



(19) **United States**

(12) **Patent Application Publication**  
**Bush**

(10) **Pub. No.: US 2004/0156388 A1**

(43) **Pub. Date: Aug. 12, 2004**

(54) **SYSTEM FOR MAINTAINING QUALITY OF SERVICE**

**Related U.S. Application Data**

(75) Inventor: **Stephen F. Bush**, Latham, NY (US)

(60) Provisional application No. 60/445,579, filed on Feb. 7, 2003.

Correspondence Address:  
**TAROLLI, SUNDHEIM, COVELL & TUMMINO LLP**  
Suite 1111  
526 Superior Avenue  
CLEVELAND, OH 44114 (US)

**Publication Classification**

(51) Int. Cl.<sup>7</sup> ..... **H04J 3/16**

(52) U.S. Cl. .... **370/465; 370/400**

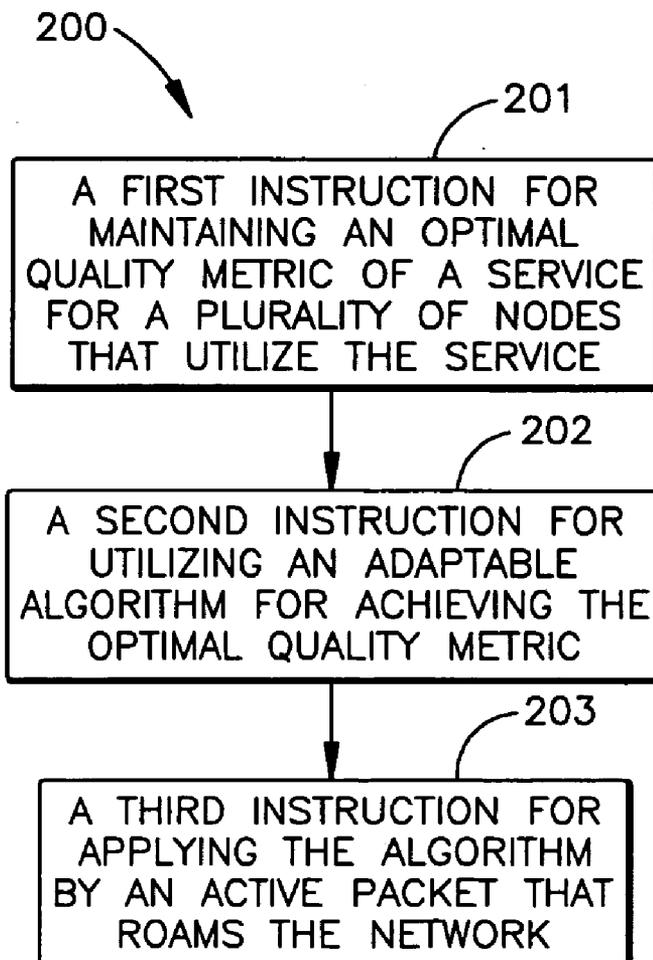
(57) **ABSTRACT**

A system in accordance with the present invention operates a wireless ad hoc network. The system includes a plurality of nodes and an active packet. The active packet maintains an optimal quality metric of a service for nodes of the plurality of nodes that utilize the service. The active packet utilizes an adaptable algorithm for achieving the optimal quality metric.

(73) Assignee: **Lockheed Martin Corporation**, Owego, NY (US)

(21) Appl. No.: **10/771,027**

(22) Filed: **Feb. 3, 2004**



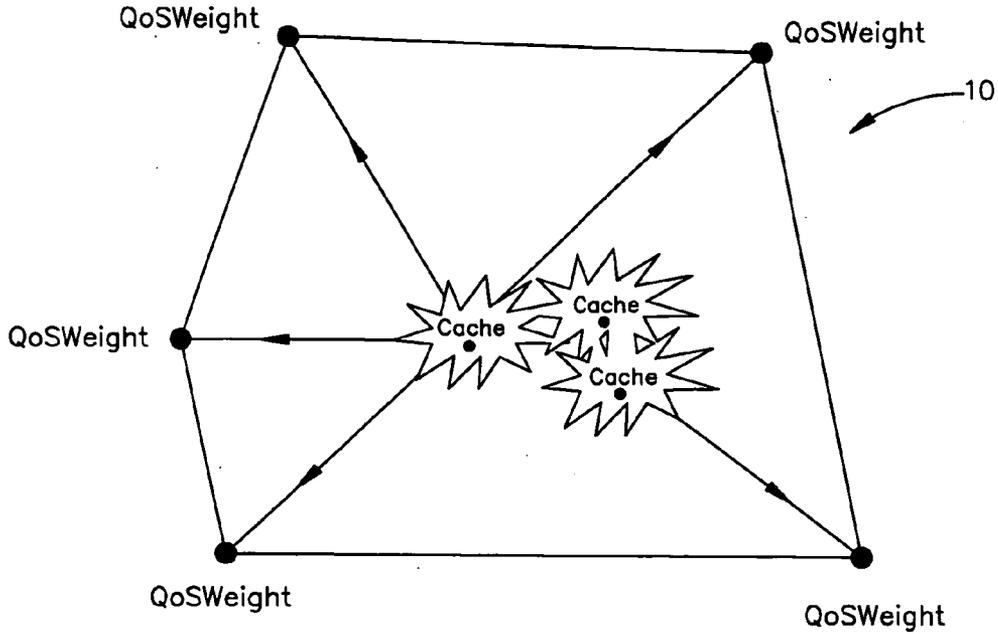


Fig.1

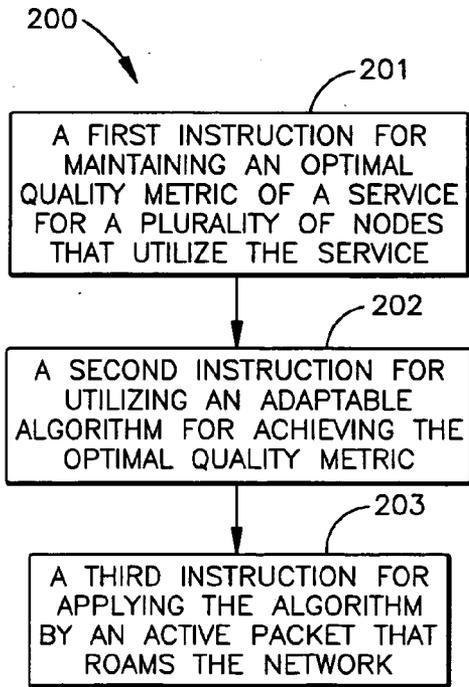


Fig.2

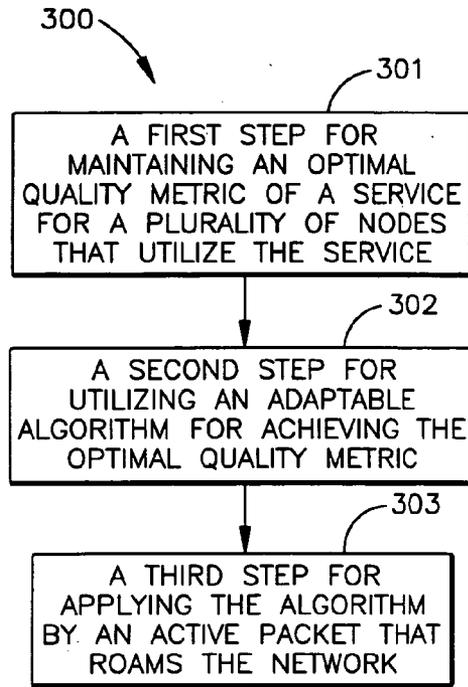


Fig.3

## SYSTEM FOR MAINTAINING QUALITY OF SERVICE

### RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application No. 60/445,579, filed Feb. 7, 2003.

### FIELD OF INVENTION

[0002] The present invention relates to a system for maintaining quality of service within a wireless network, and more specifically, to a system for maintaining a quality of service center of mass within wireless network operation.

### BACKGROUND OF THE INVENTION

[0003] Ad-Hoc Networks are a subset of variable topology networks. The goal of variable topology networks is to maintain message delivery as the network topology varies. Network nodes ideally should be able to dynamically form transient networks. Nodes, which may be located on rapidly moving platforms such as aircraft, ideally should be able to join, leave, and re-join networks which may form at any time. Networks spontaneously form and their topologies may change rapidly. An additional challenge required by airborne and heterogeneous air and ground environments is the ability to provide predictable and optimized quality of service (QoS) for reliable information transmission over variable topologies.

[0004] In order to provide predictable and optimal QoS in ad-hoc networks, a network architecture may support adaptation to a rapidly changing infrastructure. The degree to which the network adapts may be dependent on the rate of change of the topology. QoS requirements are conventionally stated in the form of an optimization problem with a cost function that is optimized by adaptation within the network.

[0005] An applicable option from complexity theory, the No Free Lunch Theorem defines a limit on the ability of any single algorithm, or protocol, to meet all QoS requirements. The No Free Lunch Theorem has proven that all algorithms perform exactly the same as the algorithms search for an extremum, when averaged over all cost functions. If an apparently superior algorithm appears to outperform an apparently poor algorithm on some cost functions, then there exist exactly as many cost functions where the apparent poor algorithm will outperform the apparently superior algorithm. In other words, no single algorithm, or ad-hoc protocol, may optimize all possible QoS constraints.

[0006] Two forms of conventional adaptation are a constant algorithm with tunable parameters and a variable algorithm. Conventional ad hoc network protocol research has focused upon the constant algorithm with tunable parameters. In other words, conventional research is searching for a fixed algorithm that has enough degrees of freedom that optimal operation may be found by tuning a fixed set of parameters. This may be due in part to the difficulty in breaking away from the fixed node architecture of Internet Protocol (IP) that has conventionally been used.

[0007] There is great potential in examining the latter form of adaptation, particularly in light of the implication of the No Free Lunch Theorem suggesting that simply tuning a fixed algorithm may not be as optimal as changing to a better algorithm given a different cost function. Two high-level

frameworks that are flexible and customizable enough to allow change in fundamental algorithmic content are programmable networks and active networks. Programmable networks allow control software of a network to be dynamically re-programmed. Active networks are an extreme form of programmable networks that allow code and data to travel through the network, often in the same packet structure. Active packet code may execute on any node along the path that the packet travels.

[0008] Active networks have been applied to both mobile and ad-hoc networking. The applications have focused on solving the problem of mismatch among adaptation mechanisms within individual layers of the IP and improving adaptation to suit the characteristics of ad-hoc network environments.

[0009] The most significant challenge that has been identified with regard to adaptation within ad-hoc networking is the lack of synergistic adaptation among network layers. Early network implementation focused upon network layering as a mechanism for partitioning computer communications into a set of tractable subtasks. Layering resulted in many forms of adaptation occurring simultaneously within the network. However, adaptation in one layer (i.e., congestion control, etc.) may occur in a manner antithetical to adaptation in another layer (i.e., route repair, etc.).

[0010] A "meta"-adaptation view, or how adaptive mechanisms work together, would be extremely useful for ad-hoc network environments. Attempts to develop this meta-adaptation view have resulted in Explicit Link Failure Notification. Congestion and routing each try to adapt based upon limited knowledge of each other, resulting in sub-optimal global behavior. Another example of sub-optimal adaptation behavior is MAC to IP layer address resolution.

[0011] Also, benefit may be gained by examining the lighter, un-layered ad hoc communication of sensor networks. Sensor networks tend to assume large numbers of constrained sensor devices that transmit asymmetrically to a central location. However, ad hoc routing should be implemented on the sensors using as little power and processing as possible. This has conventionally resulted in fewer network layers and better in-network utilization via active networking.

[0012] Another challenge is service migration. Load balancing algorithms may be used to help determine optimal placement of services and routing within an ad-hoc network. A related issue is adaptive cache location. Caching data to improve performance may occur within the network. However, an ad-hoc network adds additional changes in optimizing cache locations within the network. Active caches, or caches implemented within active networks with the capability to migrate from node to node, may address this challenge.

[0013] The No Free Lunch Theorem has produced a myriad of ad-hoc routing protocols. Each protocol may be optimal under different conditions. The choice of a protocol itself might be considered an adaptable parameter. Further, multiple routing protocols may be combined within the same network.

[0014] Ad hoc networking typically requires a more flexible communications architecture than fixed networks in order to maximize QoS. A cross-layer design has been

suggested as a means of achieving improved performance. However, this design lacks flexibility and suggests a fundamental limitation exists in the ability of a single a priori optimization function to meet all QoS requirements in an ad hoc network.

[0015] Active networking and programmable networking enable unprecedented flexibility for ad hoc communication networks. However, in order to best leverage the results of active and programmable networking for ad hoc networks, metrics that best indicate the location and nature of the required flexibility are desired.

#### SUMMARY OF THE INVENTION

[0016] A system in accordance with the present invention operates a wireless ad hoc network. The system includes a plurality of nodes and an active packet. The active packet maintains an optimal quality metric of a service for a plurality of nodes that utilize the service. The active packet utilizes an adaptable algorithm for achieving the optimal quality metric.

[0017] A computer program product in accordance with the present invention maintains optimal quality of service of a wireless network. The computer program product includes a first instruction for maintaining an optimal quality metric of a service for a plurality of nodes that utilize the service and a second instruction for utilizing an adaptable algorithm for achieving the optimal quality metric, the algorithm being applied by an active packet that roams the wireless network.

[0018] A method in accordance with the present invention maintains optimal quality of service of a wireless network. The method includes the steps of maintaining an optimal quality metric of a service for a plurality of nodes that utilize the service and utilizing an adaptable algorithm for achieving the optimal quality metric. The algorithm is applied by an active packet that roams the wireless network.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0019] The foregoing and other features of the present invention will become apparent to one skilled in the art to which the present invention relates upon consideration of the following description of the invention with reference to the accompanying drawings, wherein:

[0020] FIG. 1 is a schematic representation of an example system in accordance with the present invention;

[0021] FIG. 2 is a schematic representation of another example system in accordance with the present invention; and

[0022] FIG. 3 is a schematic representation of still another example system in accordance with the present invention.

#### DESCRIPTION OF PREFERRED EMBODIMENT

[0023] An active network allows intelligent, autonomous packets to roam the active network. These packets may be caches of previously sent information for improving network performance. A system 10 in accordance with the present invention may utilize this roaming to maintain caches in optimal locations in such a dynamically changing network (FIG. 1). The system utilizes autonomous, active caches to intercept and store information likely to be retrieved again by other nodes at an optimal node. The

system further utilizes a fitness function designed to maintain quality of service (QoS) within a "center of mass" of a client node population. Client nodes may be weighted to move a service closer to those client nodes with higher priorities (i.e., client nodes that should receive a higher QoS, etc.).

[0024] The system 10 does not define yet another ad-hoc routing protocol, but rather achieves QoS in a variable topology network. A highest-level performance metric may be qualitatively stated as the degree to which the system is able to meet any reasonable QoS requirement regarding the transmission and reception of information over a variable topology network. It is assumed that QoS requirements are specifically defined as potential trade-offs of traditional networking requirements such as throughput, delay, and degree of information assurance. Information is particularly important for meeting military requirements. In addition to these metrics, a QoS metric may be introduced that is concerned with the best possible QoS achievable in a rapidly changing variable topology environment.

[0025] The system 10 provides adaptation, defined as the ability of an algorithm to change, that is a radical improvement in ad-hoc network capabilities. The code itself may be capable of change, as well as inter-node code transport, deep within a network infrastructure. The ability of the code itself to adapt makes the system 10 extremely flexible and resilient.

[0026] The system 10 defines a metric quantifying a relation of a degree of dynamism to QoS within the ad hoc network. Dynamism quantifies the complexity and rate of change of a variable topology. A critical requirement is how quickly the system 10 may adapt as a function of the complexity and rate of change of the topology. This dynamism may be represented by a specified set of force vectors to be applied to each network node. Thus, a velocity and direction may be applied to each node and the network's characteristics may be monitored as the system 10 responds to applied forces. Based on the above-described No Free Lunch Theorem, a single algorithm or protocol to meet all possible QoS requirements does not exist. Further, only a subset of QoS requirements are likely to be requested by a reasonable user under reasonable operating conditions. However, the envelope of what may be considered reasonable, particularly under military conditions, may constantly expand.

[0027] An ad hoc network desirably eliminates inefficiency caused by competing fitness requirements. In addition to multiple users with potentially competing fitness requirements within a single protocol or application, multiple protocol layers in the same protocol stack may produce inefficiencies due to competing fitness requirements. These inefficiencies become highly exacerbated in an ad hoc environment.

[0028] Adaptation may be defined as autonomic change in network state designed to maintain optimal communication performance. Most network functionality falls under some form of adaptation. The problem with adaptation as defined above is that algorithms, defined a priori (i.e., before installation within the network, etc.), are typically developed to meet high performance standards for specific predefined cases within the network. These algorithms may not have the flexibility, or adaptability, required to meet unanticipated

conditions. However, the definition of adaptation may include self-directed change by an algorithm designed to maintain optimal performance. Adaptation may be a self-directed change in an algorithm, or program code, designed to maintain optimal ad-hoc network performance.

[0029] Adaptation may be related to traditional metrics. Route discovery time, or route acquisition time including reactive protocols, may be a specific metric that is critical to a variable topology network. Route discovery time may be influenced by a location announcement interval (i.e., beaconing, a hello message, etc.). End-end delay and throughput are metrics that may also be considered. These metrics may be influenced by packet loss rate, route reconstruction time, and routing overhead, which may be additional candidate metrics.

[0030] Changing an algorithm requires changing code. Services require capability to adjust their location within an ad hoc network in order to maximize QoS to all client nodes. Code that remains resident to only one node may not reside in an optimal location given ad hoc movement patterns. Installing all potentially useful code on all nodes creates much overhead, particularly for low capability ad hoc nodes and sensors. It is desirable that code have the capability to move from node to node. The location and movement of a service affect network performance.

[0031] The factors influencing service migration include not only basic route acquisition delays, but also the size of a service code. Route acquisition delays have the impact of potentially providing false network topology information and may cause a service to migrate towards an incorrect location. Large service code size may be due to a complex service that is being offered or due to the overhead of a service location algorithm itself.

[0032] The system **10** in accordance with the present invention provides the ability to maintain a service at an optimal location relative to a set of mobile client nodes. Specifically, because nodes are mobile and the direction and velocity of the nodes may be highly variable, nodes may be temporarily or permanently unreachable. The system **10** assigns a penalty for 'no quality of service'.

[0033] Additionally, the system **10** may define a metric that captures the precision, including overhead, with which the network is able to maintain an optimal location for the service that maximizes overall QoS normalized by the amount of dynamism in the network topology. By means of simulation, an optimal location may be determined at all times.

[0034] As discussed above, a fundamental limitation exists in the ability of a single a priori optimization function to meet all QoS requirements. Metrics that best indicate the location and nature of required flexibility are desirable. A system **10** in accordance with the present invention provides such a metric as applied to an active ad-hoc network. The system **10** maximizes QoS by adaptation of algorithms, rather than simply tuning static algorithms. Algorithms may change form either in vivo within a node, or be defined a priori, before network operation. In either case, the code implementing the algorithm may migrate to the best service location in the network as the network topology changes. The topology changes due to node movement patterns.

[0035] Network topology is thus dependent upon the effects of node movement and those effects may be exam-

ined via vector fields. Vector field representation of node movement may be directly applied to the estimation of QoS and the requirements for service migration performance (i.e., service transmission rate as a function of code size, expected link transmission rate, etc.).

[0036] FIG. 1 shows an example representation of a system **10** in accordance with the present invention. The dots represent mobile nodes. The caches are also located on mobile nodes. The QoSWeights are attributes of client nodes' requests for services. Client nodes may reside on any nodes. The caches are mobile and move themselves (and their contents) to an optimal location in the network. The best location may be a simple solution to the objective function  $\text{Min } E(x) + \text{Var}(x)$  where  $E$  is Expected Value,  $\text{Var}$  is Variance, and  $x$  is a hop count for each client node to its cache multiplied by its QoSWeight (i.e.,  $h * \text{QoSWeight}$  where  $h$  is hop count). The mobile caches may thus be pulled into the center of mass of the client nodes.

[0037] In accordance with the present invention, a computer program product **200** maintains optimal quality of service of a wireless network (FIG. 2). The computer program product **200** includes a first instruction **201** for maintaining an optimal quality metric of a service for a plurality of nodes that utilize the service and a second instruction **202** for utilizing an adaptable algorithm for achieving the optimal quality metric, the algorithm being applied by an active packet that roams **203** the wireless network.

[0038] In accordance with the present invention, a method **300** maintains optimal quality of service of a wireless network (FIG. 3). The method **300** includes the steps of maintaining **301** an optimal quality metric of a service for a plurality of nodes that utilize the service and utilizing **302** an adaptable algorithm for achieving the optimal quality metric. The algorithm is applied by an active packet that roams **303** the wireless network.

[0039] In order to provide a context for the various aspects of the present invention, the following discussion is intended to provide a brief, general description of a suitable computing environment in which the various aspects of the present invention may be implemented. While the invention has been described above in the general context of computer-executable instructions of a computer program that runs on a computer, those skilled in the art will recognize that the invention also may be implemented in combination with other program modules. Generally, program modules include routines, programs, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the inventive methods may be practiced with other computer system configurations, including single-processor or multiprocessor computer systems, minicomputers, mainframe computers, as well as personal computers, hand-held computing devices, microprocessor-based or programmable consumer electronics, and the like. The illustrated aspects of the invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications argument model. However, some, if not all aspects of the invention can be practiced on stand-alone computers. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

**[0040]** An exemplary system for implementing the various aspects of the invention includes a conventional server computer, including a processing unit, a system memory, and a system bus that couples various system components including the system memory to the processing unit. The processing unit may be any of various commercially available processors. Dual microprocessors and other multi-processor architectures also can be used as the processing unit. The system bus may be any of several types of bus structure including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of conventional bus architectures. The system memory includes read only memory (ROM) and random access memory (RAM). A basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within the server computer, such as during start-up, is stored in ROM.

**[0041]** The server computer further includes a hard disk drive, a magnetic disk drive, e.g., to read from or write to a removable disk, and an optical disk drive, e.g., for reading a CD-ROM disk or to read from or write to other optical media. The hard disk drive, magnetic disk drive, and optical disk drive are connected to the system bus by a hard disk drive interface, a magnetic disk drive interface, and an optical drive interface, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions, etc., for the server computer. Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and a CD, it should be appreciated by those skilled in the art that other types of media which are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, and the like, may also be used in the exemplary operating environment, and further that any such media may contain computer-executable instructions for performing the methods of the present invention.

**[0042]** A number of program modules may be stored in the drives and RAM, including an operating system, one or more application programs, other program modules, and program data. A user may enter commands and information into the server computer through a keyboard and a pointing device, such as a mouse. Other input devices (not shown) may include a microphone, a joystick, a game pad, a satellite dish, a scanner, or the like. These and other input devices are often connected to the processing unit through a serial port interface that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor or other type of display device is also connected to the system bus via an interface, such as a video adapter. In addition to the monitor, computers typically include other peripheral output devices (not shown), such as speaker and printers.

**[0043]** The server computer may operate in a networked environment using logical connections to one or more remote computers, such as a remote client computer. The remote computer may be a workstation, a server computer, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the server computer. The logical connections include a local area network (LAN) and a wide area network

(WAN). Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the internet.

**[0044]** When used in a LAN networking environment, the server computer is connected to the local network through a network interface or adapter. When used in a WAN networking environment, the server computer typically includes a modem, or is connected to a communications server on the LAN, or has other means for establishing communications over the wide area network, such as the internet. The modem, which may be internal or external, is connected to the system bus via the serial port interface. In a networked environment, program modules depicted relative to the server computer, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

**[0045]** In accordance with the practices of persons skilled in the art of computer programming, the present invention has been described with reference to acts and symbolic representations of operations that are performed by a computer, such as the server computer, unless otherwise indicated. Such acts and operations are sometimes referred to as being computer-executed. It will be appreciated that the acts and symbolically represented operations include the manipulation by the processing unit of electrical signals representing data bits which causes a resulting transformation or reduction of the electrical signal representation, and the maintenance of data bits at memory locations in the memory system (including the system memory, hard drive, floppy disks, and CD-ROM) to thereby reconfigure or otherwise alter the computer system's operation, as well as other processing of signals. The memory locations where such data bits are maintained are physical locations that have particular electrical, magnetic, or optical properties corresponding to the data bits.

**[0046]** It will be understood that the above description of the present invention is susceptible to various modifications, changes and adaptations, and the same are intended to be comprehended within the meaning and range of equivalents of the appended claims. The presently disclosed embodiments are considered in all respects to be illustrative, and not restrictive. The scope of the invention is indicated by the appended claims, rather than the foregoing description, and all changes that come within the meaning and range of equivalence thereof are intended to be embraced therein.

Having described the invention, we claim:

1. A system for operating a wireless ad hoc network, said system comprising:

a plurality of nodes; and

an active packet for maintaining an optimal quality metric of a service for said plurality of nodes that utilize the service, said active packet utilizing an adaptable algorithm for achieving the optimal quality metric.

2. The system as set forth in claim 1 further including a fitness function for quantifying the quality of the service.

3. The system as set forth in claim 2 wherein at least one of said plurality of nodes has a weight value other than 1.0.

4. The system as set forth in claim 3 wherein said fitness function allows functional evolution of said plurality of nodes.

5. The system as set forth in claim 4 wherein said system relocates the service from one of said plurality of nodes to another of said plurality of nodes in order to increase the quality metric.

6. The system as set forth in claim 5 wherein the algorithm autonomously changes in order to adjust to changing conditions of said plurality of nodes.

7. The system as set forth in claim 6 wherein the changing conditions of said plurality of nodes are represented by vector fields that are directly applied to an estimation of the quality metric.

8. A computer program product for maintaining optimal quality of service of a wireless network, said computer program product comprising:

a first instruction for maintaining an optimal quality metric of a service for a plurality of nodes that utilize the service; and

a second instruction for utilizing an adaptable algorithm for achieving the optimal quality metric, the algorithm being applied by an active packet that roams the wireless network.

9. The computer program product as set forth in claim 8 further including a third instruction for quantifying the quality of the service using a fitness function.

10. The computer program product as set forth in claim 8 further including a third instruction for assigning weight values to each of said plurality of nodes.

11. The computer program product as set forth in claim 8 further including a third instruction for allowing functional evolution of the plurality of nodes by a fitness function.

12. The computer program product as set forth in claim 8 further including a third instruction for relocating the service

from one of the plurality of nodes to another of the plurality of nodes in order to increase the quality metric.

13. The computer program product as set forth in claim 8 further including a third instruction for autonomously changing the algorithm to adjust to changing conditions of the plurality of nodes.

14. The computer program product as set forth in claim 8 further including a third instruction for representing the plurality of nodes by vector fields in order to estimate the quality metric.

15. A method for maintaining optimal quality of service of a wireless network, said method comprising the steps of:

maintaining an optimal quality metric of a service for a plurality of nodes that utilize the service; and

utilizing an adaptable algorithm for achieving the optimal quality metric, the algorithm being applied by an active packet that roams the wireless network.

16. The method as set forth in claim 15 further including the step of quantifying the quality of the service using a fitness function.

17. The method as set forth in claim 15 further including the step of assigning weight values to each of said plurality of nodes.

18. The method as set forth in claim 15 further including the step of allowing functional evolution of the plurality of nodes by a fitness function.

19. The method as set forth in claim 15 further including the step of relocating the service from one of the plurality of nodes to another of the plurality of nodes in order to increase the quality metric.

20. The method as set forth in claim 15 further including the step of autonomously changing the algorithm to adjust to changing conditions of the plurality of nodes.

\* \* \* \* \*