly infringe one or more claims of the Mesh Networking Patent, in violation of 35 U.S.C. § 271(b) by actively inducing users of the Accused Mesh Networking Instrumentalities to directly infringe one or more claims of the Mesh Networking Patent. For example, (a) Google had actual knowledge of or was willfully blind to the existence of the Mesh Networking Patent and its infringement thereof no later than its receipt of the proposed amended complaint alleging infringement of the Mesh Networking Patent in May 2025, and (b) Google intentionally causes, urges, or encourages users of the Accused Mesh Networking Instrumentalities to take action that,

when taken, directly infringes one or more claims of the Mesh Networking Patent. Google's encouragement is accomplished by promoting, advertising, and instructing customers and potential customers to use the Accused Mesh Networking Instrumentalities, including infringing uses thereof. Google knows (based at least on the proposed amended complaint provided in May 2025 and/or after reading this Complaint) that its actions will induce users of the Accused Mesh Networking Instrumentalities to directly infringe one or more claims of the Mesh Networking Patent, and users thereof directly infringe one or more claims of the Mesh Networking Patent. For instance, at a minimum, Google has supplied and continues to supply the Accused Mesh Networking Instrumentalities to customers while knowing that use thereof will infringe one or more claims of the Mesh Networking Patent.

121. Due to Google's knowledge of the Mesh Networking Patent and its infringement thereof no later than May 2025, Google's infringement of the Mesh Networking Patent has been and continues to be willful since no later than that date.

122. Google's acts of infringement have occurred within this District and elsewhere throughout the United States.

123. As a result of Google's infringing conduct, K.Mizra has suffered damages. Google is liable to K.Mizra in an amount that adequately compensates K.Mizra for Google's infringement in an amount that is no less than a to-be-calculated fully paid-up, lump-sum, reasonable royalty, together with interest and costs as fixed by this Court under 25 U.S.C. § 284.

VII. REQUEST FOR RELIEF

WHEREFORE, K.Mizra respectfully requests the Court find in its favor and against Google, and that the Court grant K.Mizra at least the following relief:

A. Judgment that Google has infringed, literally and/or under the doctrine of equivalents, one or more claims of the Asserted Patents;

B. Awarding damages to K.Mizra in an amount to be proven at trial and in the form of a fully paid-up, lump sum, reasonable royalty that considers and runs through expiration of all the Asserted Patents;

C. Awarding enhanced damages, as appropriate, under 35 U.S.C. § 284;

D. Awarding K.Mizra's costs (including internal and external costs and disbursements) and declaring this an exceptional case and awarding K.Mizra its attorneys' fees in accordance with 35 U.S.C. § 285;

E. Pre-judgment and post-judgment interest at the maximum rate permitted by law on the damages caused by reason of Google's infringing activities and other conduct complained of herein; and

F. Awarding such other and further relief as this Court deems just and proper under the circumstances.

VIII. DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38(b), K.Mizra hereby demands a trial by jury on all issues so triable.

Dated: March 27, 2026

Respectfully submitted,

By: /s/ Michael C. Smith

Michael C. Smith
Texas Bar No. 18650410
michael.smith@solidcounsel.com
Scheef & Stone, LLP
113 E. Austin Street
Marshall, TX 75670
(903) 938-8900

Robert R. Brunelli*

CO State Bar No. 20070
rbrunelli@sheridanross.com
Matthew C. Holohan*
CO State Bar No. 40996
mholohan@sheridanross.com
Bart A. Starr*
CO State Bar No. 50446
bstarr@sheridanross.com
Brian S. Boerman*
CO State Bar No. 50834
bboerman@sheridanross.com
Tristan D. Lewis*
CO State Bar No. 60968
tlewis@sheridanross.com
SHERIDAN ROSS P.C.
1560 Broadway, Suite 1200
Denver, CO 80202
Telephone: 303-863-9700
Facsimile: 303-863-0223
litigation@sheridanross.com

Of Counsel:

Claire A. Henry
Texas Bar No. 24053063
Miller Fair Henry PLLC
1507 Bill Owens Parkway
Longview, TX 75604
Telephone: (903) 757-6400
Facsimile: (903) 757-2323
claire@millerfairhenry.com

**To be admitted pro hac vice
Attorneys for Plaintiff K.Mizra LLC*

EXHIBIT 1



US008438120B2

(12) **United States Patent**
Raaijmakers

(10) **Patent No.:** **US 8,438,120 B2**

(45) **Date of Patent:** **May 7, 2013**

(54) **MACHINE LEARNING HYPERPARAMETER ESTIMATION**

(56) **References Cited**

(75) Inventor: **Stephan Alexander Raaijmakers**,
Amsterdam (NL)

PUBLICATIONS

(73) Assignee: **Nederlandse Organisatie voor
toegepast-natuurwetenschappelijk
Onderzoek TNO**, Delft (NL)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 607 days.

Duckstein, et al., Bayesian forecasting of hydrologic variables under
changing climatology, the Influence of Climate Change and Climatic
Variability on the Hydrologic Regime and Water Resources (Pro-
ceedings of the Vancouver Symposium, Aug. 1987). IAHSPubl. No.
168, 1987, pp. 301-311.*

Raaijmakers, "Sentiment classification with interpolated information
diffusion kernels," Proceedings of the 1st International Workshop on
Data Mining and Audience Intelligence for Advertising, pp. 34-39,
(Aug. 12, 2007).

AMIDA Project: "Chapter 3: subjectivity recognition" Deliverable
5.2: Multimodal Content Abstraction, 2007, pp. 28-47.

De Boer et al., "A Tutorial on the Cross-Entropy Method," Annals of
Operating Research, vol. 134, No. 1, (Feb. 2005), pp. 19-67.

Mannor et al., "The cross entropy method for classification," Pro-
ceedings of the 22nd International Conference on Machine Learning
(Aug. 9, 2005) (the whole document).

International Search Report dated Oct. 9, 2008 for PCT/NL2008/
050247.

* cited by examiner

(21) Appl. No.: **12/597,257**

(22) PCT Filed: **Apr. 25, 2008**

(86) PCT No.: **PCT/NL2008/050247**

§ 371 (c)(1),

(2), (4) Date: **Jul. 8, 2010**

(87) PCT Pub. No.: **WO2008/133509**

PCT Pub. Date: **Nov. 6, 2008**

(65) **Prior Publication Data**

US 2010/0280979 A1 Nov. 4, 2010

(30) **Foreign Application Priority Data**

Apr. 25, 2007 (EP) 07106963

Jul. 9, 2007 (EP) 07112037

(51) **Int. Cl.**
G06N 5/00 (2006.01)

(52) **U.S. Cl.**
USPC 706/12; 706/45

(58) **Field of Classification Search** 706/12,
706/45

See application file for complete search history.

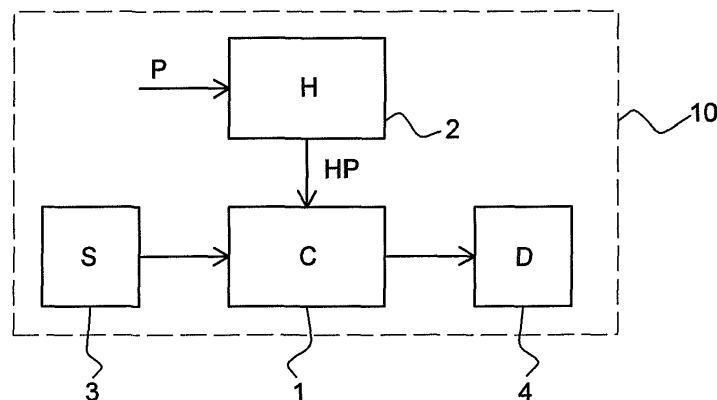
Primary Examiner — Wilbert L Starks

(74) *Attorney, Agent, or Firm* — Leydig, Voit & Mayer, Ltd.

(57) **ABSTRACT**

A method of determining hyperparameters (HP) of a classi-
fier (1) in a machine learning system (10) iteratively produces
an estimate of a target hyperparameter vector. The method
comprises the steps of selecting from the random sample the
hyperparameter vector producing the best result in the present
and any previous iterations, and updating the estimate of the
target hyperparameter vector by using said selected hyperpa-
rameter vector. The random sample may be restricted by
using the hyperparameter vector producing the best result in
the present and any previous iterations.

16 Claims, 1 Drawing Sheet



U.S. Patent

May 7, 2013

US 8,438,120 B2

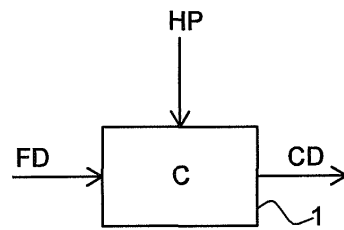


FIG. 1

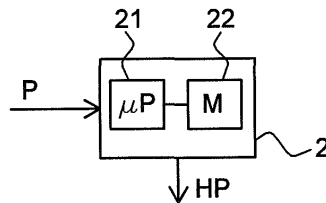


FIG. 2

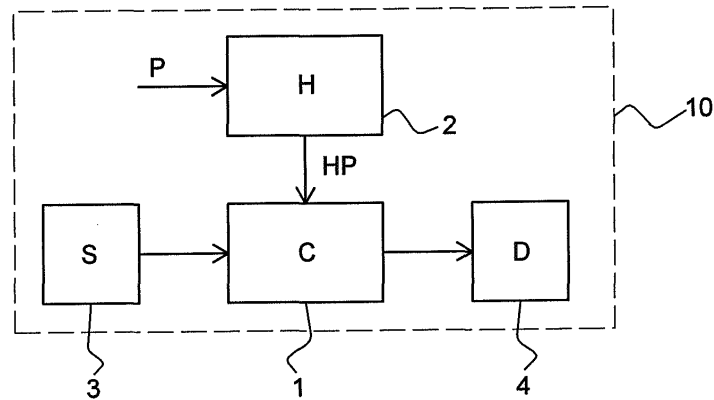


FIG. 3

US 8,438,120 B2

1

MACHINE LEARNING HYPERPARAMETER ESTIMATION**FIELD OF THE INVENTION**

The present invention relates to hyperparameter estimation. More in particular, the present invention relates to a method and device for determining hyperparameters of classifiers in machine learning systems and applications.

BACKGROUND

Classifiers are used in machine learning systems to classify physical objects and/or their (typically digital) representations. Machine learning systems may, for example, be used to assist fruit picking robots. The classifier of the machine learning system is trained to distinguish ripe fruit (e.g. tomatoes) from unripe fruit. For each fruit item, the classifier determines a set of parameters which are compared with stored parameters in order to classify the items as “ripe” or “unripe”. The classification process is, in turn, controlled by hyperparameters which determine, for example, decision criteria such as threshold values.

Classifiers are typically trained using a training set prepared on the basis of human input: an operator indicates the correct classification for the items of the training set. On the basis of these training data, the correct hyperparameters of the classifier can be estimated.

Hyperparameters may be determined using various methods, for example heuristic or statistical optimisation methods. However, many heuristic methods are ad hoc methods, while not all statistical optimisation methods are suitable for determining hyperparameters.

A particularly efficient yet relatively simple method for estimating parameters in general is the cross-entropy (CE) method. This iterative method comprises the repeated steps of drawing, in a parameterised way, a random sample of candidate solutions, and updating the parameters on the basis of the random sample. At the time of writing, the paper “A Tutorial on the Cross-Entropy Method” by P. T. de Boer et al. could be found at <http://few3.technion.ac.il/CE/tutor.php>, said paper is herewith incorporated in this document in its entirety.

The paper “The cross-entropy method for classification” by S. Mannor, D. Peleg and R. Rubinstein, Proceedings of the 22nd International Conference on Machine Learning, 2005, discloses an application of the CE algorithm for searching the space of support vectors. Hyperparameter values are determined by using a simple grid search, not by using the CE algorithm, for the reasons discussed below.

The cross-entropy method as described in the above-mentioned papers is unfortunately not suitable for determining hyperparameters. Determining optimal sets of hyperparameter values is a difficult problem due to the extremely large size of the search space: the optimal solution typically is a rare event. While the cross-entropy method is geared towards sampling rare event spaces, it is not a priori clear how the process of drawing hyperparameter values can be parameterised, as the hyperparameter samples are not classical parameterised probability density functions.

SUMMARY OF THE INVENTION

It is an object of the present invention to overcome these and other problems of the Prior Art and to provide a method and device for determining hyperparameters which allow efficient methods similar to the cross-entropy method to be utilized.

2

Accordingly, the present invention provides a method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, and

updating the estimate of the target hyperparameter vector by using the random sample,

the method being characterised in that each iteration comprises the step of:

selecting from the random sample a hyperparameter vector producing the best result in the present and any previous iterations,

and in that the step of updating the estimate of the target hyperparameter vector involves said selected hyperparameter vector.

By selecting from the random sample a or the hyperparameter vector producing the best result in the present and any previous iterations, and using said selected hyperparameter vector to update the estimate of the target hyperparameter vector, the properties of the best performing hyperparameters are used to guide the next iteration. In this way, a method similar to the cross-entropy method can be used effectively for determining hyperparameters, even when the hyperparameters are not continuous and their effects are not transparent. By using the method according to the present invention, an improved classification accuracy is achieved.

In the method of the present invention, hyperparameter vectors obtained through randomly sampling a set of hyperparameter vectors are used to estimate a desired (that is, target) hyperparameter vector to be used in the classifier. Each hyperparameter vector preferably comprises two or more elements, that is hyperparameters, but it is also possible for a hyperparameter vector to contain only a single hyperparameter. Although the samples are preferably completely random, pseudo-random samples may be used instead.

In a preferred embodiment, the method according to the present invention further comprises the steps of:

selecting from the random sample a further hyperparameter vector producing the best result in any previous iterations, and

restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.

The step of restricting the random sample preferably involves the use of an interval surrounding the further selected hyperparameter vector which produces the best result in any previous iteration.

It is preferred that the step of restricting the random sample is carried out prior to the step of determining the hyperparameter vector producing the best result in the present and any previous iterations. However, in alternative embodiments, the step of restricting the random sample is carried out during the step of updating the estimate of the target hyperparameter vector.

In a preferred embodiment, E^t is the hyperparameter vector X_i^t at iteration t producing the best result $S(X_i^t)$ in the present iteration t and all previous iterations (if any). The random sample X_i^t has elements X_{ij}^t , while the step of updating the

US 8,438,120 B2

3

hyperparameters comprises the step of determining the (target) hyperparameter v_j^* , where

$$v_j^* = \frac{\sum_{i=1}^n I\{S(X_i^t) \geq \gamma^t\} W(X_i^t; E^t) X_{ij}^t}{\sum_{i=1}^n I\{S(X_i^t) \geq \gamma^t\} W(X_i^t; E^t)},$$

wherein γ^t is a threshold value, W is a weighting function and $I\{S(X_i^t) \geq \gamma^t\}$ is an indicator function which is equal to 1 if $S(X_i^t)$ greater than or equal to the threshold value γ^t , and else is equal to 0. The weighting function W is preferably given by

$$W(X_i^t; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^m (X_{ij}^t - E_j^t)^2}}{\sqrt{\sum_{j=1}^m (X_{ij}^t)^2} \sqrt{\sum_{j=1}^m (E_j^t)^2}}.$$

Accordingly, the step of updating the hyperparameters may involve a weighting function based upon a distance function, preferably a Euclidean distance function. It is noted that when the Euclidean distance between E^t and X_{ij}^t is zero, the weighting function W is equal to one.

The method according to the present invention is typically carried out by computer apparatus, for example a general purpose computer system comprising a processor and an associated memory, one part of the memory storing a software program for instructing the processor to carry out the method steps of the present invention, and another part of the memory storing data, said data comprising the hyperparameter values referred to above.

The present invention also provides a computer program product for carrying out the method as defined above. A computer program product may comprise a set of computer executable instructions stored on a data carrier, such as a CD or a DVD. The set of computer executable instructions, which allows a programmable computer to carry out the method as defined above, may also be available for downloading from a remote server, for example via the Internet.

The present invention additionally provides a classifier for use in a machine learning system, the classifier being arranged for using hyperparameters as control parameters, wherein the hyperparameters have been determined by the method according to any of the preceding claims. The classifier is preferably embodied in software, but may also be embodied in hardware, or in a combination of hardware and software.

The present invention still further provides a device for determining hyperparameters of a classifier in a machine learning system, the device comprising a processor arranged for iteratively producing estimates of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, and

updating the estimate of the target hyperparameter vector by using the random sample,

the device being characterised in that the processor is arranged such that each iteration comprises the step of:

selecting from the random sample a hyperparameter vector

producing the best result in the present iteration and any previous iterations, and

4

in that the step of updating the estimate of the target hyperparameter vector involves said selected hyperparameter vector.

The processor may be arranged for carrying out the steps mentioned above by providing a suitable software program in the memory associated with the processor. Alternatively, or additionally, the processor may be designed specifically to carry out the steps mentioned above.

A machine learning system comprising a device as defined above and/or arranged for carrying out the method as defined above is also provided by the present invention. The machine learning system may comprise hardware and/or software components.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will further be explained below with reference to exemplary embodiments illustrated in the accompanying drawings, in which:

FIG. 1 schematically shows a classifier having hyperparameters as input control variables.

FIG. 2 schematically shows a device for determining hyperparameters according to the present invention.

FIG. 3 schematically shows a machine learning system according to the present invention.

DETAILED DESCRIPTION OF THE DRAWINGS

The classifier (C) 1 shown schematically in FIG. 1 receives feature data FD and hyperparameters HP, and produces classification data CD. The feature data FD may include image data produced by a CCD (Charge Coupled Device) array or other light-sensitive device. Additionally, the feature data FD may include sound data or data concerning database items, such as library data (book numbers and titles). The classifier may be constituted by an apparatus (hardware embodiment), but is preferably constituted by a software program running on a computer apparatus, for example a general purpose computer ("personal computer"), a dedicated computer, or other suitable apparatus.

The classifier 1 is capable of classifying the items represented by the input feature data FD. The feature data FD typically comprise parameters of the items, for example their size and/or colour in the case of fruit or machine parts. Under control of the hyperparameters HP the items are each assigned to a class, for example "ripe" in the case of fruit. At least two different classes are used, and typically an item is assigned to one class only.

The hyperparameters HP determine the decision function of the classifier that classifies items. For example, when this function is based upon thresholding, items producing an aggregate value (derived from the feature values) equal to or greater than the threshold are assigned to class A, while items producing a value smaller than the threshold are assigned to class B. It is noted that other classification techniques may be used as well.

Applications of the classifier 1 include picking fruit, sorting parts (e.g. machine parts), classifying books, selecting songs, recognising adult video material, detecting damage, etc.

The present invention allows a method similar to the cross-entropy (CE) method to be used for estimating and determining hyperparameters. To this end, the present invention proposes to include a memory facility into the algorithm by preserving samples which produce a good result (the result being defined by a suitable function, such as a loss function). Additionally, or alternatively, the sampling of candidate

US 8,438,120 B2

5

hyperparameter settings is made dependent on the (hyper) parameter vector updated by the algorithm, and/or a suitably weighting function (also known as “change of measure” or “likelihood ratio” in the CE algorithm) guiding the search process should be provided.

The method of the present invention uses parameters N , ρ and μ which are typically predetermined. It is noted that these parameters may be thought of as hyperparameters of the method but are distinct from the hyperparameters to be determined by the method. The parameter N represents the number of random draws; the parameter N may for example be equal to 10, 100 or 1000.

In the iterations a parameter n is used which represents the size of each data vector, that is, the number of elements X_{ij} of each hyperparameter vector X_i of the set of possible hyperparameter vectors. The parameter n may for example be equal to 5, 10 or 100, although other numbers are also possible. The parameter ρ represents a fraction and may have a value ranging from 0.2 to 0.01, although other values may also be used. The width parameter μ , which will be explained in more detail below, indicates a preferred sampling interval.

The initial set of target hyperparameters, the hyperparameter vector u , can typically be chosen arbitrarily. Each estimated (target) hyperparameter vector v^t at point in time (or iteration number) t consists of hyperparameters v_j^t .

In the present invention, the “best” hyperparameter vector v of iterations $1 \dots t$ is kept for use in the next iteration and is denoted E^t . This “best” (or “elitist”) hyperparameter vector is found using a function $S(X^t)$ which may be a loss function or any other suitable function producing a result value based upon X^t . The value of $S(X^t)$ may also be viewed as an importance measure. The value of X^t corresponding with the maximum value of $S(X^t)$ is denoted $\arg\max S(X^t)$. In addition, the values of $S(X^t)$ are ranked and are used to determine a threshold γ^t , involving ρ and N to determine the $((1-\rho)N)^{th}$ value of the ranked values of $S(X^t)$. It is noted that the number $((1-\rho)N)$ is not necessarily an integer and is therefore rounded up, expressed mathematically as $\lceil ((1-\rho)N) \rceil$.

Subsequently, the (target) hyperparameters are determined using an expression involving indicator functions $I\{S(X_i^t) \geq \gamma^t\}$ and weighting functions $W(X_i^t; E^t)$. The indicator functions are equal to 1 when $S(X_i^t) \geq \gamma^t$ and equal to 0 when $S(X_i^t) < \gamma^t$. The weighting functions $W(X_i^t; E^t)$, also known as “change of measure” functions, have the data vector elements X_i (actually X_i^t) and E^t as input variables. This implies that the best data vector E^t of the present iteration is used to determine the (target) hyperparameters.

The weighting function $W(X_i^t; E^t)$ is preferably defined as:

$$W(X_i^t; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^m (X_{ij}^t - E_j^t)^2}}{\sqrt{\sum_{j=1}^m (X_{ij}^t)^2} \sqrt{\sum_{j=1}^m (E_j^t)^2}} \quad (1)$$

This function produces a value of 1 when the Euclidean distance between a sample and the “best” sample is 0.

Accordingly, a preferred embodiment of the method according to the present invention comprises the following steps:

1. Choose initial parameters N , ρ and μ and generate an initial (target) hyperparameter vector u .
2. Let $v^0 = E^0 = u$ and set iteration number $t=1$.
3. Generate a random sample $X_1^t, \dots, X_n^t$ of hyperparameters drawn from a (given) set of possible hyperparameters. The

6

random sample preferably is restricted by v^{t-1} . Compute the performance $S(X_i^t)$ for every value of i and rank the results: $S_{(1)} \leq \dots \leq S_{(n)}$. Compute $\gamma^t = S_{\text{percentile}}$ and compute $E^t = \arg\max_{v_j=1 \dots n} X_j^t$.

4. For every (target) hyperparameter v_j^t let

$$v_j^t = \frac{\sum_{i=1}^n I\{S(X_i^t) \geq \gamma^t\} W(X_i^t; E^t) X_{ij}^t}{\sum_{i=1}^n I\{S(X_i^t) \geq \gamma^t\} W(X_i^t; E^t)} \quad (2)$$

5. Let $t=t+1$, repeat from step 3 until the stopping condition is met.

The final set v of j (target) hyperparameters v_j is the desired optimal set.

By using the best sample E_j^t in hyperparameter estimation/update formula (2), the better results are reinforced, leading to a more efficient convergence of the method. In addition, discontinuity problems associated with hyperparameters are overcome by the method of the present invention.

The stopping criterion (or condition) is preferably a non-changing value of the parameter γ^t during a number of iterations, but could also be a threshold value. The value $S_{\text{percentile}}$ is determined using the $((1-\rho)N)^{th}$ value of S in the ranking $S_{(1)} \leq \dots \leq S_{(n)}$, where rounding up may be used if this value is not an integer. The weighting function $W(X_i^t; E^t)$ used in formula (2) is preferably defined by formula (1) above.

The method of the present invention may be summarized as deriving a target hyperparameter vector from a random sample of hyperparameter vectors while taking the performance of the hyperparameters of the random sample into account.

The restriction which may be applied when generating a random sample at time t in step 3 may be defined by the interval:

$$[v_j^{t-1}(1-\mu), v_j^{t-1}(1+\mu)] \quad (3)$$

where μ is a width parameter and “*” denoted multiplication. If the sample X_i^t lies within this interval, its distance relative to the best sample E_j^t of the previous iteration is limited to the value of μ . It is noted that the use of this interval is preferred but not essential as the method of the present invention could be carried out without using this interval.

The device 2 for determining hyperparameters comprises a processor (μP) 21 and an associated memory (M) 22. The memory stores both instructions for the processor and data. The device 2 receives parameters P (more in general: feature data), such as the parameters ρ and μ mentioned above, and outputs hyperparameters HP . The stored instructions allow the processor to carry out the method according to the present invention.

The machine learning system 10 shown merely by way of non-limiting example in FIG. 3 comprises a classifier (C) 1 and a device (H) 2 for determining hyperparameters. The machine learning system 10 may further comprise at least one sensor (S) 3 and a display screen (D) 4. The sensor 3 serves to collect data while the display screen 4 serves to display classification result and other data.

The present invention is based upon the insight that methods similar to the cross-entropy method can be adapted for hyperparameter estimation by adding a best performance preservation feature. The present invention benefits from the additional insight that the best performance in a particular iteration may be preserved by restricting the sample using the best performance of the previous iterations.

US 8,438,120 B2

7

It is noted that any terms used in this document should not be construed so as to limit the scope of the present invention. In particular, the words “comprise(s)” and “comprising” are not meant to exclude any elements not specifically stated. Single (circuit) elements may be substituted with multiple (circuit) elements or with their equivalents.

It will be understood by those skilled in the art that the present invention is not limited to the embodiments illustrated above and that many modifications and additions may be made without departing from the scope of the invention as defined in the appending claims.

The invention claimed is:

1. A method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, updating the estimate of the target hyperparameter vector by using the random sample, and selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

2. The method according to claim 1, further comprising the steps of:

selecting a further hyperparameter vector producing the best result in any previous iterations, and restricting the random sample of hyperparameter vectors by using the further selected hyperparameter vector.

3. The method according to claim 2, wherein the step of restricting the random sample of hyperparameter vectors involves using an interval surrounding the further selected hyperparameter vector.

4. The method according to claim 3, wherein the step of restricting the random sample of hyperparameter vectors is carried out prior to the step of selecting the hyperparameter vector producing the best result in the present and any previous iterations.

5. The method according to claim 3, wherein the step of restricting the random sample of hyperparameter vectors is carried out during the step of updating the estimate of the target hyperparameter vector.

6. The method according to claim 2, wherein the step of restricting the random sample of hyperparameter vectors is carried out prior to the step of selecting the hyperparameter vector producing the best result in the present and any previous iterations.

7. The method according to claim 2, wherein the step of restricting the random sample of hyperparameter vectors is carried out during the step of updating the estimate of the target hyperparameter vector.

8. The method according to claim 1, wherein the step of updating the estimate of the hyperparameter vector uses a weighting function.

9. The method according to claim 1, wherein E' is the selected hyperparameter vector X'_t at iteration t producing the best result $S(X'_t)$, the selected hyperparameter vector X'_t having elements X'_{tj} , wherein the step of updating the estimate of the target hyperparameter vector comprises the step of determining the hyperparameter v'_{jt} , where

8

$$v'_{jt} = \frac{\sum_{i=1}^n I(S(X'_i) \geq \gamma^t) W(X'_i; E') X'_{ij}}{\sum_{i=1}^n I(S(X'_i) \geq \gamma^t) W(X'_i; E')},$$

wherein γ^t is a threshold value and W is a weighting function.

10. The method according to claim 9, wherein the weighting function W is given by

$$W(X'_i; E') = 1 - \frac{\sqrt{\sum_{j=1}^m (X'_{ij} - E'_j)^2}}{\sqrt{\sum_{j=1}^m (X'_{ij})^2} \sqrt{\sum_{j=1}^m (E'_j)^2}}.$$

11. The method according to claim 1, wherein the steps are carried out by a programmed computer apparatus including a processor and a computer readable medium including computer executable instructions.

12. A classifier for use in a machine learning system, the classifier using hyperparameters as control parameters, wherein the hyperparameters are determined according to the set of steps recited in claim 1.

13. A non-transitory computer readable medium product including computer-executable instructions for carrying out a method of determining hyperparameters of a classifier in a machine learning system by iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, updating the estimate of the target hyperparameter vector by using the random sample, and selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

14. A device for determining hyperparameters of a classifier in a machine learning system, the device comprising a processor arranged for iteratively producing an estimate of a target hyperparameter vector, each iteration comprising the steps of:

drawing a random sample of hyperparameter vectors from a set of possible hyperparameter vectors, updating the estimate of the hyperparameter vector by using the random sample, and selecting, from the random sample of hyperparameter vectors, a hyperparameter vector producing a best result in the present and any previous iterations, and wherein the step of updating the estimate of the target hyperparameter vector uses said hyperparameter vector producing the best result.

15. The device according to claim 14, wherein the processor is further arranged for:

selecting a further hyperparameter vector producing the best result in any previous iterations, and restricting the random sample by using the hyperparameter vector producing the best result in any previous iteration.

16. A machine learning system, comprising the device for determining hyperparameters of a classifier defined in claim 14.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 8,438,120 B2
APPLICATION NO. : 12/597257
DATED : May 7, 2013
INVENTOR(S) : Stephan Alexander Raaijmakers

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page:

The first or sole Notice should read --

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 864 days.

Signed and Sealed this
Eighth Day of September, 2015



Michelle K. Lee
Director of the United States Patent and Trademark Office

EXHIBIT 2



US008144717B2

(12) **United States Patent**
Blange et al.

(10) **Patent No.:** **US 8,144,717 B2**
(45) **Date of Patent:** **Mar. 27, 2012**

(54) **INITIALIZATION OF A WIRELESS COMMUNICATION NETWORK**

(75) Inventors: **Marinus Johannes Blange**, Nootdorp (NL); **Miodrag Djurica**, Rotterdam (NL)

(73) Assignee: **Nederlandse Organisatie Voor Toegepast-natuurwetenschappelijk Onderzoek TNO**, Delft (NL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 784 days.

(21) Appl. No.: **12/159,753**

(22) PCT Filed: **Jan. 2, 2007**

(86) PCT No.: **PCT/NL2007/000001**

§ 371 (c)(1),
(2), (4) Date: **Dec. 2, 2008**

(87) PCT Pub. No.: **WO2007/078193**

PCT Pub. Date: **Jul. 12, 2007**

(65) **Prior Publication Data**

US 2009/0215482 A1 Aug. 27, 2009

(30) **Foreign Application Priority Data**

Dec. 30, 2005 (EP) 05078044

(51) **Int. Cl.**
H04L 12/28 (2006.01)

(52) **U.S. Cl.** **370/402**; 370/395.2; 370/400;
370/401; 455/41.2; 455/445

(58) **Field of Classification Search** 455/41.2,
455/435.1, 445; 370/338, 328, 329, 341,
370/395.2, 395.21, 400-402, 406

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,548,523 B2 * 6/2009 Babbar et al. 370/331
(Continued)

FOREIGN PATENT DOCUMENTS

EP 1 471 693 A2 10/2004

OTHER PUBLICATIONS

Forno et al., "Design and implementation of a Bluetooth ad hoc network for indoor positioning" IEE Proceedings: Software, IEE, vol. 152, No. 5, Oct. 7, 2005 pp. 223-228.

(Continued)

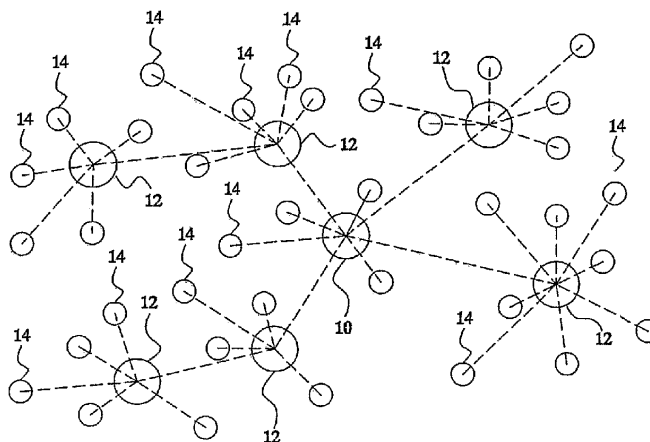
Primary Examiner — Dominic E Rego

(74) *Attorney, Agent, or Firm* — Leydig, Voit & Mayer Ltd.

(57) **ABSTRACT**

A wireless communication network comprises an association unit and a plurality of stations. At least part of the stations store association information indicative of an associated route for transmission of messages from the station to the association unit. The association information is generated by installing the stations in a not-associated state; transmitting association request messages from stations in the not-associated state; and transmitting association grant messages in response to the association request messages. Each association grant message is transmitted only from the association unit or, at least when the station that has transmitted association request message is outside a reception range of the association unit, from one of the stations that is in an associated state. Each station that has received the association grant message in response to its association request message is switched to the associated state. Association information is stored that indicates the station that has sent the association grant message as part of the associated route to the association unit.

20 Claims, 3 Drawing Sheets



US 8,144,717 B2

Page 2

U.S. PATENT DOCUMENTS

7,823,199	B1 *	10/2010	Rathi et al.	726/22
8,045,531	B2 *	10/2011	Cheng et al.	370/338
2001/0014868	A1 *	8/2001	Herz et al.	705/14
2003/0151513	A1	8/2003	Herrmann et al.	
2003/0174682	A1 *	9/2003	Barker	370/338
2004/0047307	A1 *	3/2004	Yoon et al.	370/324
2004/0223476	A1 *	11/2004	Jose et al.	370/338
2005/0088980	A1 *	4/2005	Olkkonen et al.	370/255
2005/0122927	A1	6/2005	Wentink	
2005/0195842	A1 *	9/2005	Dowling	370/401
2005/0221803	A1 *	10/2005	Hayashi	455/413
2006/0133266	A1 *	6/2006	Kim et al.	370/228

2006/0133328	A1 *	6/2006	Levendel et al.	370/338
2006/0166683	A1 *	7/2006	Sharma et al.	455/456.5
2006/0268756	A1 *	11/2006	Wang et al.	370/310
2007/0038346	A1 *	2/2007	Ehrlich et al.	701/33

OTHER PUBLICATIONS

Rubin et al., "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks" Aerospace Conference Proceedings, IEEE Mar 9-16, 2002. vol. 6, pp. 2727-2740.
 International Search Report for PCT/NL/2007/0000001, dated Jun. 6, 2007.

* cited by examiner

U.S. Patent

Mar. 27, 2012

Sheet 1 of 3

US 8,144,717 B2

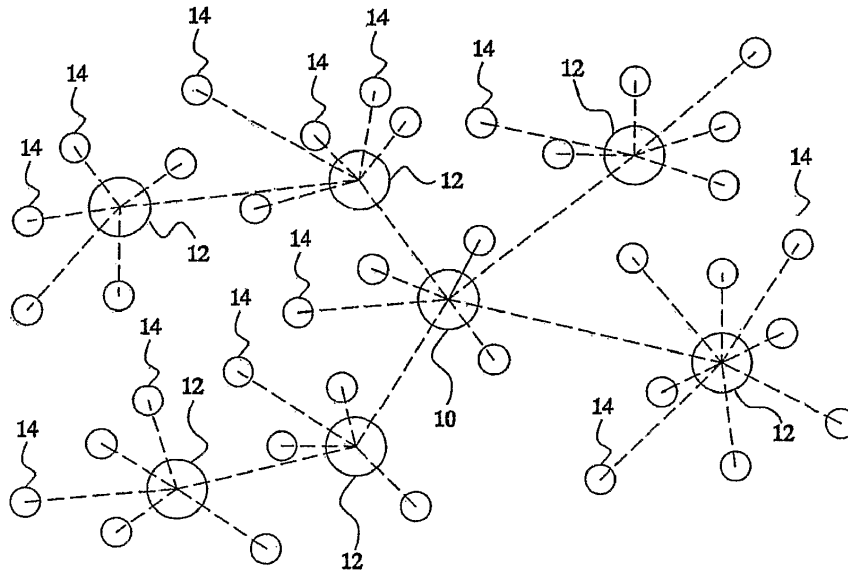


Fig.1

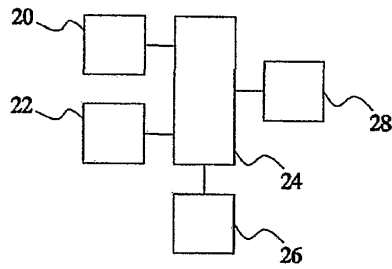


Fig.2

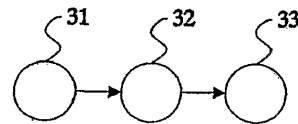


Fig.3

U.S. Patent

Mar. 27, 2012

Sheet 2 of 3

US 8,144,717 B2

Fig4

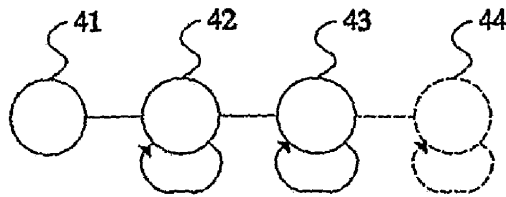


Fig.5a

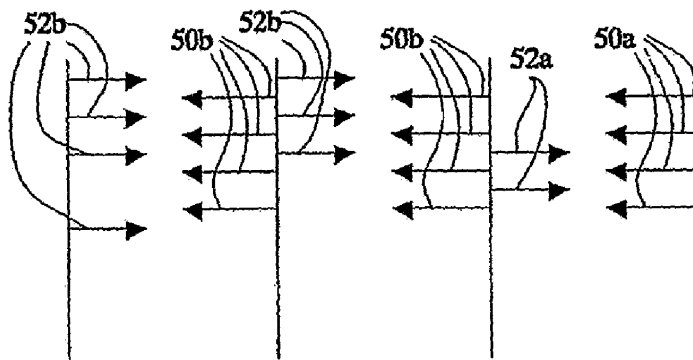
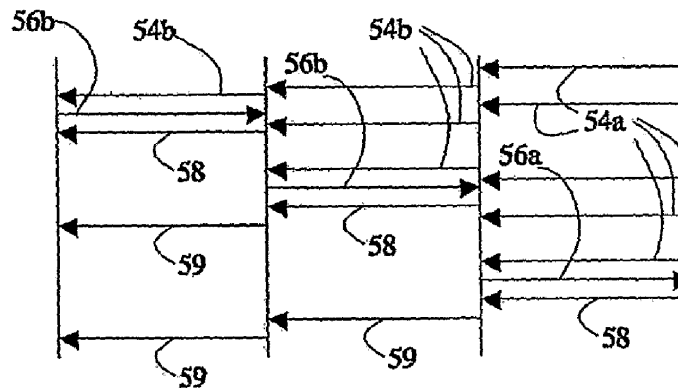


Fig.5b



U.S. Patent

Mar. 27, 2012

Sheet 3 of 3

US 8,144,717 B2

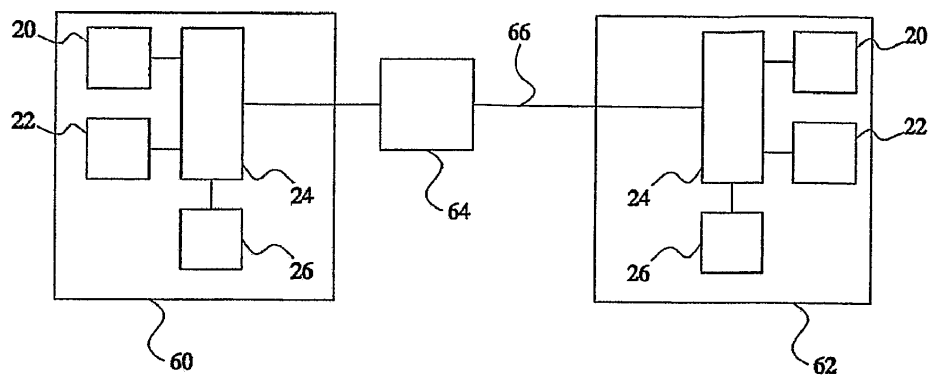


Fig.6

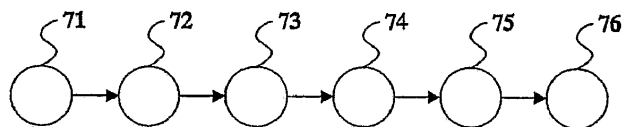


Fig.7

US 8,144,717 B2

1

**INITIALIZATION OF A WIRELESS
COMMUNICATION NETWORK****FIELD OF THE INVENTION**

The invention relates to a method of operating a wireless communication network, to a wireless communication system that uses such a method and to stations for use in such a network.

BACKGROUND

US patent application No. 2003/0151513 describes a wireless sensor station network. In operation data from the sensor stations is transmitted in multiple hops to a central panel and vice versa. This is because the sensor stations are not able to transmit directly to the central panel. Each sensor station is given its own, fixed route for transmitting and receiving data to and from the central panel. Typically, the data is first sent to another sensor station and from there to another sensor station and so on until a cluster head is reached. Then the data is sent through a network of cluster heads to the central panel.

After installation of the sensor stations at various locations in a sensing area a network topology is dynamically formed. That is, the routes for each of the sensor stations are automatically selected and programmed into the sensor stations. In US patent application No. 2003/0151513 describes a number of methods of dynamically forming the routes from the sensor stations to cluster head units that channel communication from groups of sensor stations. The initiative is taken by the cluster head unit, which transmits link discovery messages, which alert nearby sensor stations about the availability of the cluster head. These sensor stations forward the link discovery messages. Further sensor stations may receive and forward the forwarded link discovery messages and so on. The link discovery packets function to synchronize communication clocks of the cluster head and the sensor stations.

Next the routes for transmitting messages from the sensor stations to the association unit are defined. In one embodiment this is done by transmitting and forwarding route discovery packets from the association unit. The forwarded route discovery packets indicate the number of hops over which they have been forwarded. In this embodiment each sensor station picks a route by selecting the association unit or sensor station from which it received a route discovery packet over least hops. The sensor stations record an indication of these routes for later use in normal operation. Subsequently, the sensor stations send route registration packets to the association unit to indicate the route that they have selected.

In another embodiment the sensor stations send link registration messages without waiting for route discovery messages. These link registration messages may reach the association unit via each available route. In this embodiment the association unit selects routes for the sensor stations from which it has received link registration messages. After making the selection the association unit sends route definition messages to the sensor stations for recording an indication of these routes for later use in normal operation.

A problem with these techniques is that they require transmission of information via sensor stations before the routing topology has been defined, i.e. when plural routes are still possible. This complicates the (sensor) stations, because they have to support different modes of communication while operating in low-power mode. If forwarding of information is only done using the definite routing topology, a central approach is slow (each station has to wait until its upward station has centrally been assigned a route) and a decentral

2

approach is inefficient (a routing topology with an unnecessary amount of branches has been created)

Another problem that may occur is that the wireless links between nodes are hindered by barriers and that using the same frequency for upward and downward link routes requires synchronization. This complicates the (sensor) station, because they have to support synchronization and powerful RF signals.

A mobile communications protocol is known from an article titled "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks" by Izhak Rubin et al. and published in the IEEE Aerospace Conference proceedings 2002 (Piscataway N.J., Mar. 9-16, 2002) page 2727-2740. The mobile protocol of this article uses two types of network: a high power network termed the BNET, or backbone, and a plurality of low power networks called ANETs. In each low power network (ANET) low power nodes communicate under control of a central node of that ANET. Each of the central nodes also functions as backbone node, i.e. as a node in the high power backbone network.

Devices are provided that are capable of operating as backbone nodes. Initially, each of these devices decides whether it will switch to operation as backbone node, dependent on whether it has a higher qualifications to do so than neighboring devices that are capable of operating as backbone nodes. Subsequently, those devices that have switched to operation as backbone nodes receive join requests from remaining nodes that want to become part of a low power network (ANET) centred on a particular backbone node. The low power networks (ANETs) are formed by granting these join requests. Join requests can come from both remaining backbone capable devices that have not decided to operate as backbone devices, and from low power nodes that are not capable of operating as backbone devices at all.

The article also describes the possibility that remaining backbone capable devices (that have not initially decided to operate as backbone devices) can be commanded to become backbone nodes by existing backbone nodes. This happens when a remaining backbone capable device that is part of a low power network (ANET) receives and relays a join request from a requesting node that cannot be accepted by any existing backbone node. When this happens the central backbone node of the low power network (ANET) commands the remaining backbone capable device to switch to operation as a backbone node and to grant the join request of the requesting node. In this case the remaining backbone capable device joins the high power network (BNET), sets up a low power network (ANET) of its own and ceases to use the paths in the low power network (ANET) of the commanding backbone node. The remaining backbone capable device uses the high power network to relay further messages to or from the requesting node. No provision is made for routing through successive previously established low power network (ANET) nodes and association for this form of routing.

This system addresses the use of two types of network links: links in the low power networks and links in the high power networks. Direct communication links (not via other associated nodes) are used in the low power networks. The selection of communication paths in the high power network is not described, but as high power communication is used longer range paths can be used in this network than in the low power networks. As the system is intended to provide communication in a dynamic environment rather than control of devices in the system there is no single central backbone

US 8,144,717 B2

3

node. The use of two types of network without central unit makes this system complex and inefficient in bandwidth use.

SUMMARY OF THE INVENTION

Among others, it is desirable to provide for a wireless network wherein routes for communication can be determined without the need to forward information via stations for which the routes have not yet been defined.

Among others, it is desirable to provide for a method of operating a wireless network wherein routes for communication can be determined without the need to forward information via stations for which routes have not yet been defined.

Among others it is desirable to provide for stations for flexible use in wireless networks.

The invention provides for a wireless communication network according to claim 1. This network allows routes between stations and an association station to be set up for further use, using routes that have already been defined in response to earlier associations. Each route is established when an association request from a station is granted. Establishment of a route is effected by the association unit for example by storing information that defines the route to the associated station in a memory that the association unit uses to select the route each time when a message has to be sent to the associated station. Routes are established upon an association grant from the association unit or from a station that has switched to the associated state directly or indirectly based on an association grant from the association unit, in a wave of association that radiates from nodes to nodes starting from the association unit. Only the association unit and stations that have an established route to the association unit, which has been established in this way in response to association requests and corresponding association grants by nodes along the route, are allowed to grant association requests. Therefore their routes can immediately be used for the station that is associated.

In an embodiment a station is configured to transmit association requests first for association via an active frequency channel and subsequently, if the first transmitted association request is not granted, for association via an as yet unused frequency channel. The latter request may be transmitted for example directly to the association unit. Because the route that is to be used to the station is established only upon association it is prevented that inconsistent multiple routes are assigned when such different association requests are transmitted.

More generally, in an embodiment the station that issues the association grant to a specific station is configured to make issue of the association grant dependent on whether the specific station indicates that a function that is active in that specific station or not. This means that a specific station may associate with different other stations, dependent on whether the function is active in the specific station or not. As a result the specific station has the opportunity to issue a plurality of association requests: without activating the function and after the function is activated, for example with a request directly to the association unit. Thus, for example, if the function consumes a scarce resource (such as function that uses a frequency channel for communication between the stations, thereby excluding use of the frequency channel by other stations), it can be prevented that the resource is consumed needlessly by granting the association request before activating the function. Because the routes to stations are established only upon association it is prevented that inconsistent station addresses are assigned.

4

In an embodiment association is preceded by a discovery stage, wherein no routes are yet established but station discover other stations, that can function to relay messages. Thus, stations may be enabled first to discover addresses of available other stations to which association requests can be sent, or stations that may send association requests before a route is established. In a further embodiment discovery advertisement messages are used that identify stations that are enabled to grant discovery requests. Thus, it is made possible to avoid waste of energy or bandwidth on association requests that cannot be granted.

In another embodiment part of the stations are dual function stations, which start to send association grant messages only after receiving approval from the association unit to do so. Thus, the association unit has control over the selection of stations that act as relay station, even after the dual function station has obtained a route for its "own" messages (e.g. sensor messages from a built in sensor station). In an embodiment the dual function station comprises units that function as a beacon frame transmitter and as a station with sensor station-like communication abilities respectively. The association unit may follow a strategy of enabling such dual function stations selectively to avoid enabling many stations that act as relay for only a few stations each, or to avoid selecting stations when these will become a relay for messages for too many stations.

In another embodiment dual function stations are used that comprises a mechanically flexible connection between separate units that each have an antenna or antennas for transmission and reception. Thus such stations can be used to connect different spaces "around the corner".

In another embodiment dual function stations are used that operate concurrently in different frequency channels that are periodically active in time slots that repeat with different asynchronous periods. This simplifies coordination between different network parts.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and advantageous aspects will be described for exemplary examples, using the following figures.

FIG. 1 shows a wireless network

FIG. 2 shows a station

FIG. 3 shows a state diagram of a sensor station

FIG. 4 shows a state diagram of a beacon frame transmitter

FIG. 5a shows messages during a discovery phase

FIG. 5b shows messages during an association phase

FIG. 6 shows an embodiment of a beacon frame transmitter

FIG. 7 shows a state diagram of a beacon frame transmitter

DETAILED DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a wireless network. The network comprises an association unit 10, a plurality of beacon frame transmitters 12 and a plurality of sensor stations 14 (only some numbered explicitly). Fixed communication routes are defined between association unit 10 and sensor stations 14, using wireless communication links (shown by dashed lines) between association unit 10, beacon frame transmitters 12 and sensor stations 14. The communication routes form a tree structure with association unit 10 at its root and beacon frame transmitters 12 at its nodes. Each beacon frame transmitter 12 is coupled in a star topology to a number of the sensor stations 14.

FIG. 2 shows a station. At the level of detail shown, the figure applies to beacon frame transmitters 12 as well as

US 8,144,717 B2

5

sensor stations 14. The station contains a receiver circuit 20, a transmitter circuit 22, a processor circuit 24, a memory 26 and, at least in case of sensor stations, a sensing circuit 28. Processor circuit 24 is coupled to receiver circuit 20, transmitter circuit 22, memory 26 and optionally to sensing circuit 28. Each of receiver circuit 20 and transmitter circuit 22 include an antenna (not shown). Preferably, the antennas of receiver circuit 20 and transmitter circuit 22 are formed by a shared antenna of receiver circuit 20 and transmitter circuit 22.

Preferably the stations are battery operated, with on board batteries (not shown) or supplied by other on board power supplies such as solar cells. This means that it is desirable to reduce power consumption as much as possible.

In operation, data is exchanged between sensor stations 14 and association unit 10 via the links that are shown in FIG. 1. Because the links form a tree structure, a single route to and from association unit 10 is defined for each sensor station 12.

It should be appreciated that the communication routes are a result of selection from a greater number of possible routes. In principle, each sensor station 14 could communicate directly with association unit 10 and each beacon frame transmitter 12. In practice, a number of these direct communication routes are not possible, because of insufficient signal strength. But direct wireless communication from a sensor station 14 to still more than one beacon frame transmitters 12 and/or to association unit 10 may be possible. Similarly, each beacon frame transmitter 12 may be able to communicate directly with association unit 10 and/or a number of beacon frame transmitters 12.

The wireless connections shown in FIG. 1 are the result of a selection from these possible routes. A number of operable wireless communication links between sensor stations 14, beacon frame transmitters 12 and association unit 10 is not used. These links are not shown in FIG. 1. Sensor stations 14 transmit messages to association unit 10 via fixed routes and conversely association unit 10 transmits messages to sensor stations via the fixed routes. The routes are defined by information in memory 26.

Each sensor station 14 stores at least information that is selective for the beacon frame transmitter 12 at the start of its route to association unit 10; this address information is used to form messages that the sensor station 14 sends to association unit 10. Typically, each beacon frame transmitter 12 stores at least information that is selective for the next beacon frame transmitter 12 in the route or for association unit 10. This information is used to forward messages from the sensor station 14 sends to association unit 10. Alternatively, each sensor station 14 may store information that is selective for the entire route to association unit 10 and the sensor station 14 may use this information to form messages, from which beacon frame transmitters 12 can determine how to forward the message.

Conversely, association unit 10 preferably stores information for each sensor station 14 that is selective for the beacon frame transmitter 12 at the start of the route to that sensor station 14. Association unit 10 uses this information to form messages that association unit 10 transmits to the sensor station 14. Typically, association unit 10 stores information that identifies the entire route to each sensor station 14. The association unit 10 may use this information to form messages, from which beacon frame transmitters 12 can determine how to forward the message to the sensor station 14. Alternatively, each beacon frame transmitter 12 stores, for each sensor station 14 for which it can receive messages, at least information that is selective for the next beacon frame transmitter 12 in the route to that sensor station 14. This

6

information is used to forward messages for the sensor station 14 from association unit 10 to the sensor station 14.

An embodiment of the part of operation wherein the routes are established will now be described. Basically this embodiment includes two stages:

a discovery stage wherein beacon frame transmitters 12 and sensor stations 14 become "aware" of stations (association unit 10 or beacon frame transmitters 12 in the case of sensor stations and association unit 10, other beacon frame transmitters 12 and sensor stations 14 in the case of beacon frame transmitters 14) that are within reception range and

an association stage wherein sensor stations 14 and beacon frame transmitters 14 (or sensor stations that are fixedly coupled to these beacon frame transmitters) send requests to associate to association unit 10 or beacon frame transmitters 12, i.e. to establish a route through the association unit 10 or beacon frame transmitter 12 for subsequent use.

The association stage for a sensor station 14 (or beacon frame transmitter 12) is completed when an association request is granted by the association unit 10 or a beacon frame transmitter 12. The route via the granting station is thereby established for subsequent use. In one embodiment each beacon frame transmitter 12 is enabled to grant association requests immediately after it has received a grant of an association request itself, but preferably the beacon frame transmitter 12 is enabled to grant association requests only after receiving secondary grant message, this time from the association unit 14 that allows the beacon frame transmitter 12 to grant association requests. Association unit 10 may for example make issuance of the secondary grant message dependent on information about the number of as yet unassociated stations that the beacon frame transmitter 12 has discovered. Thus, association unit 10 can control network topology.

Preferably, the association requests are addressed specifically to beacon frame transmitters 12 or the association unit 10, using addresses (or more generally any identifier of these stations) that have been discovered in the discovery stage. Thus, it is excluded that conflicting association grants can be received in response to an association requests. However, addressing may be made unnecessary, for example when stations address all subsequent messages to one station that has sent a grant, or if some conflict resolution scheme is used.

In the embodiment the discovery stage comprises use of discovery messages from sensor stations 14 and beacon frame transmitters 12 (or sensor stations that are fixedly coupled to these beacon frame transmitters 12) and responses in reaction to these discovery messages. Alternatively or in addition, stations may be discovered from unsolicited messages. The discovery stage and the association stage may overlap, the stations continuing to collect identifiers of discovered stations between transmission of association requests during the association stage. Alternatively non-overlapping consecutive stages may be used. In another embodiment the discovery stage may even be eliminated, for example if association requests are used that are not specifically addressed, or if association requests are used to discover stations.

When the discovery stage and the association stage may overlap, it is preferred that beacon frame transmitters 12 include information in their messages to indicate whether they have been enabled to grant association requests (preferably in general, i.e. not limited to requests from specific sensor stations 14, but in a further embodiment specific sensor stations 14 or beacon frame transmitters 12 from which association requests may be accepted may be indicated).

US 8,144,717 B2

7

Thus, sensor stations 14 and/or beacon frame transmitters 12 can avoid transmission of association requests to discovered beacon frame transmitters 12 that are not enabled to grant association requests.

FIG. 3 shows a state diagram of operation of a sensor station 14 before a route is established. FIGS. 5a,b shows messages exchanged with sensor station 14 during discovery and association respectively. Initially the sensor station 14 is in a first state 31. In this first state 31 processor circuit 24 causes transmitter 22 to transmit discovery messages 50a, for example once every minute, every two minutes or every hour. When processor circuit 24 detects that receiver 20 has received a beacon frame 52a from a beacon frame transmitter 12 or association unit 10, processor circuit 24 switches sensor station 14 to a second state 32. In this way, the beacon frame 52a acts as a discovery confirm message.

Preferably, sensor station 14 switches to second state 32 in response indiscriminately to both beacon frames 52a that are transmitted in response to the discovery message or to any other beacon frame that is received spontaneously. In an embodiment sensor station 14 switches to second state 32 in response only to beacon frames 52a wherein it is indicated that the transmitting station is prepared to grant association requests and not to optional other beacon frames 52a that do not contain such an indication.

In second state 32 processor circuit 24 causes transmitter 22 to transmit association request messages 54a for example once every minute, every two minutes or every hour. Preferably each association request message 54a includes an identification of the sensor station 14 that allows messages specifically directed at the sensor station 14 to be sent in response. Also preferably, processor 24 reads an identification of the beacon frame transmitter 12 or association unit 10 from a previously received discovery confirm message and includes that identification in the association request message. This may be done for example when it is desirable to avoid collisions between association grant messages in response to the association request message.

When processor circuit 24 detects that receiver 20 has received an association grant message 56a from a beacon frame transmitter 12 or association unit 10 directed at the sensor station 14 of the processor circuit 24 in response to an association request message 54a, processor circuit 24 switches sensor station 14 to a third state 33 and records information for reaching this beacon frame transmitter 12 or the association unit 10 in memory 26 to define route to association unit 10 for use during normal operation. Optionally sensor station sends an acknowledgment 58 of the association grant message back to the beacon frame transmitter 12 or association unit 10.

In third state 33 sensor station 14 stops transmitting association request messages 54a. In one embodiment sensor station 14 stops transmitting discovery messages 50a already when it switches to second state 32. Alternatively sensor station 14 may be arranged to continue transmitting discovery messages 50a in second state 32, only to stop after switching to third state 33 or after a predetermined time interval has elapsed after switching to second state 32, if this occurs earlier than the switch to third state 33.

In an embodiment a station is configured to transmit association requests first for association via an active frequency channel and subsequently, if the first transmitted association request is not granted, for association via an as yet unused frequency channel. The latter request may be transmitted for example directly to the association unit. Because the route that is to be used to the station is established only upon

8

association it is prevented that inconsistent multiple routes are assigned when such different association requests are transmitted.

More generally, in an embodiment the station that issues the association grant to a specific station is configured to make issue of the association grant dependent on whether the specific station indicates that a function that is active in that specific station or not. This means that a specific station may associate with different other stations, dependent on whether the function is active in the specific station or not. As a result the specific station has the opportunity to issue a plurality of association requests: without activating the function and after the function is activated, for example with a request directly to the association unit. Thus, for example, if the function consumes a scarce resource (such as function that uses a frequency channel for communication between the stations, thereby excluding use of the frequency channel by other stations), it can be prevented that the resource is consumed needlessly by granting the association request before activating the function. Because the routes to stations are established only upon association it is prevented that inconsistent station addresses are assigned.

FIG. 4 shows a state diagram of operation of a beacon frame transmitter 12 before a route is established. Initially the beacon frame transmitter 12 is in a first state 41. In this first state 41 processor circuit 24 causes transmitter 22 to transmit discovery messages 50b, for example once every minute, every two minutes or every hour. When processor circuit 24 detects that receiver 20 has received a beacon frame 52b from another beacon frame transmitter 12 or association unit 10, processor circuit 24 switches beacon frame transmitter 12 to a second state 42. In this way, the beacon frame 52b acts as a discovery confirm message. In an embodiment discovery confirm messages may also be transmitted unsolicited, i.e. not in response to a discovery message, in this sense discovery messages could also be called "hello" messages. The term "discovery advertisement message" is used herein both for messages in response to discovery requests and unsolicited (hello) messages. Examples of use of discovery advertisement messages will be given in terms of discovery confirm messages, but unless indicated otherwise these examples apply generally for discovery advertisement messages. In a further embodiment stations need not transmit discovery messages 50a,b at all. In this case the stations listen only to unsolicited messages. The discovery confirm message does not need to be addressed to any station in particular: any station may react to any discovery confirm message.

Preferably, beacon frame transmitter 12 switches to second state 42 in response indiscriminately to both a beacon frame 52b that is transmitted in response to the discovery message or to any other beacon frame that is received spontaneously. In an embodiment sensor station 14 switches to second state 42 in response only to beacon frames 52b wherein it is indicated that the transmitting station is prepared to grant association requests and not to optional other beacon frames 52b that do not contain such an indication.

In second state 42 processor circuit 24 causes transmitter 22 to transmit association request messages 54b, for example once every minute, every two minutes or every hour. Preferably each association request message 54b includes an identification of the beacon frame transmitter 12 that allows messages specifically directed at the beacon frame transmitter to be sent in response. Also preferably, processor 24 reads an identification of the beacon frame transmitter 12 or association unit 10 from a previously received discovery confirm message and includes that identification in the association request message. This may be done for example when it is

US 8,144,717 B2

9

desirable to avoid collisions between association grant messages in response to the association request message.

When processor circuit 24 detects that receiver 20 has received an association grant message 56b from another beacon frame transmitter 12 or association unit 10 in response to an association request message 54b, processor circuit 24 switches beacon frame transmitter 12 to a third state 43 and records information for reaching this other beacon frame transmitter 12 or the association unit 10 in memory 26 to define a route to association unit 10 for use during normal operation. Optionally frame transmitter 12 sends an acknowledgment 59 of the association grant message back to the beacon frame transmitter 12 or association unit 10.

In third state 43 frame transmitter 12 stops transmitting association request messages 54b. In one embodiment frame transmitter 12 stops transmitting discovery messages 50b when it switches to second state 42. Alternatively frame transmitter 12 may stop after switching to third state 43, or after a predetermined time interval has elapsed after switching to second state 42, if this occurs earlier than the switch to third state 42.

Preferably beacon frame transmitter 12 transmits discovery confirm messages in at least when in second and third state 42, 43. More preferably beacon frame transmitter 12 includes an indication in the discovery confirm messages indicating whether the beacon frame transmitter 12 is ready to respond to association grant messages with association requests. This allows stations to avoid using bandwidth for transmission of useless association requests. In principle beacon frame transmitter 12 may also be arranged to transmit discovery confirm messages in the first state 41. Preferably, when beacon frame transmitter 12 receives a discovery request message from a sensor station 14 or another beacon frame transmitter 12 beacon frame transmitter 12 records an identification of the sensor station 14 or other beacon frame transmitter 12 from which the discovery request was received. Optionally beacon frame transmitter 12 also records identifications of stations that have transmitted association requests that the beacon frame transmitter 12 did not respond to, if any.

In first state 41 and second state 42 beacon frame transmitter 12 does not respond to association request messages 54a. However, in third state 43 beacon frame transmitter 12 does respond to association request messages 54a. In this state processor circuit 24 causes transmitter 22 to transmit an association grant message 56a in response to each received association request message 54a, directed at the station that has sent association request message 54a, using the station identification in the association request message 54a. Processor circuit 24 includes information for reaching the beacon frame transmitter 12 in the association grant message 56a and optionally also further information for reaching association unit 10.

Furthermore, beacon frame transmitter 12 transmits an update message 59 with information that the sensor station 14 has been associated to association unit 10 after sending the association grant message 56a (and after receiving an optional acknowledgment 58 of that message).

This response of beacon frame transmitter 12 to association request messages 54a is not limited to such messages from sensor stations 14, but applies also to association request messages 54b from other beacon frame transmitters 12.

It should be noted that beacon frame transmitter 12 and sensor station 14 become stations of the network with well defined routes only when reaching their respective third states 33, 43. Furthermore it should be noted that beacon frame transmitters 12 respond to association request messages

10

54a,b only once they have entered their third state 43, that is, only when the beacon frame transmitter 12 has a well defined route to association unit 10. Thus, the information about association can be transmitted along the route that will be used during normal operation. This route may also be used as part of the route to the newly associated sensor station 14. Association unit 10 may use information from its memory, about the route to the beacon frame transmitter 12 to define the route to the sensor station 14. In another embodiment, the beacon frame transmitters 12 along the route may add their identity to inform association unit 10 of the route, or in yet another embodiment the beacon frame transmitter 12 that associates to the sensor station 14 may insert this information.

It may be noted that not all beacon frame transmitters 12 need to become part of the network as beacon frame transmitters. If a beacon frame transmitter 12 receives no association grant signal 56b or grants no association grant requests from other beacon frame transmitters 12 or sensor stations 14 the beacon frame transmitter 12 does not become a functional beacon frame transmitter in the network.

In an embodiment association unit 10 decides on whether a beacon frame transmitter 12 becomes a functional beacon frame transmitter. In this embodiment functional beacon frame transmitters 12 in third state 43 forward association requests 54a from other beacon frame transmitters 12 (and optionally also from sensor stations 14) to association unit 10 before sending an association grant message 56b. (Alternatively a further message derived from this association request is forwarded).

In this embodiment association unit 10 is arranged to grant or deny the forwarded association request in response to forwarding (or to the further message). When the association request is granted association unit sends back a grant message, which is used to generate the association grant signal 56b to the original association request 54b. Thus association unit 10 (and/or a beacon frame transmitter 12) is enabled to control generation of the network topology, for example by first granting association to selected preferred beacon frame transmitters and granting association requests to others later only if this is still necessary to complete connections to all sensor stations 14.

In a further embodiment beacon frame transmitters 12 are arranged to supply information about the number and/or identity of the sensor stations 14 and/or beacon frame transmitters 12 from which it has received beacon frame request messages and/or association request messages. The identity may be provided by including identity information in beacon frame request messages and/or association request messages. The beacon frame transmitters 12 may supply this information for example in their own beacon frame request messages and/or association request messages. In this further embodiment the association unit 10 (and/or a beacon frame transmitter) is provided with information which it may use to select beacon frame transmitters 12 for granting association. This may be used for example to distribute links over several beacon frame transmitters 12, or on the contrary to avoid distribution by concentrating links on one beacon frame transmitter 12.

In a further embodiment beacon frame transmitters 12 return grants of in response to association requests only after receiving a grant of a secondary association request, distinguished, at least in function, if not in format, from the previously described association request. For this purpose optionally an additional fourth state 44 may be added to the states of FIG. 4. When in third state 43 (after receiving the previously described grant to one of the previously described association requests, which will be termed primary association requests beacon frame transmitter 12 transmits a secondary associa-

US 8,144,717 B2

11

tion request to association unit 10. Upon receiving a grant of this secondary association request beacon frame transmitter 12 switches to the fourth state 44. In this embodiment the beacon frame transmitter 12 is enabled to begin to return association grant messages to requesting stations only in fourth state 44. Optionally the beacon frame transmitter indicates this in discovery confirm messages.

Preferably beacon frame transmitter 12 is configured to include a list of stations from which beacon frame transmitter 12 has received discovery request messages (and/or not-granted association request) in the secondary association request. As an alternative beacon frame transmitter 12 may optionally be configured to transmit one or more accompanying messages to provide such a list to association unit 10. This enables association unit 10 to select to which of the beacon frame transmitters 12 it will grant association requests of the second type. In this way association unit 10 can control the network topology, after admitting beacon frame transmitters 12 to the network by granting a primary association request.

In a preferred embodiment each functional beacon frame transmitter 12 (and association unit 10, which acts as a beacon frame transmitter in this respect) operates in a time slot scheme during normal operation, periodically transmitting a beacon signal to which the sensor stations 14 that are associated with that beacon frame transmitter may synchronize. After the beacon signal the beacon frame transmitter 12 and its associated stations exchange messages for a limited period.

This allows the beacon frame transmitters 12 and the sensor stations 14 to switch between a low power sleep mode and a normal operating mode. After exchanging messages the beacon frame transmitters 12 and the sensor stations 14 switch to the sleep mode wherein power supply and/or clock signals to at least part of the beacon frame transmitter 12 and the sensor station 14 is switched off. After counting off a predetermined time-interval the beacon frame transmitters 12 and the sensor stations 14 switch back to the normal operating mode. With a predetermined delay after switching to the normal operating mode the beacon frame transmitter 12 transmits the beacon frame and the sensor stations 14 receive this beacon frame.

In this embodiment the association unit 10 and/or the beacon frame transmitters preferably select the timing of the first of the periodic beacon frames so that the time slots of different beacon frame transmitters 12 that are within range to receive each others messages do not overlap. In this embodiment the information in the sensor stations 14 after association defines the time slot in which the sensor station 14 should switch to the normal operating mode.

In one embodiment each beacon frame transmitter 12 also switches to a forwarding mode in the time slot defined by the further beacon frame transmitter 12 through which the original beacon frame transmitter 12 transmits messages to the association unit 10 (or in the time slot of the association unit 10). When in the forwarding mode the original beacon frame transmitter 12 exchanges messages with the further beacon frame transmitter 12.

This embodiment has the disadvantage that overlap of the time slots of neighboring beacon frame transmitters 12 must be avoided. This means that beacon frame transmitters 12 must monitor time slots of other beacon frame transmitter 12. In turn this means that the beacon frame transmitters 12 cannot operate in the sleep mode as much as desirable.

In an alternative embodiment the beacon frame transmitter 12 are designed to transmit and receive in a selectable one of a plurality of frequency channels. In this embodiment the

12

frequency channels for different functional beacon frame transmitters 12 are selected during initialization, for example by the beacon frame transmitters themselves after detecting that no messages are received in the frequency channel during a predetermined time period, or by the association unit 10 so that beacon frame transmitters 12 that have been detected to receive each other get different frequency channels to avoid collisions. In an embodiment the beacon frame transmitters 12 and their associated sensor stations 14 switch to the selected frequency channel when they start normal operation. In this embodiment the information in the sensor stations 14 after association defines the frequency channel for communicating with the beacon frame transmitter 12. Use of different frequency channels has the advantage that no extensive synchronization between time slots of beacon frame transmitters 12 is needed. Therefore beacon frame transmitters 12 can switch to the sleep mode more often.

In this further embodiment one frequency channel may be used during initialization. Alternatively a plurality of the frequency channels may be used in parallel. As an alternative the beacon frame transmitters 12 themselves may select a frequency channel during initialization after detecting that no messages are received in the frequency channel during a predetermined time period. Subsequently each beacon frame transmitter 12 performs all of its operations towards sensor stations 14 using the selected frequency channel, and uses another frequency channel to make upstream connections toward association unit 10.

FIG. 6 shows a further embodiment of a beacon frame transmitter. The beacon frame transmitter contains two unit 60, 62 as shown in FIG. 2, plus a message exchange memory 64 coupled between the processor circuits 24 of the two units. In one embodiment mutually different units 60, 62 are used, a first unit 60 arranged to perform the functions of a beacon frame transmitter (having a higher power transmitter 22 and/or a differently programmed processor circuit for example than the other unit) and a second unit 62 of a type used in sensor stations 14. A mechanically flexible connection 66 is provided between the units 60, 62 (e.g. between message exchange memory 64, which may be part of the first unit 60 and second unit 62). Flexible connection 66 is for example at least 5 centimetre long.

In this embodiment a first unit 60 acts as beacon frame transmitter, except for the transmission of discovery requests 50b, association requests 54b and any communication towards association unit (e.g. update message 59). The second unit 62 performs the remaining functions of the beacon frame transmitter. It may be noted that this second unit performs much like a sensor station 14, except that it does not generate or receive its own messages. Instead it forwards messages to or from the other unit. Messages to and from the association unit 10 or other beacon frame transmitters 12 (or data from such messages) are placed in message exchange memory 64.

In normal operation units 60, 62 switch between the sleep mode and normal active mode in time slots defined by respective beacon frames. The unit that acts as beacon frame transmitter transmits its own beacon signal, preferably in the select frequency channel for the beacon frame transmitter. The unit that acts as sensor station switches to the normal operation mode in the time slot defined by the beacon frames of the further beacon frame transmitter to which the beacon frame transmitter 60, 62, 64 is associated. Thus, no power expensive synchronization is needed.

FIG. 7 shows a state diagram of operation of the paired units of FIG. 6 during discovery and association. The paired units 60, 62 start in a first state 71 wherein a second unit 62 of

US 8,144,717 B2

13

the pair 60, 62 that acts like a sensor station transmits discovery messages. When the second unit 62 detects that a beacon frame from a beacon frame transmitter 12 or association unit 10 has been received, the paired units switch to second state 72. In second state 72 the first unit 60 of the pair that acts as a beacon frame transmitter starts transmitting beacon frames in response to received discovery messages by transmitting a discovery confirm message. In these beacon frames the first unit 60 includes information to indicate that the first unit 60 is not yet associated.

Also in second state 72 the second unit 62 starts monitoring whether it receives beacon frames with information that indicates that the station that transmits the beacon frame has been associated (or is association unit).

When the second unit 62 detects a beacon frame with information that indicates that the unit that transmits the beacon frame has been associated, the paired units 60, 62 switch from second state 72 to a third state 73. Herein the second unit 62 starts transmitting association requests. When the second unit 62 receives an association confirmation in response, the paired units 60, 62 switch to a fourth state 74. In second, third and fourth state 72, 73 and 74 first unit 60 starts and restarts counting a time out interval. The time out interval is restarted each time when first unit 60 receives a discovery message from a new source (sensor station or other beacon frame transmitter). The time out interval expires after a predetermined duration (e.g. a minute or more) wherein first unit 60 has received no discovery message from any new source. After expiration of the time out interval unit 60 stops transmitting beacon frames in response to received discovery messages.

From fourth state 74, after expiry of the time out interval the paired units 60, 62 switch to a fifth state 75 wherein second unit 60 transmits a beacon frame transmitter association request for the first unit 62, listing the sources from which the first unit has received a discovery message. This beacon frame transmitter association request is forwarded to association unit 10, which grants or denies the request, dependent on the list (granting the request for example if more than a threshold number of sources are listed that have not yet been associated). When the beacon frame transmitter association request is denied, new beacon frame transmitter association requests may be sent. Association unit 10 may be configured to lower the threshold with time so that the same request may be granted later if a source that is mentioned in the request cannot be reached in any other way even after an extended period of time.

Association unit 10 sends a grant message with a grant signal to the first unit 60 via the second unit 62. When the grant message is received the paired units 60, 62 switch to a sixth state 76. Wherein first unit 62 starts transmitting beacon frames indicating that it has been associated, after which it can cause other units to switch from the first state 72 to the third state 73.

Although completely separate units 60, 62 are shown, it should be appreciated that part or all of these units may use shared circuits. Use of separate circuits has the advantage that a standard sensor station unit may be used. The use of a flexible connection has the advantage that the different units may be located "around a corner" from each other, so that one unit has contact with a space to which the other has no contact. Thus, a system in a building can be realized very easily with this type of beacon frame transmitter.

Although the network has been described with beacon frame transmitters 12, it should be appreciated that instead of beacon signal synchronized transmission other forms of message exchange may be used. In this case beacon frame trans-

14

mitters 12 act simply as stations. In that case beacon frame requests are only discovery requests and beacon frame transmissions responses to discovery requests.

Furthermore, although an example has been discussed wherein the beacon frame transmitters 12 merely act to relay sensor data, it should be appreciated that the beacon frame transmitters 12 themselves may also be equipped with sensors, so that the beacon frame transmitters 12 can also transmit their own sensor data.

Although a network of sensor stations has been described (each provided with a temperature sensor, and/or a smoke sensor and/or a light intensity sensor and/or a moisture sensor and/or a gas sensor etc.) has been described it should be appreciated that the described techniques can also be applied to non-sensor or part-sensor networks wherein sensor stations 14 are simply information handling stations.

Although an association process with a discovery phase (FIG. 5a) and an association phase has been described, it should be appreciated that the two phases can be integrated partly or wholly. In one embodiment beacon frame requests 50a double as association requests 54a. In another embodiment association grant messages 56a double as beacon transmissions 52a. Both embodiments may be combined. However, the use of separate phases has the advantage that the whole process runs more quickly.

The invention claimed is:

1. A wireless communication network that comprises an association unit and a plurality of stations, each station being configured to start up in a not-associated state and to transmit association request messages in the not-associated state, each station being configured to switch to an associated state upon reception of an association grant in response to one of its association request messages, the association request message and/or the association grant in response thereto establishing an operating route associated with the station for exchanging operating messages with the association unit during subsequent operation, the operating route associated with the station running through a source of the association grant in response to said one of the association request messages and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations, in which network

the association unit is configured to transmit association grants as responses to received ones of the association request messages;

at least part of the stations is configured to become active to transmit association grants as responses to received ones of the association request messages, but only after switching to the associated state upon an association grant from the association unit or one of the at least part of the stations switched to the associated state directly or indirectly based on an association grant from the association unit.

2. A wireless communication network according to claim 1, wherein said part of the stations comprises at least one dual function station, said dual function station being configured to transmit association grants only after receiving a secondary association grant, the dual function station being configured to transmit one or more secondary association request messages to the association unit along its operating route when in the associated state, only the association unit being configured to originate the secondary association grant in response to the secondary association request message.

US 8,144,717 B2

15

3. A wireless communication network according to claim 2, wherein the dual function station is configured to transmit information to the association unit indicating from which of the stations said dual function station has received transmissions, the association unit being configured to select whether to grant the secondary association request dependent on said information.

4. A wireless communication network according to claim 3, wherein the stations are configured to transmit discovery messages before switching to the associated state, said transmissions comprising at least the discovery messages.

5. A wireless communication network according to claim 2, wherein the dual function station comprises a first and second unit, each with at least a respective antenna, coupled via a mechanically flexible connection, the first unit being configured to transmit first ones of the association request messages and to receive first ones of the association grants, the second unit being configured to receive second ones of the association request messages and to transmit second ones of the association grants in response to the first ones of the association request messages.

6. A wireless communication network according to claim 2, wherein the dual function station is configured to operate using respective, mutually distinct first and second frequency channels, using the first frequency channel for transmitting association request messages and receiving association grants, and using the second frequency channel for receiving association request messages and transmitting association grants.

7. A wireless communication network according to claim 1, wherein each station from said at least part of the stations is configured to transmit discovery advertisement messages including an identification of the station from said at least part of the stations, the stations that transmit association request messages being configured to address the association request messages using identifications from the discovery advertisement messages.

8. A wireless communication network according to claim 7, wherein each station in said at least part of the stations is configured to indicate, in the discovery advertisement messages, whether the station is enabled to transmit association grants, the stations that transmit association requests being configured to transmit association requests addressed to stations from said at least part of the stations only if the addressed station has indicated in at least one of the discovery advertisement messages that it is enabled to transmit association grants.

9. A wireless communication network according to claim 1, wherein said at least part of the stations are configured to transmit route information toward the association unit after transmitting an association grant, the route information defining the associated route for the station to which the association grant has been sent.

10. A wireless communication network according to claim 9, wherein said at least part of the stations is configured to forward at least part of the route information towards the association unit, to store at least a further part of the route information and to route subsequent messages from the association unit to the station for which the route information was sent according to said stored at least further part.

11. A station for use in a wireless communication network, the station comprising a transmitter arrangement, a receiver arrangement and a processor circuit coupled to the transmitter arrangement and the receiver arrangement, the processor circuit being configured to start up in a not-associated state and to cause the transmitter arrangement to transmit association request messages in the not-associated state, the processor

16

circuit being configured to switch to an associated state upon reception at the receiver arrangement of an association grant in response to one of the association request messages, the processor circuit being configured to establish an associated route for communication with an association unit during subsequent communication of operating messages to an association unit, the associated route including a source of the association grant, and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations, the processor circuit being configured to cause the transmitter arrangement to initiate and transmit association grants in response to association request messages, but only after switching to the associated state.

12. A station according to claim 11, wherein the processor circuit is configured, when switched to the associated state, to cause the transmitter arrangement to transmit a secondary association request message to the association unit, and to cause the transmitter arrangement to transmit the association grants only after receiving a secondary association grant in response to said secondary association request message.

13. A station according to claim 11, wherein the processor circuit is configured to send information to the association unit indicating from which one of the stations said dual function station has received messages prior to receiving the secondary association grant.

14. A method of operating a wireless communication network that comprises an association unit and a plurality of stations, the method comprising

installing the stations in a not-associated state;

transmitting association request messages from stations in the not-associated state;

transmitting the association grants in response to the association request messages,

switching each station to an associated state when the station has received an association grant in response to one of its association request messages;

storing route information defining a route between the station that has switched to the associated state and the association unit, the route including the station that has sent the associated grant and, if the source of the association grant is not the association unit, the operating route between the source of the association grant and the association unit previously established for the source of the association grant in response to association request messages and/or the association grants for intermediate stations; wherein

each association grant is transmitted only from the association unit or from an intermediate station, the latter only after said intermediate station has switched to the associated state upon an association grant from the association unit or one of the at least part of the stations switched to the associated state directly or indirectly based on an association grant from the association unit.

15. A method according to claim 14, wherein the intermediate station is a dual function station, the method comprising transmitting a secondary association request message from the dual function station when the dual function is in the associated state;

transmitting a secondary association grant originating from the association unit in response to the secondary association request message;

enabling the dual function station to transmit the association grants only upon receiving the secondary association grant.

US 8,144,717 B2

17

16. A method according to claim 15, comprising transmitting information from the intermediate station to the association unit about from which one of the stations the intermediated station has received transmissions; selecting, in the association unit, whether to grant the secondary association request dependent on said information.

17. A method according to claim 16, further comprising transmitting discovery messages from the stations before switching to the associated state, said transmissions comprising the discovery messages.

18. A method according to claim 15, further comprising the steps of:

transmitting discovery advertisement messages from at least part of the stations, including an identification of the station that has transmitted the discovery advertisement message; addressing the association request messages using identifications from the discovery advertisement messages.

18

19. A method according to claim 18, further comprising the steps of:

including an indication in the discovery advertisement messages, whether the station that transmits the discovery advertisement message is enabled to transmit association grants,

transmitting association requests addressed to stations only if the addressed station has indicated that it is enabled to transmit association grants.

20. A method according to claim 15, further comprising switching the intermediate station to operate at a first and second, mutually different transmission frequency after switching to the associated state, for directly or indirectly communicating with the association unit and for communicating with its associated stations respectively.

* * * * *