



US 20120031984A1

(19) **United States**

(12) **Patent Application Publication**
Feldmeier et al.

(10) **Pub. No.: US 2012/0031984 A1**

(43) **Pub. Date: Feb. 9, 2012**

(54) **PERSONALIZED BUILDING COMFORT CONTROL**

Publication Classification

(75) Inventors: **Mark Feldmeier**, Waukesha, WI (US); **Joseph Paradiso**, Medford, MA (US)

(51) **Int. Cl.**
F24F 7/00 (2006.01)

(52) **U.S. Cl.** **236/49.3**

(73) Assignee: **MASSACHUSETTS INSTITUTE OF TECHNOLOGY**, Cambridge, MA (US)

(57) **ABSTRACT**

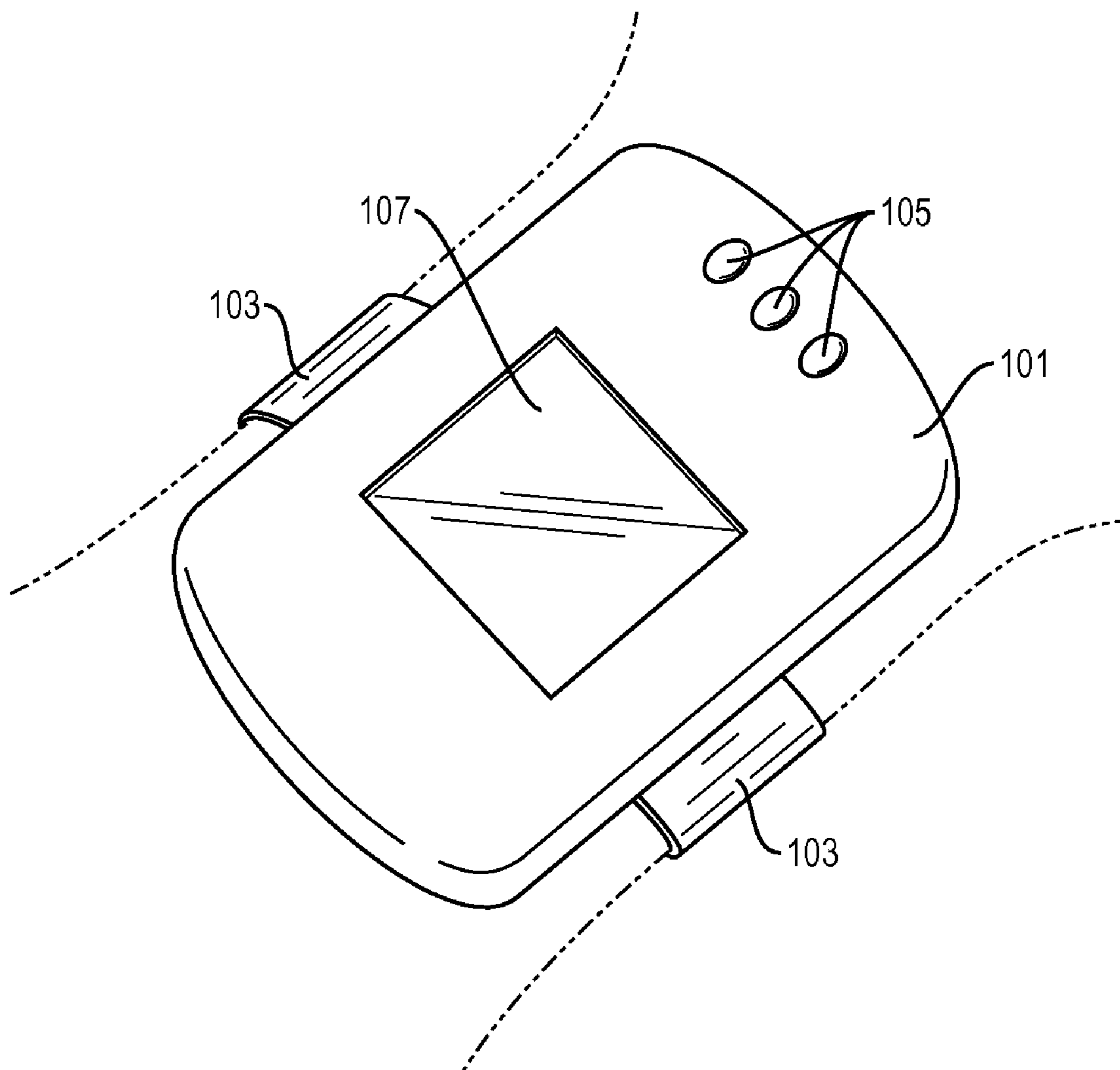
(21) Appl. No.: **13/197,375**

(22) Filed: **Aug. 3, 2011**

In exemplary implementations of this invention, control apparatus for a HVAC system provides personalized comfort control. It can adjust local conditions in different rooms within a building in order to maximize the perceived comfort of individual occupants. The control apparatus locates individuals within a building. For each individual, it senses temperature, humidity and other parameters at the individual's location, calculates a comfort metric indicative of the user's comfort, and can control the flow of chilled or heated air to the individual's location in order to adjust local conditions to maximize the individual's comfort.

Related U.S. Application Data

(60) Provisional application No. 61/370,338, filed on Aug. 3, 2010.



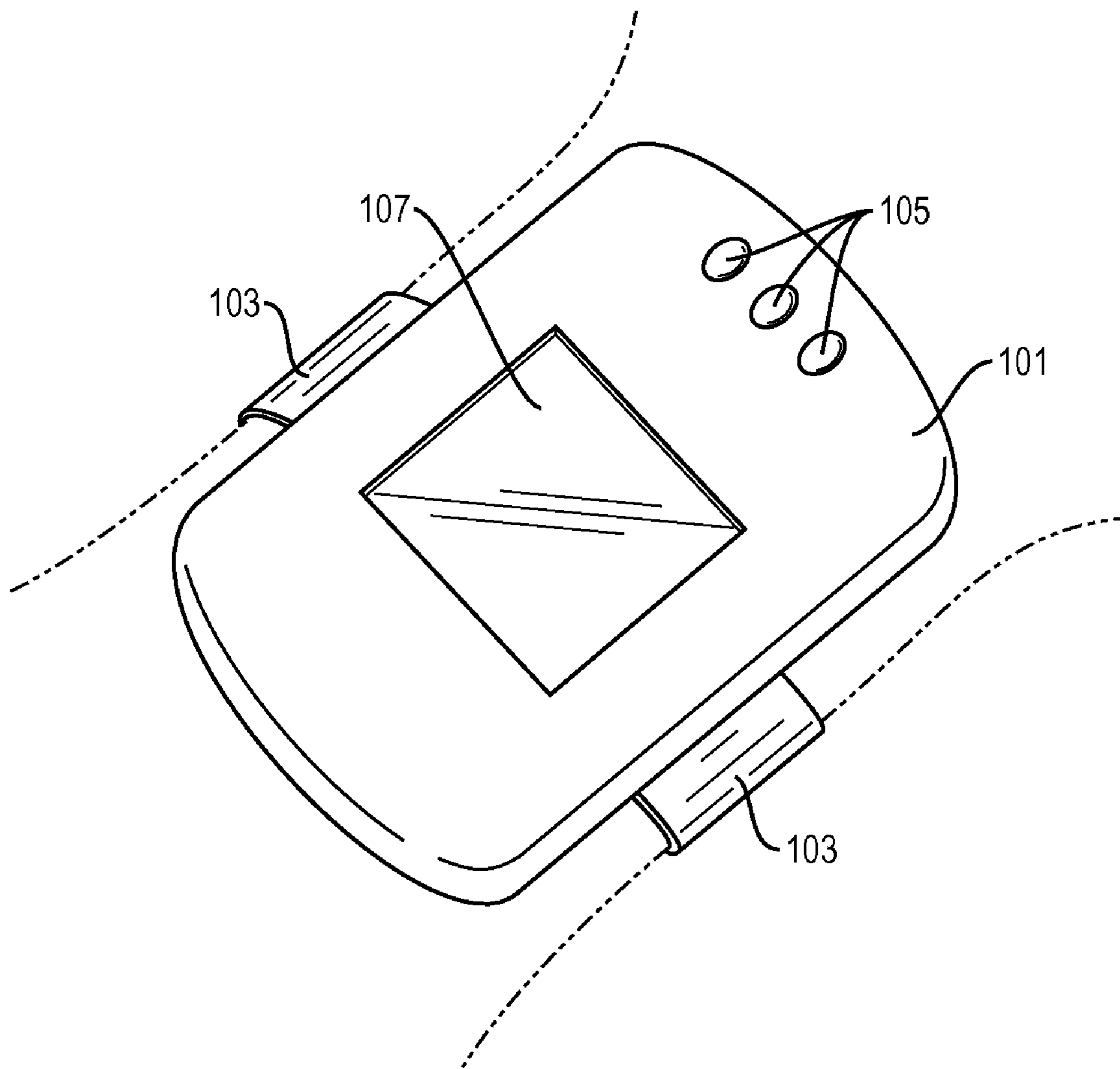


FIG. 1

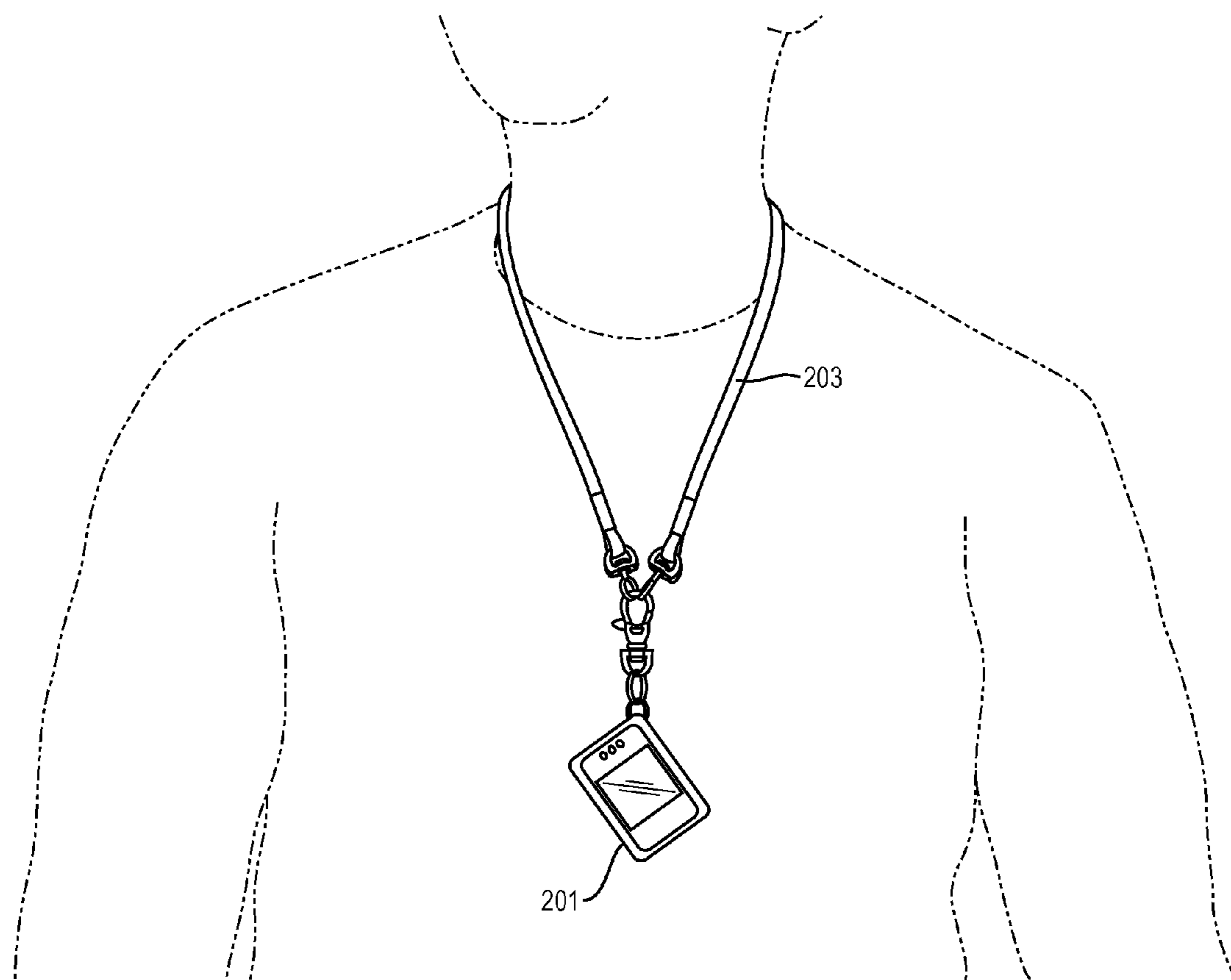


FIG. 2

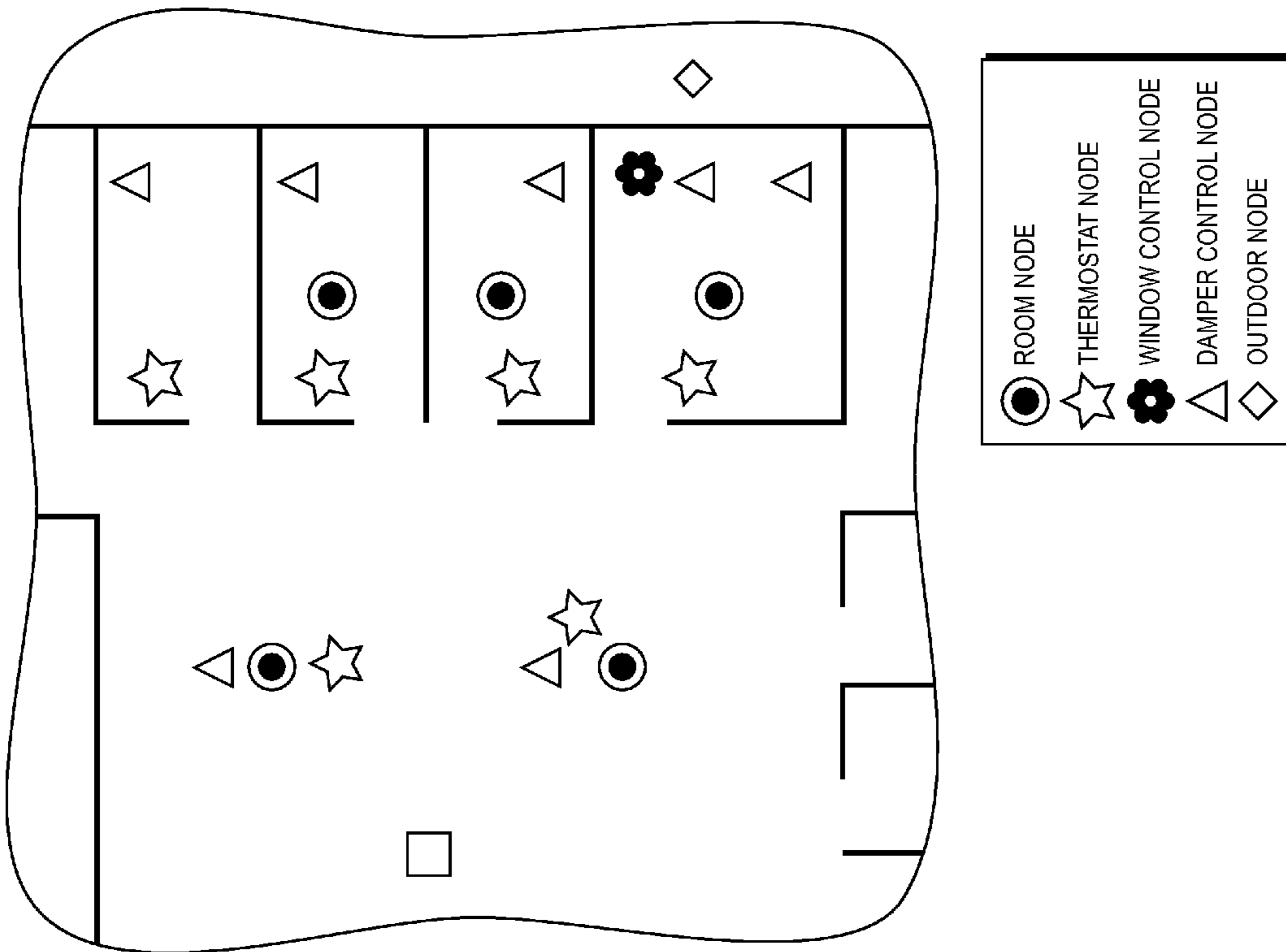


FIG. 3

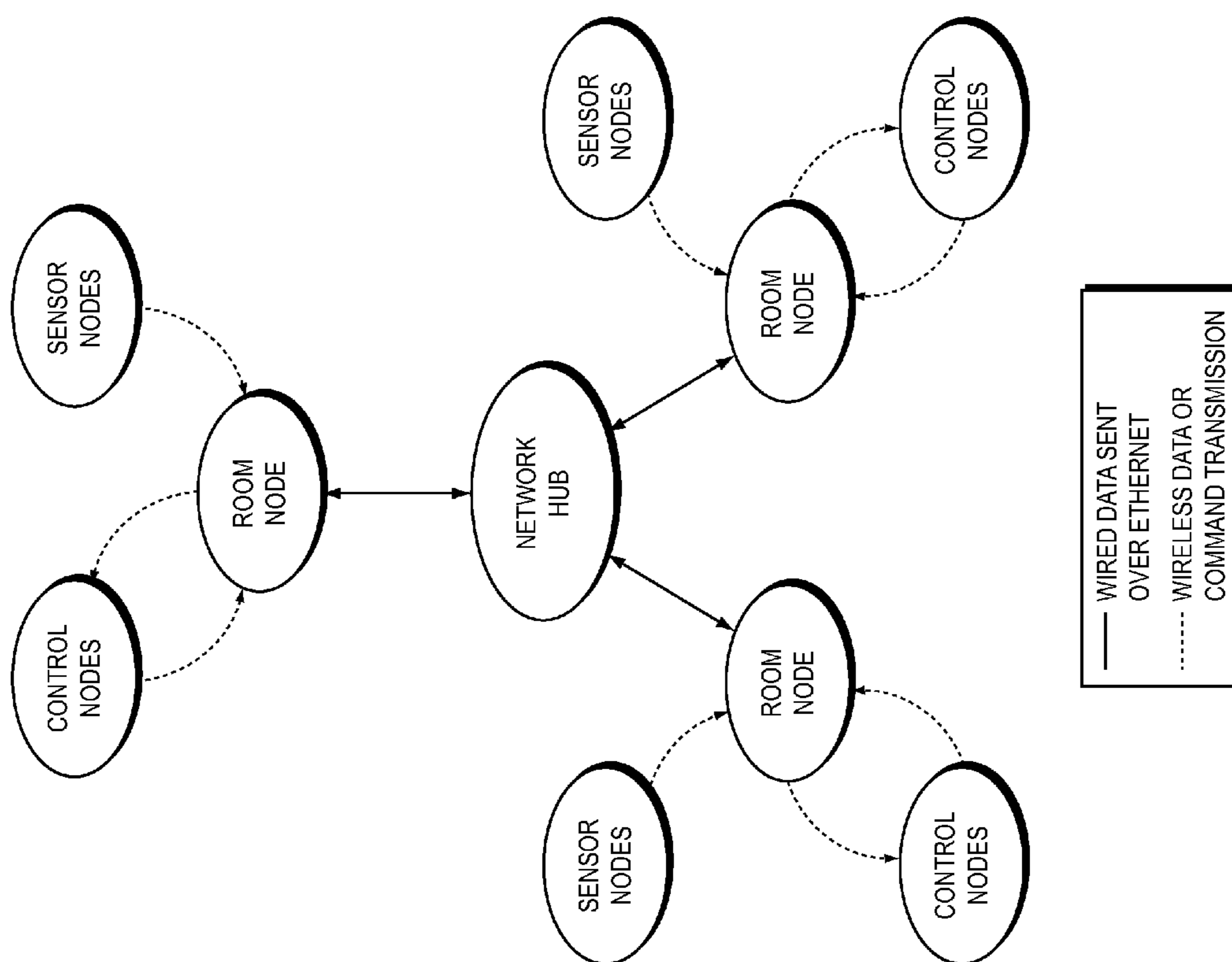


FIG. 4

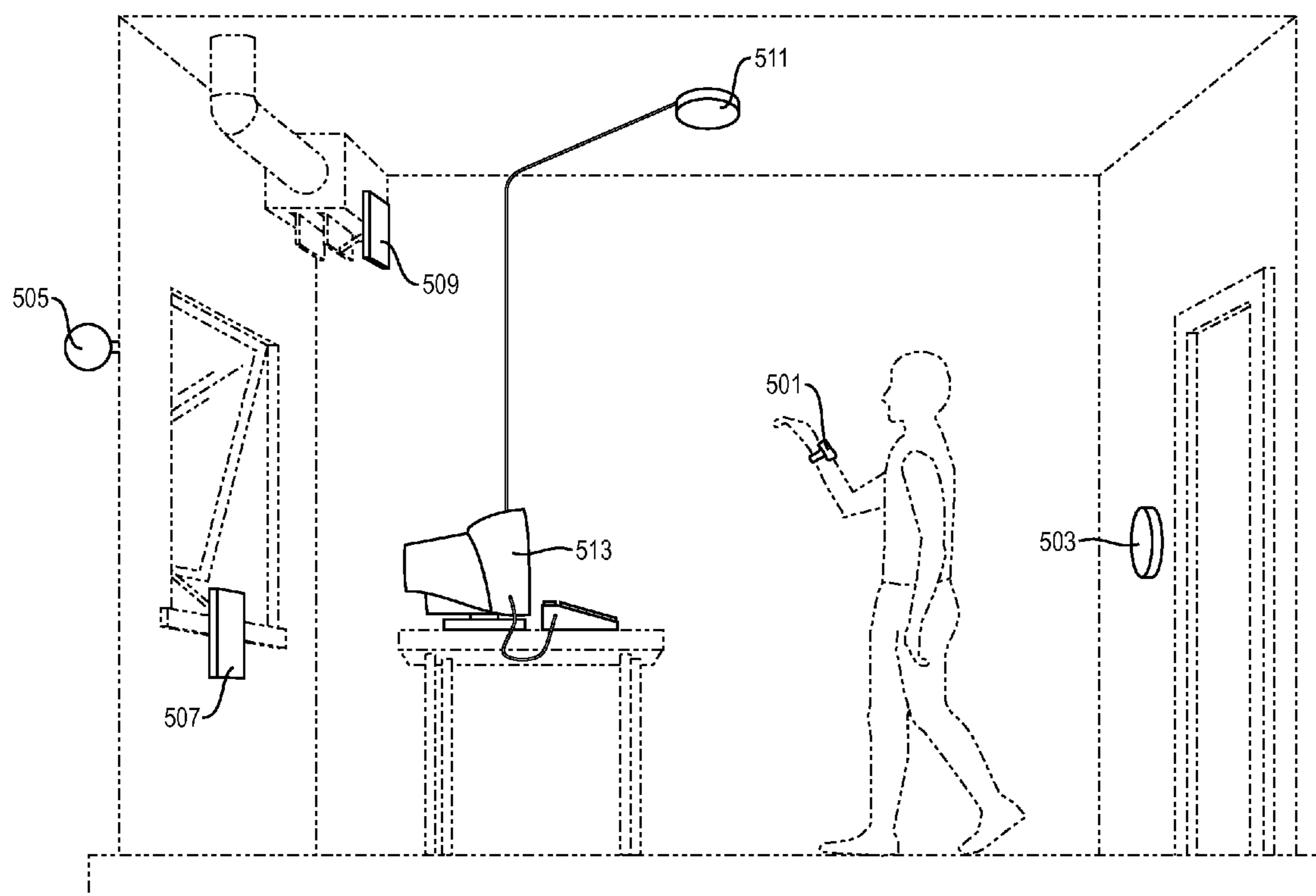


FIG. 5

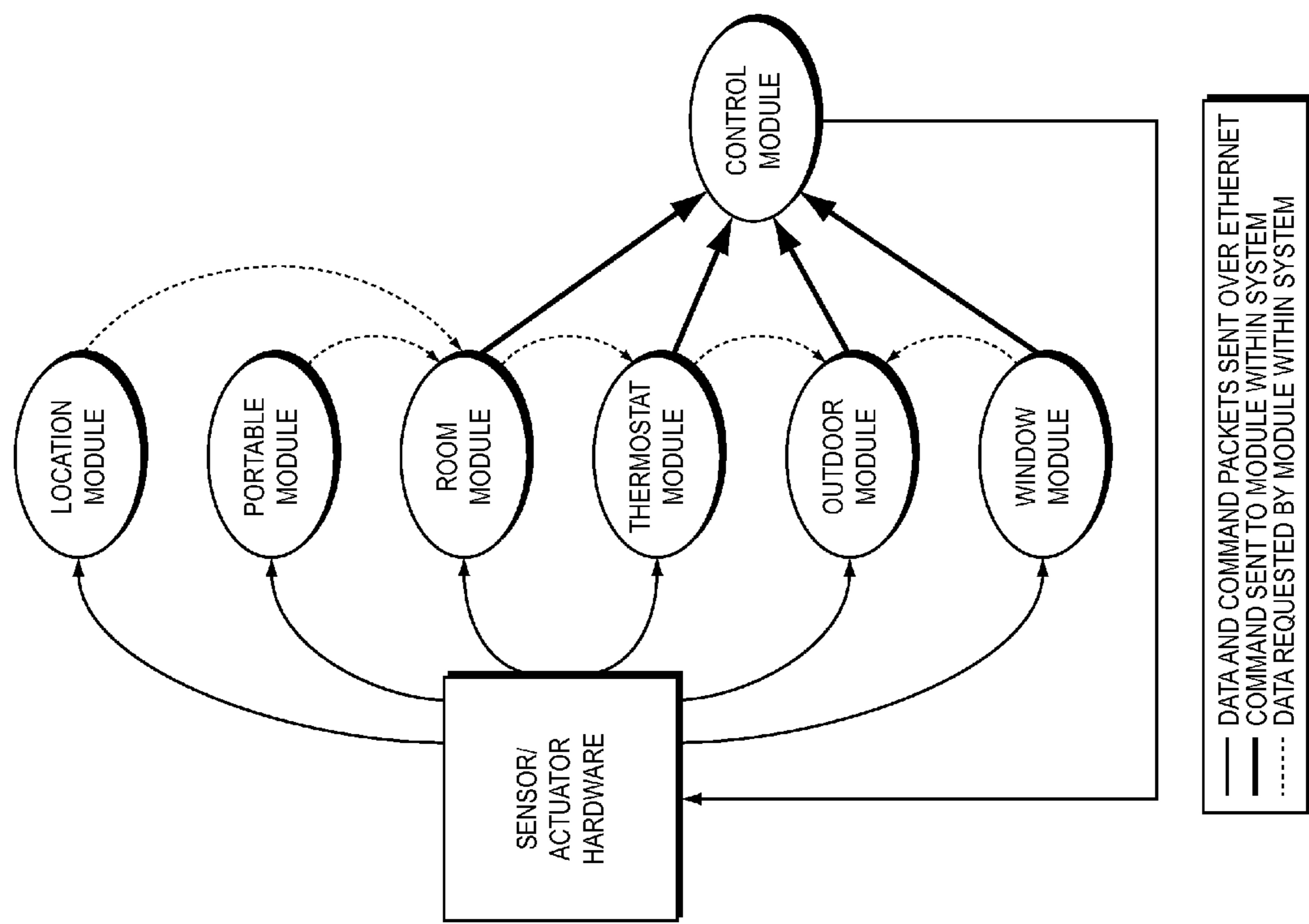


FIG. 6

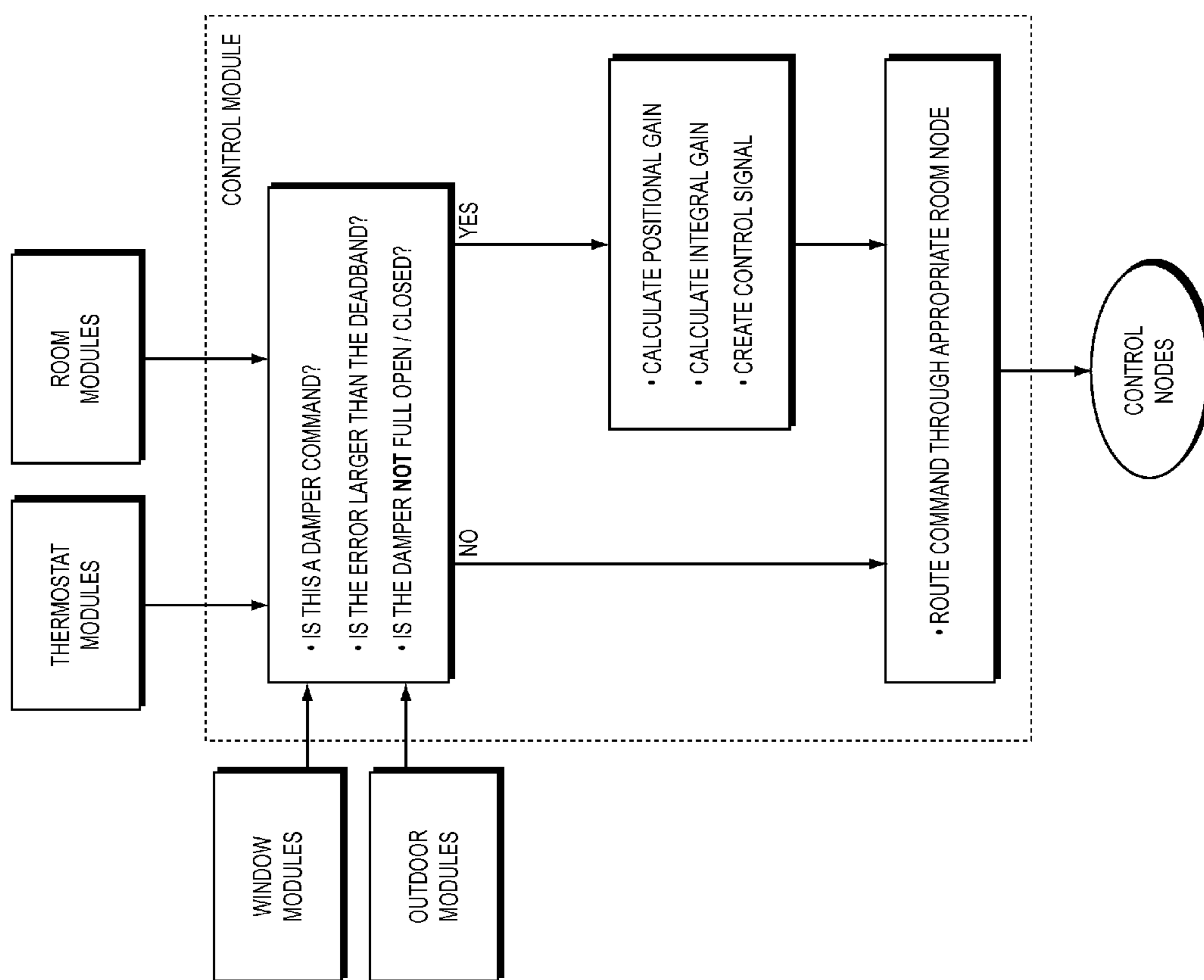


FIG. 7

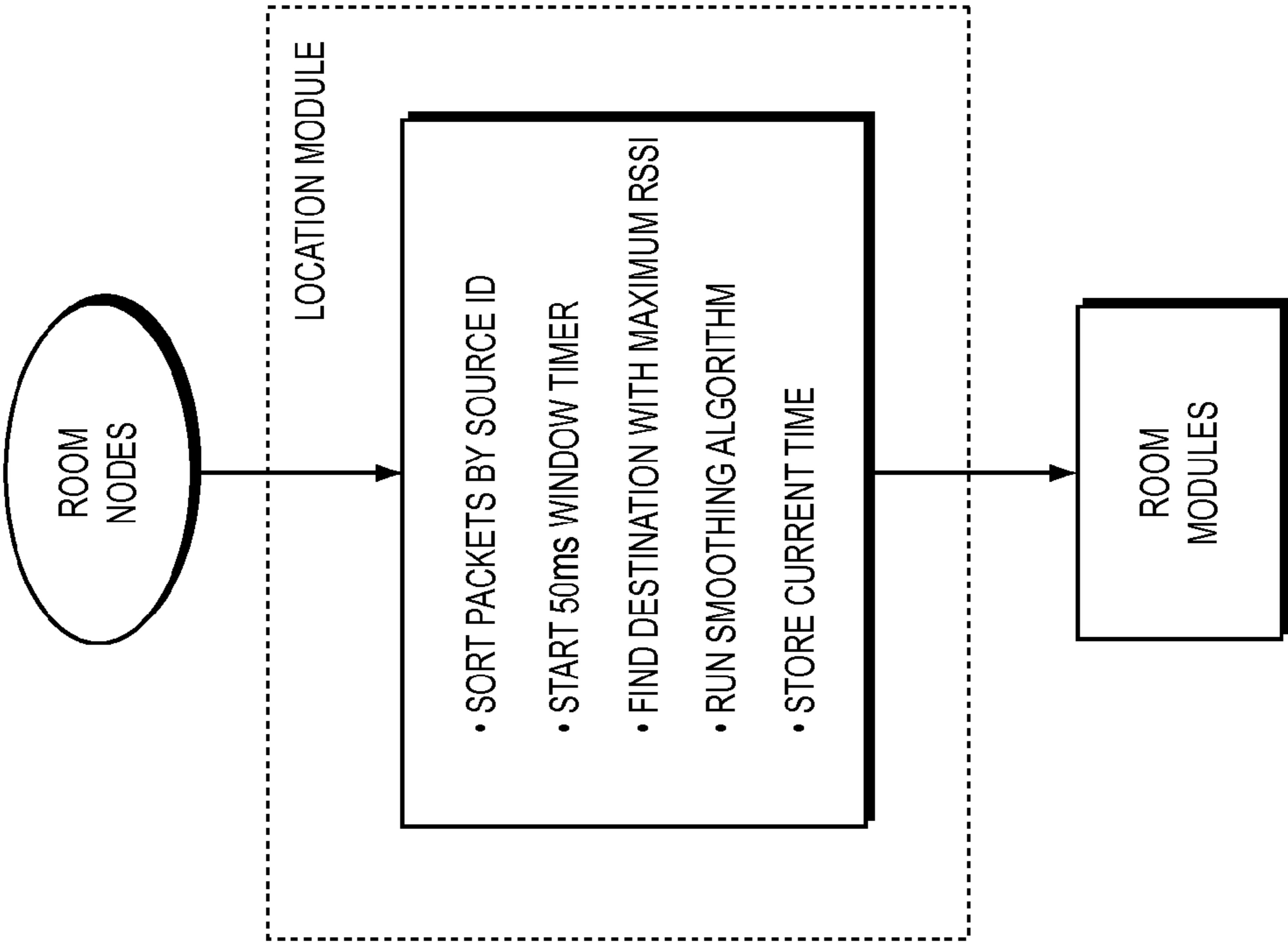


FIG. 8

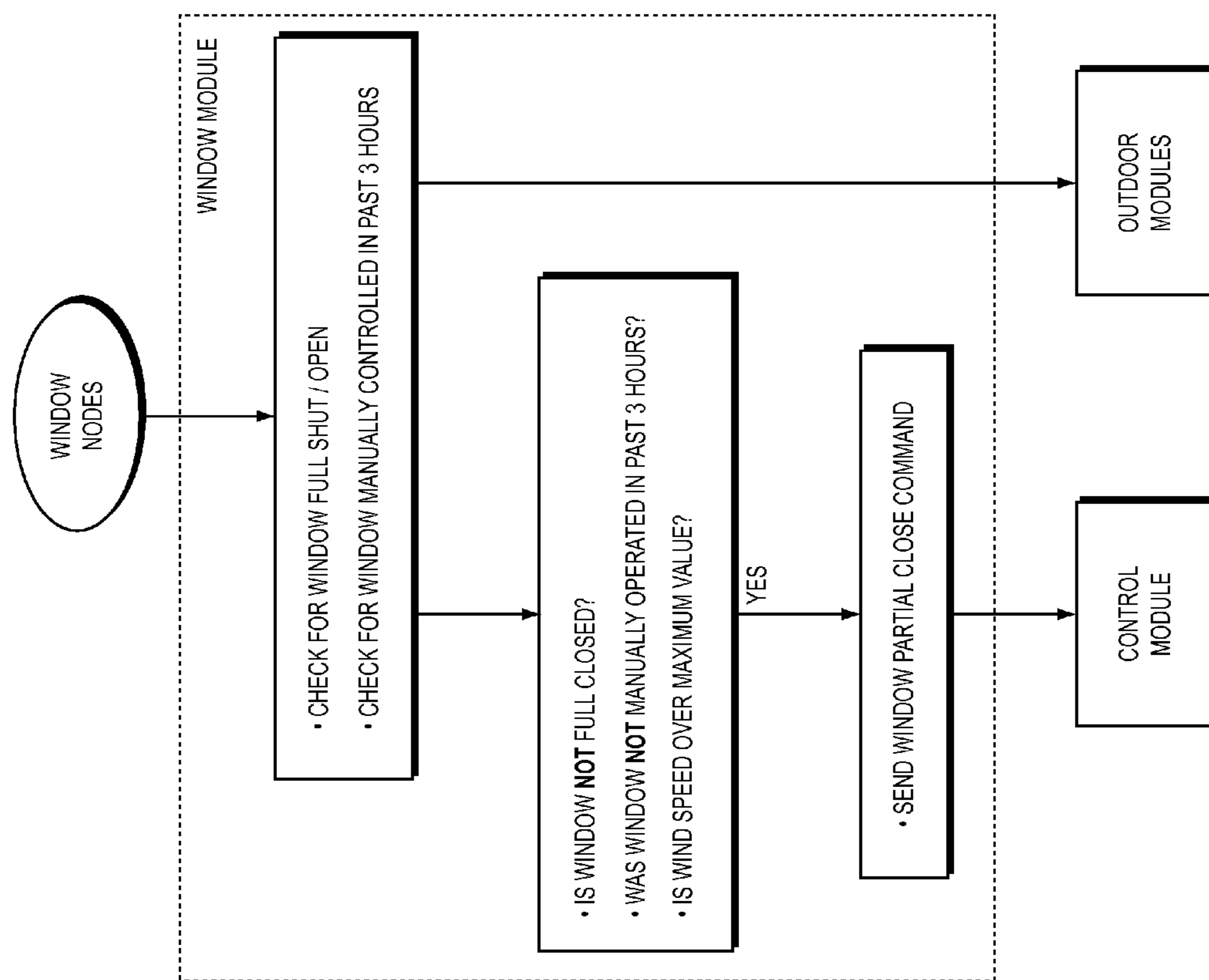


FIG. 9

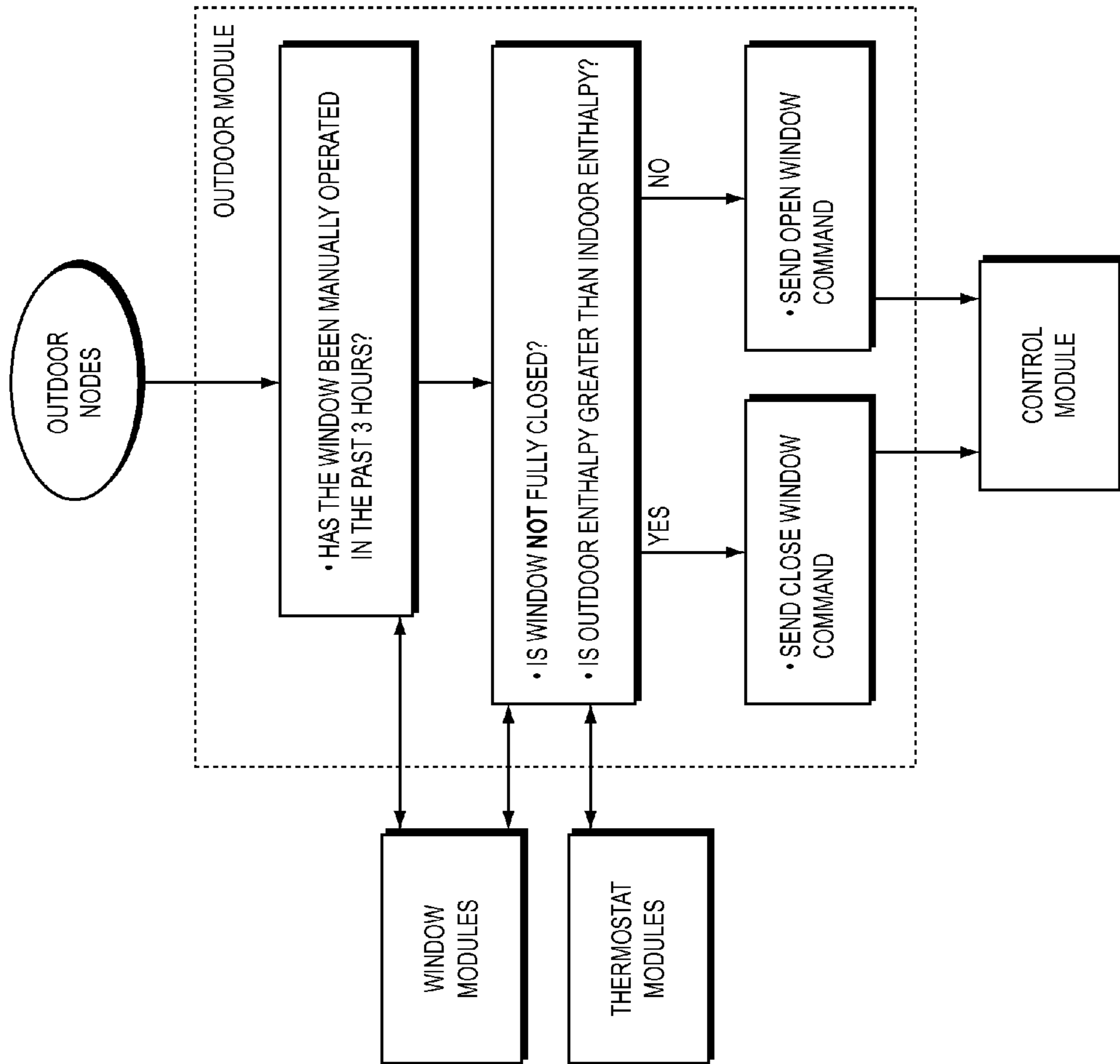


FIG. 10

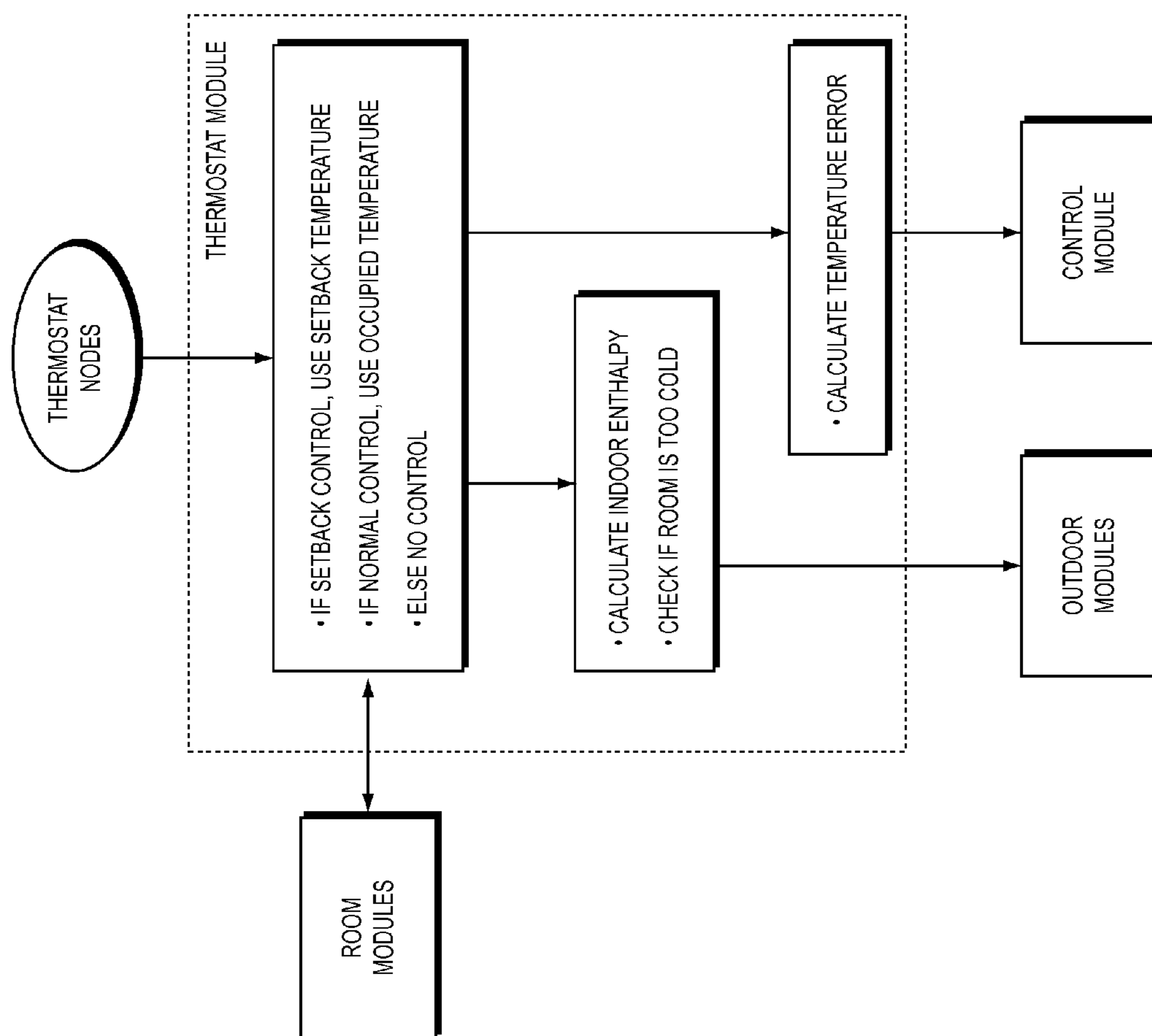


FIG. 11

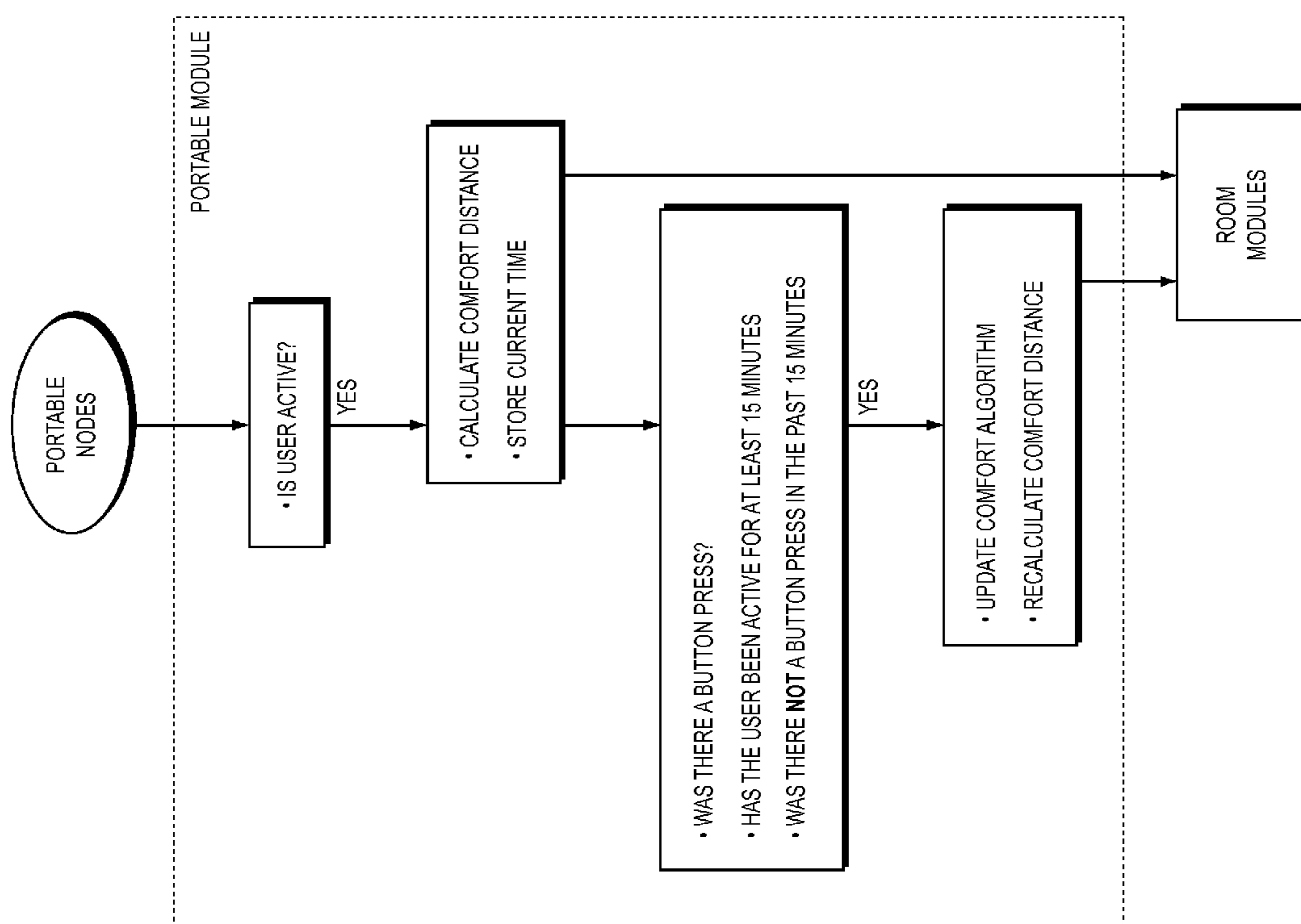


FIG. 12

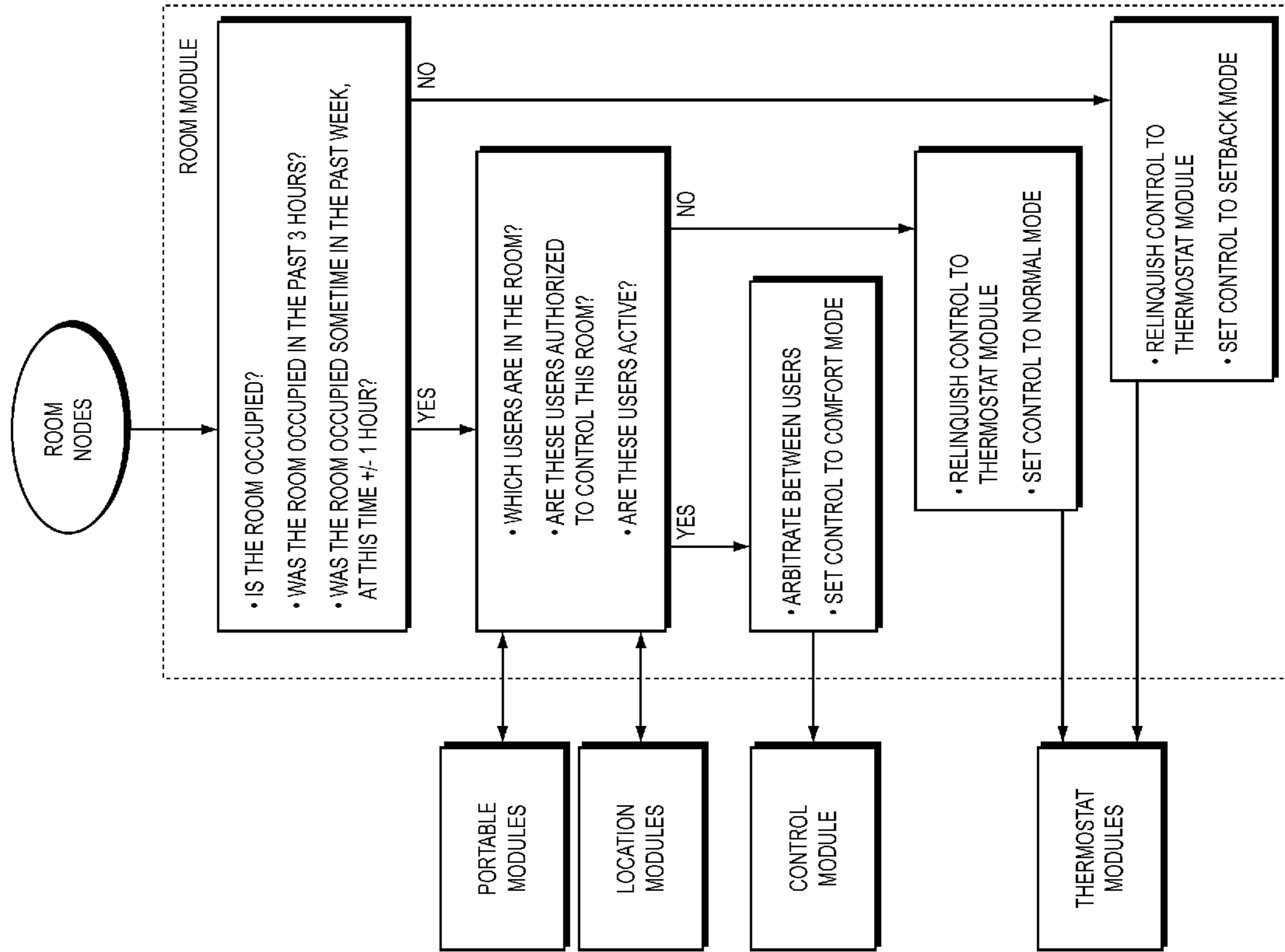


FIG. 13

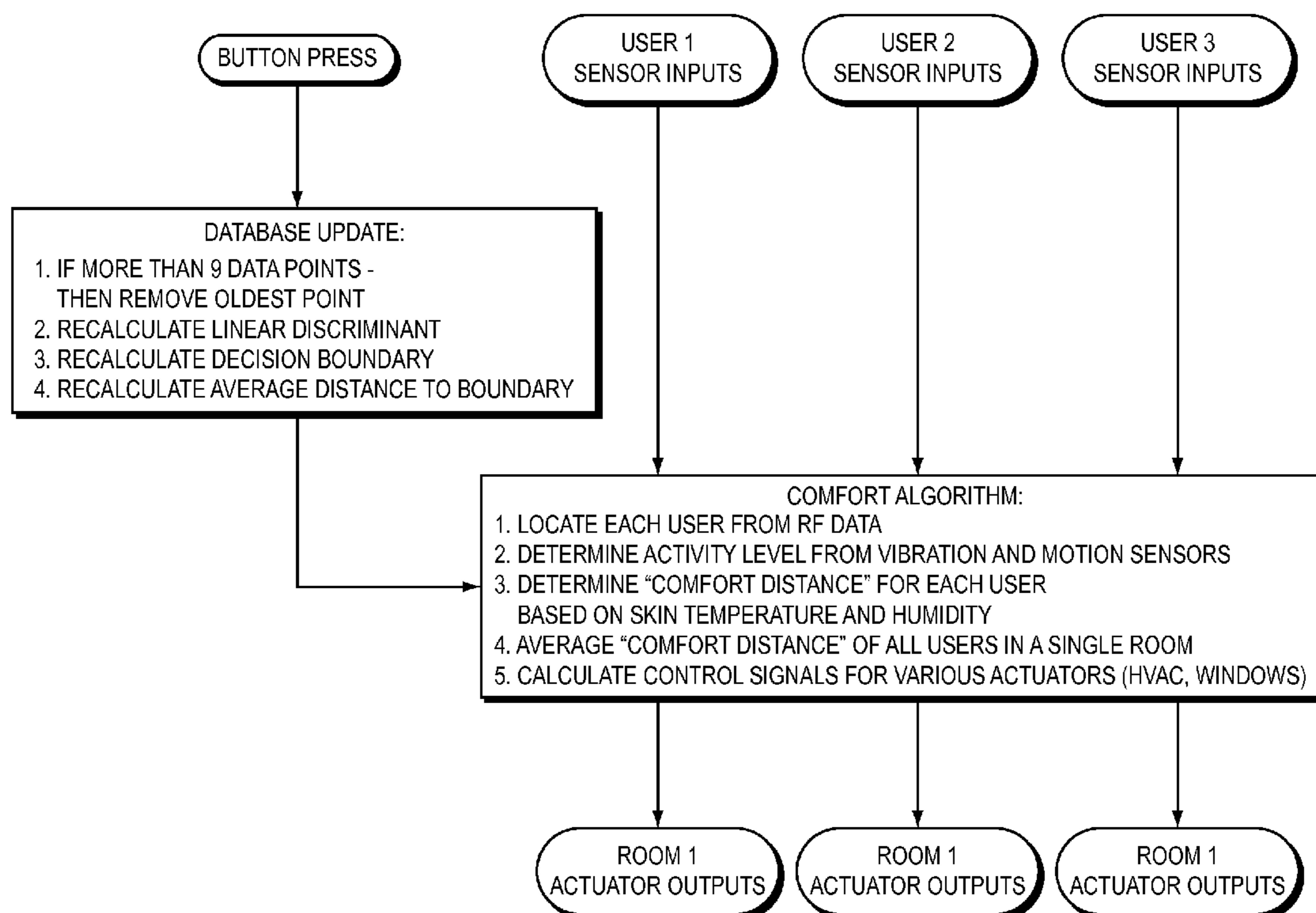


FIG. 14

PERSONALIZED BUILDING COMFORT CONTROL

RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application Ser. No. 61,370,338, filed Aug. 3, 2010, the entire disclosure of which is herein incorporated by reference.

FIELD OF THE TECHNOLOGY

[0002] The present invention relates generally to controls for heating, ventilation and air conditioning (HVAC).

Computer Program Listing

[0003] Attached are five ASCII text files: (1) appd2_portable.txt, created Jul. 26, 2011 with a size of about 39 KB (the “Portable Node Firmware”); (2) appd1_damper.txt, created Jul. 26, 2011 with a size of about 43 KB (the “Damper Node Firmware”); (3) appf1_window.txt, created Jul. 25, 2011 with a size of about 44 KB (the “Window Node Firmware”); (4) appg1_room.txt, created Jul. 25, 2011 with a size of about 40 KB (the “Room Node Firmware”); and (5) apph1_control.txt, created Jul. 26, 2011 with a size of about 39 KB (the “Control Node Firmware”). Each of these five ASCII text files comprises a computer program listing for firmware in a prototype implementation of this invention. These five ASCII text files are each incorporated by reference herein.

SUMMARY

[0004] Different individuals are comfortable under different conditions. What is too cold or too warm for one person is just right for another.

[0005] In exemplary implementations of this invention, control apparatus for a HVAC system provides personalized comfort control. It can adjust local conditions in different rooms within a building in order to maximize the perceived comfort of individual occupants. The control apparatus locates individuals within a building. For each individual, it senses temperature, humidity and other parameters at the individual’s location, calculates a comfort metric indicative of the user’s comfort, and can control the flow of chilled or heated air to the individual’s location in order to adjust local conditions to maximize the individual’s comfort.

[0006] In conventional HVAC systems, temperature is used as a setpoint. In contrast, in exemplary implementations of this invention, the comfort of individual occupants is used as a setpoint. This comfort setpoint applies locally (e.g., to the room which the user occupies) rather than globally for the entire building.

[0007] In exemplary implementations, users may input data indicative of their comfort state. In some implementations, this input is limited to whether the user feels hot, cold or neutral. In others, user input can further convey the magnitude of discomfort—e.g., how hot or cold the user feels. Also, in some implementations, the system detects and assesses other user behavior, such as opening or closing window shades, to infer the user’s comfort state.

[0008] The control system uses machine learning to adapt to changing circumstances. It employs both pre-assigned knowledge (e.g., which dampers affect air flow in which rooms) and dynamically acquired knowledge based on updates from sensor readings and inputs from users.

[0009] In exemplary implementations, the control apparatus can arbitrate between competing needs of different occupants (e.g., if two people are in the same room). Also, it can arbitrate between competing goals (e.g., conserving energy vs. maximizing comfort of the individual occupants).

[0010] In exemplary implementations, users wear or carry portable nodes with embedded sensors that detect temperature, humidity, ambient light level and inertial activity. For example, the portable nodes may comprise lightweight, dedicated sensor modules that are mounted on a wrist band.

[0011] In exemplary implementations, other sensors are distributed in static positions in the building (and in some cases, outside the building) to detect temperature, humidity, light level, motion and wind speed. Data is transmitted wirelessly from some of the sensors (e.g., those embedded in the portable nodes) and by Ethernet or other wired connection from other sensors (sensors mounted on the wall). The data is processed in a central hub, in order to determine the positions of individual occupants and the local conditions at their respective positions.

[0012] In some implementations, sensors are densely distributed throughout the rooms, so that a user has a sensor nearby, regardless of the user’s position. For example, dozens of temperature sensors may be positioned in each room. Each of these temperature sensors may have very low power consumption and be solar-powered.

[0013] If a sufficiently dense distribution of sensors is used, the sensor data can be related to the user based on the user’s position. In that case, sensors embedded in portable nodes may not be needed. Instead, the system can simply locate the occupant and determine local conditions based on readings from a nearby sensor.

[0014] In some implementations, cell phones, smart phones, or wrist watches may be used to perform the functions of a portable node, such as assessing occupant identity, location, activity level, and comfort status.

[0015] In exemplary implementations, a comfort algorithm is used to determine the current comfort of the individual occupants. For example, this algorithm may use a Fisher Linear Discriminant or a K-Nearest-Neighbor analysis.

[0016] In exemplary implementations, the control apparatus can adjust the flow of heated or chilled area to the individual’s location by controlling motors to open or close a damper in a Variable Air Volume box. It can also control motors to open or close windows. For example, if the system detects that outside air is cooler than inside air, and determines that an occupant of the room would be more comfortable if the room were cooler, then the system may control a motor to open a window to let in the cooler outer air. Alternatively, or in addition, the control system can do one or more of the following: (1) control fans; (2) control motors to open or close window shades; (3) control light switches or light dimmers (to affect thermal load); (4) control power supplied by electrical outlets (for example, to turn on or off lamps or other energy sinks); or (5) adjust temperature setpoints for zones in an HVAC system.

[0017] In some implementations, the control system provides perceptible feedback to the users. For example, this feedback may indicate the system’s belief about the user’s current comfort, the action that is being taken, or that comfort is intentionally being compromised for energy reasons. Such feedback may, for example, be visual, auditory or haptic.

[0018] The above description of the present invention is just a summary. It is intended only to give a general introduction

to some illustrative implementations of this invention. It does not describe all of the details of this invention. This invention may be implemented in many other ways.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] FIG. 1 shows a portable node attached to a wrist band.

[0020] FIG. 2 shows a portable node attached to a lanyard.

[0021] FIG. 3 is a floor plan, showing the position of hardware nodes in a deployment of a prototype of this invention.

[0022] FIG. 4 shows the topology of the network, in this prototype.

[0023] FIG. 5 shows hardware used in this prototype.

[0024] FIG. 6 is a high level flowchart of the control software system, in this prototype.

[0025] FIG. 7 is a high level flowchart of the Control Module, in this control system.

[0026] FIG. 8 is a high level flowchart of the Location Module, in this control system.

[0027] FIG. 9 is a high level flowchart of the Window Module, in this control system.

[0028] FIG. 10 is a high level flowchart of the Outdoor Module, in this control system.

[0029] FIG. 11 is a high level flowchart of the Thermostat Module, in this control system.

[0030] FIG. 12 is a high level flowchart of the Portable Module, in this control system.

[0031] FIG. 13 is a high level flowchart of the Room Module, in this control system.

[0032] FIG. 14 is a high level flowchart of a comfort algorithm, in this prototype.

[0033] The above Figures illustrate some illustrative implementations of this invention, or provide information that relates to those implementations. The above Figures do not show all of the details of this invention.

DETAILED DESCRIPTION

[0034] The following is a description of hardware employed in a prototype of this invention. This prototype is a non-limiting example of one of the many ways that this invention may be implemented.

[0035] In this prototype, the system is deployed to help control climate in part of the third floor of an office building. Specifically, it is deployed in four offices and one common space, encompassing the workspace of ten people. Two operable windows exist in this space, and they are equipped with window control nodes and motorized openers. The flow of chilled air through HVAC ducts for that space is regulated by seven variable-air-volume (VAV) dampers (also known as VAV boxes or VAV terminal units) which were retrofit with damper control nodes and motors. This prototype was deployed during the summer only, so only chilled air was provided. (If this prototype were deployed in the winter instead, then heated air would be provided)

[0036] In this prototype, each human user wears a portable sensor node (a "portable node"). Each portable node is wearable, weighs 30 g, and its housing measures 54 mm by 44 mm by 14 mm. Thus, advantageously, this sensor node is sufficiently lightweight and small to remain almost unnoticed by its user. Also, this sensor node has low average power consumption (11.5 μ A). Advantageously, this avoids the need for frequent battery recharges or large battery size, which would be a nuisance to the user. (Alternately, the circuitry in the

portable node can be made much smaller, for example, by using a different layout, different component choice, or more integration ASICs. In that case, the housing can be much smaller as well. Also, alternately, even lower power consumption can be achieved by, for example, using a different clock or processor.)

[0037] In this prototype, each portable node measures the local temperature, humidity, light level, and inertial activity level of the user. It sends these data wirelessly, at one minute intervals, to a central network hub. The portable node also has three buttons on the side which allow the user to input current comfort state (one button each for hot, cold, and neutral). These buttons are held down for at least two seconds to guarantee a successful transmission of data. This delay eliminates false transmissions due to jostling or bumping of the device.

[0038] FIG. 1 shows a portable node 101 attached to a wrist strap 103 and worn around the wrist, in this prototype. The portable node 101 includes three buttons 105 that allow a user to input the user's current comfort state (one button each for hot, cold and neutral). It also includes sensors 107, including sensors to detect temperature, humidity, light level and inertial activity.

[0039] In this prototype, the housing for the portable node is taken from another wearable electronic device, a keychain picture viewer. The portable node may come with a clip, and can be attached to clothing, keys, or merely placed in a pocket.

[0040] Alternately, as shown in FIG. 2, a portable node 201 may be attached to a lanyard 203 and worn about the neck, in this prototype.

[0041] In this prototype, other sensor nodes are mounted on the wall under a thermostat ("thermostat nodes"). If the room does not have a thermostat, the thermostat node is mounted on the wall, away from areas where humans would come in contact with it, but still in the general vicinity to gather proximate data. The thermostat nodes measure temperature and humidity.

[0042] In this prototype, there are also two sensor nodes mounted on the exterior of the building (one east-facing, one west-facing) to gather external climate data ("outdoor nodes"). The outdoor nodes measure temperature, humidity and light level.

[0043] In this prototype, the housing of the thermostat nodes and outdoor nodes is the same as the housing of the portable node (described above). However, the outdoor nodes are modified to be protected from excessive moisture. The penetration in the housing for the humidity sensor (in the outdoor module) is covered with a long tube, and the remainder of the case is sealed with silicone. The top of the outdoor node is covered with a light filter, to bring light levels down into the range of the light sensor.

[0044] In this prototype, the actuation of the various air sources (windows and air-conditioning dampers) is achieved via "window control" nodes and "damper control" nodes, respectively. Each of these control nodes is tethered to a 24 V_{DC} power source, and has a motor which can open or close the associated mechanical element via wireless commands. These control nodes also have sensors to monitor local wind speed, temperature, humidity and light level. These data, along with the current state of the motor, are sent off wirelessly at one minute intervals to the central network hub. The motor can also be actuated locally with a set of pushbuttons, allowing the user direct control of the air source.

[0045] In this prototype, “room” nodes are also employed. These room nodes receive the data from the portable sensor nodes, thermostat nodes, outdoor nodes, window control nodes and the damper control nodes, and act as network coordinators, ensuring that each proximate node is only talking to one device. Since each room has at least one room node, the locations of the portable nodes can be inferred from the received signal strength indicator (RSSI) of the RF device. The room nodes also include sensors to assess the local temperature, humidity, light level, and activity level and send these data to the central network hub at thirty second intervals. Communication to the central network hub is accomplished via an on-board Ethernet module. This Ethernet module not only routes all sensor data back to the central network hub, but also transfers motor control commands from the central network hub to the room boards, where they are sent off wirelessly to the control nodes.

[0046] In this prototype, the central network hub is a computer that receives all of the data over Ethernet and processes it according to the comfort and control algorithms. It checks for current activity in the network, and only responds to new data, resetting room nodes if they are no longer active. It also backs up current system state in case of failure, and timestamps and logs each data point for offline system analysis.

[0047] FIG. 3 is a diagram that shows the location of hardware nodes used in this prototype, including: room nodes, thermostat nodes, window control nodes, damper control nodes, and an outdoor node. FIG. 3 does not show the central network hub or the varying positions of the portable nodes that are worn by human users and that move as the users move.

[0048] FIG. 4 shows a high level view of the topology of the network employed in this prototype. In FIG. 4, “sensor nodes” means portable nodes, thermostat nodes and outdoor nodes, as the case may be. In FIG. 4, “control nodes” means window control nodes and damper control nodes, as the case may be.

[0049] In the network of this prototype, the main topology for the network can be described by analogy as a tree, with the central network hub acting as the trunk, room nodes acting as branches, and control and portable nodes acting as leaves. All data is sent to the central network hub before being processed and distributed back to the relevant leaf nodes. This organization was chosen for its ease of implementation, as a stable network routing table could be established once for the duration of the work. It also reduces network overhead, as nodes do not spend much time establishing connections. A single RF channel is used, as collisions are low enough not to warrant the extra power demand on the portable nodes of keeping aware of the network’s current channel. The RF unit, however, is capable of detecting open channels and switching frequencies.

[0050] In the network of this prototype, the physical (PHY) layer of the wireless transceivers uses a ZigBit® module (available from Atmel Corp., San Jose, Calif.). It comes pre-packaged with an Atmel® ATmega1281 microcontroller, Atmel® AT86RF230 2.4 GHz radio, and dipole chip antenna. It automatically implements aspects of the 802.15.4 medium access control (MAC) layer, including clear channel assessment, random exponential back-off, successful receipt acknowledge, and cyclic redundancy check (CRC) calculation and checking. Additionally, the radios do not respond to packets with a different personal area network (PAN) ID or

destination ID (except in the case of broadcast packets which do not require receipt acknowledge).

[0051] In the network of this prototype, each node is hard-coded with an ID when deployed, so the locations of room nodes and users of portable nodes can be tracked. Upon wakeup, leaf nodes request a branch node to communicate with. The first branch node to respond becomes the destination ID for that leaf node until the leaf node is no longer in contact. At this point, the leaf node will retry the transmission process and request another branch node from the network upon failure. A leaf node can only communicate with its allocated branch node. In this manner, the low power leaf nodes do not spend an appreciable amount of time finding optimal connections, but rather stay joined to their primary branch node for as long as possible.

[0052] In the network of this prototype, each RF packet that is sent contains a preamble, the ID of the transmitting node, the ID of the receiving node, the PAN ID, the packet type, the packet length, payload data, and a CRC. When a room node receives an RF packet, it strips the CRC and PAN ID, adds RSSI data, a header, and two footer bytes before sending it over Ethernet to the network hub. The network hub uses the header, packet length, and footer to determine valid packets, and uses the same format for transmissions to the room nodes. For example, the network hub can send motor control commands over the network.

[0053] In the network of this prototype, successful branch node acknowledgement packets are not sent over the network, but failed packets are sent over the network, so the system can be aware of whether a command was received or not. For example, if a motor command is sent, but the receiving node is down, that packet will get returned, and the system will be aware of the failure.

[0054] As used herein, “sensor nodes” means, collectively, portable nodes, wall nodes and outdoor nodes.

[0055] In the sensor nodes for this prototype, a data update rate of once per minute is used in order to minimize the RF unit power consumption time, but still allow for a responsive control system. Since the thermal and humidity time constants of the sensors are on the order of minutes, finer resolution data would not lead to substantial improvements. The sensor sampling procedures and RF transmissions account for 20 percent of the total power consumption, so slight increases in sampling frequency could be attained without exorbitant sacrifices in battery life.

[0056] In the sensor nodes for this prototype, the battery is a lightweight, rechargeable, 150 mAh, lithium polymer battery. This is the battery that came with the keychain picture viewers, and it has battery protection circuitry to ensure that the users can not be hurt by battery shorts. Although the battery is rechargeable, no recharging circuitry is placed on-board, as the expected battery life is two years.

[0057] The extended battery life is achieved via extremely low power sensing elements, and minimization of processor and RF unit on-times. All of these items are powered by a 2.7 V low quiescent current voltage regulator, the TI TPS780270200 (available from Texas Instruments Incorporated, Dallas, Tex.). At 500 nA, it places minimal load on the battery, but has poor transient response, with excursions up to 150 mV when high power sections such as the RF unit turn on. All voltage sensitive operations, such as ADC readings, are performed at times when such excursions do not happen. The digital operations remain unaffected.

[0058] In the sensor nodes of this prototype, the 11.5 μA current during draw is dominated by the real-time-clock (RTC) module on the Atmel® ATmega1281 microcontroller unit. This utilizes a 32 kHz crystal to keep track of time and wake up the unit every minute. By using this built in module, the lowest sleep state available is “power-save” mode, which has a base current draw of 2.5 μA at 2.7 V V_{CC} and 25° C. The remaining 3.5 μA come from the power required to vibrate the crystal. (Alternately, external RTC modules that run under 1 μA can be used to both reduce the crystal drive signal and allow for “power-down” sleep mode, to reduce sleep current consumption, and increase the battery life.)

[0059] In the sensor nodes of this prototype, 2.2 μA are a result of computation, sensing, and RF transmissions. Since the microcontroller is being run at 1 MHz from its internal RC oscillator, the power consumption is fairly low (1 mA), and the processor sleeps for most of the sensing time, making it quite negligible in comparison to the 4 ms the RF unit is on for at 20 mA. The next biggest power consumer is the SHT15 temperature and humidity unit (available from Sensirion AG, Staefa, Switzerland), which requires 55 ms at 600 μA to accomplish its task. The ADC and comparator units operate at 700 μA for 12 ms and 10 ms, respectively. The light sensor is awake for a full second due to its long start up time, and its current consumption varies with light level. On average it draws only 5 μA , though, which is small in comparison to the other sources. This gives 133 μAs of current draw for each wakeup cycle. These cycles come once a minute, averaging to 2.2 μA .

[0060] In the sensor nodes of this prototype, the temperature, humidity, and (if applicable) light level sensors are all off-the-shelf components. The SHT15 temperature and humidity sensor is run in 8-bit humidity and 12-bit temperature mode to save power. Higher resolution would allow for tighter feedback control loops around the room temperature, but the device is not accurate much past 12 bits, and this still allows for less than 0.1° F. resolution. The light sensor is the ISL29102 (available from Intersil Corp., Milpitas, Calif.), which has an integrated human eye response filter and an analog compressor at the output, allowing for its reading to be taken by the ADC unit on the microcontroller. Without this compression, the six orders of magnitude of light levels could not be represented by the 12 bits of the ADC.

[0061] In the portable nodes of this prototype, the inertial activity sensor is custom designed. Traditionally, inertial activity level is measured with a MEMS accelerometer, but these tend to draw approximately 180 μA at their lowest. Considering that the activity sensor preferably runs continuously so as not to miss any relevant motion, this would increase the device’s current consumption by an order of magnitude—without even accounting for the processor time necessary to sample an ADC or send values over a digital bus. Luckily, this application does not require the full frequency range of MEMS accelerometers. Specifically, the 0 Hz attitude information is not needed to determine if the user is moving around a little or a lot. Instead, a vibration integrator is developed around a Murata® PKGS passive piezo shock sensor (available from Murata Manufacturing Co., Ltd., Kyoto, Japan). Usually employed in hard disk drive and air bag deployment sensors, these small devices are very sensitive in all directions (1:2:4 x:y:z sensitivity ratios), mitigating the need for multiple axis integration for some applications.

[0062] The piezo ceramic is biased with high value (200 Mohm) resistors to keep the low frequency response of the

sensor. Since the capacitance of the sensor is approximately 450 pF, this gives a lower cutoff of 3.5 Hz. The remainder of the gain stages pass 0.1 Hz to 10 Hz, with the second stage being a second order low pass filter to eliminate 60 Hz pickup. An active peak detector and integrator follow to give an output which is dependent on the total activity over the past period. A reset function is implemented by the microcontroller to drain the charging capacitor between samples.

[0063] The op-amps used in this circuit (for the inertial activity sensor in this prototype) are the OPA2369 from Texas Instruments, which run at 700 nA per channel, and have very low crossover distortion. Since the passive components draw negligible power, this gives approximately 2.8 μA for the entire circuit. The low crossover distortion is helpful in reducing errors in the active peak detector. An active peak detector is required due to the low power supply rail of 2.7 V, causing losses due to a diode drop to become too large a portion of the full range.

[0064] To further keep power down, the integration charging currents are kept under 100 nA. This poses a difficult problem as reverse leakage in standard rectifiers is usually on this order or greater. To solve this issue, the low leakage FDLL300A from Fairchild is used. At 50 pA of reverse current, it exhibits low leakage for its relatively low forward voltage drop of 0.3 V.

[0065] In this prototype, the inertial activity sensor is extremely sensitive to slight movement. However, as it has a one minute integration time, these movements must be maintained for the output to show significant change. Each board is thoroughly cleaned with flux remover before commissioning, to mitigate the problem of small leakage paths discharging the integration capacitor differently. Other factors such as light level and temperature contribute to variations over time. The rectifier is light sensitive, having greater reverse leakage under higher light levels, and the charging capacitor and piezo element are temperature sensitive. Fortunately the temperature and lighting conditions remain fairly consistent in an indoor environment, and these are both measured on-board, and can therefore be compensated for in firmware. (Alternately, light effects can be reduced by placing a metal shield over the circuit, reducing electromagnetic interference as well).

[0066] As used herein, “control nodes” means, collectively, window control nodes and damper control nodes.

[0067] Since this prototype is adapted to allow for retrofitting of existing buildings, it is desirable that the boards for the control nodes accommodate many different actuators. To accomplish this, the LMD18200 PWM motor controller (available from National Semiconductor Corporation, Santa Clara, Calif.) is used. It can handle up to 3A of continuous current, and will operate 12V to 60V motors. As 24 V_{DC} is a common power supply in the control industry, this is deemed adequate for the situation. 24 V_{AC} supply is the most common in the control industry, but AC voltage does not allow for easy reversibility of motors, so DC is used. There is also an over-current comparator attached to the LMD18200, so the microcontroller can automatically shut off the unit if a motor has hit the end of its travel.

[0068] In this prototype, the motors used in the control nodes depend upon the application (window or damper). For operating the windows, a 24 V_{DC} motor is used. It comes with the appropriate mating connectors. The full window opening time is two minutes. The VAV box dampers are controlled with Belimo CM24-3-T actuators (available from Belimo

Americas, Danbury, Conn.). They have automatic over-current shutoff, and variable hard stops to set the end of travel. They also have a magnetic clutch, which allows them to be disengaged quite easily. This feature allows for quick switching between the normal control system and the experimental control system. The full damper opening time is 30 s. The damper control node is mounted near the motors scenarios to capture accurate wind speed, temperature, and humidity data.

[0069] In this prototype, the control nodes have the same light, temperature, and humidity sensing as the portable nodes. In comparison to the portable nodes, the control nodes are kept on continuously as they have a direct power source, eliminating the need to maintain any sort of battery life. The nodes also use the same ZigBit® module for their communications and sensor sampling tasks. An external header is supplied which allows for a manual pushbutton control board to be attached, and also provides for future expansions of functionality.

[0070] In this prototype, the control node also has a wind speed sensor that uses a spinning impeller design. The impellers are replacement parts for the Kestrel® 1000 wind speed meter (by Nielsen-Kellerman, Boothwyn, Pa.). They have a magnet on the shaft which is picked up by a hall effect sensor and sent through a filter and Schmitt trigger comparator to produce a square wave of the same frequency as the impeller's rotation. The impellers themselves are fairly linear, although they have a finite amount of friction, and will not sense below a certain level. These wind speed sensors are used to measure the chilled air flowing from the VAV boxes.

[0071] In this prototype, although the wind speed sensors themselves work well and have good linearity, they make a rattling noise at high wind speeds which is not tolerable to the building occupants. For this reason, in this prototype, a series of sheet metal shims are used to limit the air flow from the VAV boxes to the wind speed sensor to a level where the sensor no longer rattles. This reduces the total range of the sensor, making the initial dead band problem worse as it now occupies a greater percentage of the range. Fortunately the VAV dampers spend the majority of their time in a high flow rate regime, so the data obtained from these sensors are still meaningful. (Alternately, wind speed may be measured by the pressure drop across a Bernoulli plate, or by a hot-wire anemometer, or by a simple IR beam-breaking wind vane.)

[0072] In this prototype, the room nodes collect and distribute the RF packets, and keep track of the location of portable nodes in the network. They use the ZigBit® module and the Lantronix® Xport® Direct™ (Lantronix, Inc., Irvine, Calif.) to accomplish these goals. The XPort® is configured in UDP mode for ease of implementation.

[0073] In this prototype, the room nodes use a Panasonic® AMN31112 PIR motion sensor with a 5 m sensing radius. The output is digital, with a level change for every detected motion. These transitions are counted by the microcontroller over a 30 s sampling period, giving the average activity for that period. These data, along with temperature, humidity, and light levels, are sent out every 30 s over the Ethernet port. The temperature, humidity, and light sensors are identical to the ones used on the portable nodes, and they are kept on continuously as there is a wired power source.

[0074] The only difference in its sensing is that the SHT15 temperature sensor is mounted at an angle to the board. This is done to help limit the thermal effects from the other warm components on the board. The XPort is the main current sink on the board, running at 200 mA when active, and 100 mA

idle. since the board runs from 5 V, this gives 0.5 W to 1 W of power dissipation. The extra heat produced raises the sensor temperature significantly (~10° F.) and makes it sensitive to local air flow, as convective cooling now plays a role since the sensor temperature is no longer at the ambient temperature. The angle mounting improves this situation.

[0075] FIG. 5 shows examples of types of hardware used in this prototype, including a portable node 501, wall node 503, outdoor node 505, window control node 507, damper control node 509, room node 511 and central hub 513. Of course, in actual deployment, more than one of a type may be positioned in a single room, and all of these types may not necessarily be located in a single room.

[0076] The Control Node Firmware, Portable Node Firmware, Window Node Firmware, Damper Node Firmware and Room Node Firmware (set forth in the ASCII text files mentioned above) comprise source code employed in this prototype in the control node, portable node, window node, damper node and room node, respectively. This source code is one way, out of many, that firmware for this invention may be implemented.

[0077] In this prototype, there are multiple sensor inputs that may control any number of output devices, at any one time. This mapping necessarily changes with time as people physically move throughout the network, altering the locations of their cooling needs (or, this prototype were deployed in the winter, their heating needs).

[0078] To effectively handle the complex mapping requirements of this MIMO (Multiple Input Multiple Output) network, a hybridized control system is employed in this prototype. In hybrid systems, individual modules exchange information and trade off responsibilities in an ad hoc but hierarchical fashion. This fits particularly well with the topology of sensor networks, as the control layer matches the physical instantiation.

[0079] (Alternately, the entire control scheme could run in the network on the individual nodes. In addition to reducing wiring costs and the single point of failure that a central server represents, this would also increase battery life in the portable nodes by eliminating the need to transmit data every minute for processing. Advantageously, this could allow a more secure and private scenario for personal data contained within the portable nodes.)

[0080] FIG. 6 shows a high level system model of the control software system used in this prototype. The control system consists of a Control Module, Location Module, Window Module, Outdoor Module, Thermostat Module, Portable Module, and Room Module. Each node in the network has a module associated with it, to keep track of its own local state and needs. The Control Modules receive setpoint commands from the Window, Thermostat, Room, and Outdoor Modules, and make decisions concerning how to appropriately control the associated damper and window motors to reach these setpoints. The Location Modules aggregate all the wireless transmissions from the portable nodes, and create a location table that the Room Modules can access to find out who is where. The Window Modules receive information from the window control nodes, and keep a log of when the window was last manually operated. This is accessed by the Outdoor Modules when determining whether to open or close the windows. The Window Modules also monitor the air speed coming in through the window, and close the windows if it becomes too windy. The Outdoor Modules receive outdoor temperature and humidity information, and poll the indoor

Thermostat Modules to decide whether to open the windows and let in cool outside air. The Thermostat Modules track individual room air temperatures, and poll the Room Modules to determine whether or not they should be performing control actions. Based upon the Room Modules' states, the Thermostat Modules will either run normal control, setback control, or no control at all. For summer operation, the Thermostat Modules also determine if the room is too cold, at which point the Window Modules will not open the windows. The Portable Modules keep track of each individual user, and whether they are active and comfortable. The Room Modules poll the Portable Modules and Location Modules to find out if users are currently active, where they are located, and if they are comfortable. Based upon these findings, they either relinquish control to the Thermostat Modules, or work to minimize the discomfort in the rooms.

[0081] In this prototype, the control system also incorporates a back-up function, which is not related to the hybrid controller, but is critical to its proper functioning. After a short period of running, the control system establishes a complicated map of the state of the sensor network. This map continues to update for the entire duration of its operation. Some states evolve on a minute-by-minute basis, and others over hours or days. To maintain this state during system crashes, critical components are backed up to the hard drive whenever they are changed. Upon restart, these components are loaded before the program begins to run. Not all components are backed up, as some are modified too often, or are not important enough, to warrant the overhead of continual file writes. However, all comfort preference variables and expected occupancy variables are backed up.

[0082] FIG. 7 shows a high level flowchart for the Control Module, in the control system for this prototype. The Control Module represents one of the few hierarchical elements in the network, as it merely processes and passes incoming commands. An important purpose of the Control Module is to map the desired temperature control signals to the correct output devices, and maintain state between the various control hand-offs that occur throughout the day, as many devices control the same actuator. In this way, local control of the damper is maintained in a stable fashion, regardless of the incoming control signals. This also allows for a central repository for storing control variables specific to certain rooms.

[0083] The Control Module implements a hybrid proportional and integral (PI) dead-band controller. The PI controller is chosen for its simplicity and stability, limiting the number of unknown variables when analyzing the full control loop. Sufficient performance is obtained from the PI controller, such that a full proportional, integral, and derivative (PID) controller is not necessary. A predictive setback algorithm further reduces the need for the fast system response time of derivative control, as the room temperature is set before it is occupied. A dead-band is placed on the issuing of control signals around the setpoint in order to reduce network traffic. (The deadband is needed because commands are sent over the wireless network, and the incoming control signal is digital with a finite resolution. In the absence of a dead-band a control signal would be sent out every minute, increasing the probability of collisions). This dead-band is chosen to be ± 2 LSB ($\pm 0.1^\circ$ F.) as a good balance between maintaining temperature accuracy and limiting excessive transmissions. In practice, this keeps temperature excursions between ± 0.2 F.

[0084] In the Control Module, the PI gains were set via a combination of manual tuning and the Ziegler-Nichols

method [J. G. Ziegler and N. E. Nichols. Optimum settings for automatic controllers. ASME Transactions, 64:759-768, 1942]. The Ziegler-Nichols method involves increasing the feedback gain until resonance is reached, and then setting the gain parameters as a function of the critical resonance gain (K_c) and period of the critical resonance (τ_c). The proportional gain (K_p) is set to $0.5 \times K_c$, and the integral gain (K_i) is set to $1.2 \times K_p / \tau_c$. In a deployment of this prototype, the Ziegler-Nichols method was not employed exclusively, as long evaluation times were required to check for stability or resonance. After two days of testing, it was determined that the response was well enough understood to make a reasonable estimate of the optimal PI gains. The gain was increased by factors of two until instability is observed. The resonance was measured as having a period of approximately 0.2 hours at a positional gain of 32,000. This gave

$$K_p = 0.5 \times 32,000 = 16,000 \text{ and } K_i = \frac{1.2 \times 16,000}{0.2 \times 3,600s} = 26.7$$

Application of these control gains still produced excessive overshoot and oscillation, so the gains were halved again, assuming that the actual critical resonance could be anywhere between the 16,000 gain test and the 32,000 gain test. This produced gains of $K_p=8,000$ and $K_i=13.3$, which improved performance greatly. These gains were then manually adjusted to their final gains of $K_p=8,000$ and $K_i=10$ after more testing. Since all of the offices are of similar size, and have similar cooling equipment, the same gains are used for all offices. The only exception to this is a double office, which has two VAV boxes that are operated in parallel at one half the controller gains.

[0085] FIG. 8 shows a high level flowchart for the Location Module, in the control system for this prototype. The Location Module is in charge of keeping track of where each node is located in the system. It does this by aggregating all of the location packets, which are received by each room node and passed along via Ethernet to the network hub. Since packets arrive in tight succession, it is important to window out those packets which do not belong to a particular broadcast transmission. To do this, the Location Module starts a timer when the first location packet arrives from a particular portable node. It then only accepts packets for the next 50 ms. Since location is based upon the strongest RSSI of these received packets, the actual location may not be reflected by the data, as multi-path can easily confound RSSI data. To accommodate this, the Location Module performs a smoothing algorithm on the incoming data, taking a majority vote on the past three location results. If no majority can be reached, it takes the current result. This allows for temporary excursions to be excluded, but relatively quick response to changes in location.

[0086] FIG. 9 shows a high level flowchart for the Window Module, in the control system for this prototype. The Window Module receives data from the two window controllers in the network. These window controllers monitor the light levels, temperature, humidity, wind speed, and motor state.

[0087] The Window Module takes into account the motor state and wind speed. Via the motor state information, the Window Module can be notified if a user has pressed a button to manually open or close the window, if the window was last driven in the open or closed direction (either by a user or some other module in the network), or if the window was driven to

its limits in either the open or closed direction. From these data, the Window Module creates a repository that can be accessed by other modules, letting them know if the window is fully shut or not, or if the window has been manually operated in the past three hours or not. The manual timeout of three hours is chosen to keep other modules in the network from over-riding user preferences, but still allowing the system to respond if a window is left open overnight.

[0088] The other function the Window Module performs is keeping the wind speed coming through the window below a certain level. If wind speed greater than this level is detected, the window is closed a small amount. This process repeats until the wind speed has dropped sufficiently. Again, this automated control of the window is not performed if the window was operated manually in the past three hours. Both the window control nodes and the damper control nodes have built in functions for driving their respective motors to maintain a particular flow rate through the wind speed sensor, but these are not used at this time to allow for increased predictability of the complete system.

[0089] FIG. 10 shows a high level flowchart for the Outdoor Module, in the control system of this prototype. The Outdoor Module is driven by the two outdoor environmental sensors. The outdoor temperature and humidity are stored, and the outdoor air enthalpy is calculated. The enthalpy of a gas is a measure of its total stored energy at a given pressure and temperature. Since the contributions due to pressure variations are negligible in this case, they are ignored, and the enthalpy can be calculated based upon the temperature and moisture content of the air. To minimize computational load, enthalpy is approximated as

$$H=(0.007468 \times T^2 - 0.4344 \times T + 11.1769) \times RH + 0.2372 \times T + 0.1230 \quad \text{Equation 1}$$

where H is the enthalpy in Btu/lb, T is the temperature in degrees Fahrenheit, and RH is the relative humidity in decimal form. This approximation gives one percent accuracy over the range of conditions experienced by the sensors, which is more than adequate considering it is only the difference in enthalpy that is of interest, not absolute enthalpy.

[0090] If the outside air has less energy, i.e. lower enthalpy, it is useful in cooling down a room. For this reason, the Outdoor Module will open the window if the outside enthalpy is one Btu/lb less than the indoor enthalpy. Although the main air-conditioning system has an economizer which brings in outside air under similar conditions, air brought in through the window does not require a fan to move it, thereby reducing the net energy to cool a room. The window closes if the outdoor and indoor enthalpies are equal, effectively putting hysteresis in the system and eliminating excessive window motion. The indoor enthalpy is queried by the Outdoor Module from the room Thermostat Module associated with the window, along with an indicator of room temperature. If the room is more than a degree below its setpoint, it is assumed that the VAV dampers are already closed, and that additional cooling is not necessary. All of these functions are performed after consulting the Window Module to ensure that the window isn't already open or closed, and that it hasn't been manually operated recently.

[0091] FIG. 11 shows a high level flowchart for the Thermostat Module, in the control system of this prototype. The Thermostat Module performs many of the room temperature control functions. The Thermostat Module receives temperature and humidity information from sensors mounted in each room below the pre-existing thermostats. It calculates

enthalpy according to Equation 1, and determines if the room is too hot or too cold. Before making these assessments, it first consults the associated Room Module to check its state. If the room is occupied, but not by a person wearing a portable node, the Thermostat Module performs normal control, regulating the room temperature to a fixed setpoint, usually an average of the documented comfortable values of its occupants. If the room is unoccupied, it performs setback control, regulating the room to a fixed temperature usually six degrees Fahrenheit higher than Normal Mode. This setback temperature is limited to six degrees in order to allow for fast transitions back to comfortable temperatures when users arrive unexpectedly. In cases where the room is occupied by users with portable nodes, the Thermostat Module relinquishes control of the VAV dampers to the Room Module, which has more information as to who is present and what his or her comfort needs are.

[0092] The Thermostat Module has a number of variables that are accessible by other parts of the network. The enthalpy, temperature, and humidity are used widely throughout the system. The Thermostat Module also has a series of lower temperature limits it checks for, and sets a "too cold" indicator if any of them occur. (This describes the case in the summer. In the winter, the Thermostat Module has a series of higher temperature limits it checks for, and sets a "too warm" indicator if any of them occur.) This function is necessary as there may be many different control operations happening at any one time, some of which might be conflicting. For example, both the VAV dampers and window motors can be operated to cool the room down, and they act independently, so a hard-stop is required to keep the system from driving itself too far in one direction. Depending upon the current room occupancy condition, different lower limits are used. The lowest limit is during setback mode, when occupancy comfort is not relevant and reducing energy consumption is most important. In these conditions, temperatures are allowed to drift down to 65° F., helping to store energy in the thermal mass of the internal building. If the room is occupied, the lower limit is set to one degree Fahrenheit lower than the control temperature. In cases where the room is occupied by users wearing portable nodes, this lower limit is kept at 70° F. This is because the desired temperature is unknown by the Thermostat Module, and it is best to take the conservative approach of not cooling the room too much, but instead allowing the main system to control the temperature.

[0093] FIG. 12 shows a high level flowchart for the Portable Module, in the control system of this prototype. The Portable Module keeps track of each user in the system, and determines whether the user is active and comfortable. Activity determination is necessary, as the nodes are often left on the users' desks when they leave for the day. The activity determination is based upon activity and temperature information from the portable node. Data from the portable node light sensor was considered as a possible indicator of activity, but the variations in light levels due to sunlight, and the lack of predictability as to whether the unit is being worn under clothes, made it an inconsistent activity detection variable. Temperature is only mildly influential in the activity algorithm, as it is the main control parameter for the system. If it were to trigger a false positive, and then drive the air-conditioning system to either reinforce its state, or attempt to change its state when this is not possible, an unstable feedback loop could result. An example of such a situation would be a user leaving a portable node next to a computer which is

warm. This would heat up the portable node and could indicate activity, causing the control system to believe a user is in the room. It would then try to cool down the node, which would be impossible given the sensor's proximity to a heat source. To eliminate such scenarios, the temperature alone does not indicate activity.

[0094] In the Portable Module, the main determinants of activity are windowed mean and variance of the piezoelectric motion sensor on the portable node. The activity algorithm quickly detects when a user is no longer wearing a sensor. (This is preferable, since the sensor temperature quickly drops after removal. If the system failed to quickly detect that the user had ceased to wear a sensor, the system would react to the temperature drop by shutting off the air-conditioning, to the displeasure of the remaining users in the room.) To achieve this quick detection, a window time of three samples is chosen, giving a maximum delay of three minutes to clear the buffer. As the average delay of a windowed sample is one half the window time, the average delay for this system is only 1.5 minutes. This also has the advantage of giving a fast start-up time, so users are added to the system as soon as they arrive. The only disadvantage is the increase in false positives, which, fortunately, are short lived due to the small window time.

[0095] In the Portable Module, the incoming data from the portable nodes are first separated by time since last arrival, as packets are sometimes sent twice due to the acknowledge transmission failing even though the data packet transmission was successful. Only packets which arrive more than 55 s since the last packet are accepted. Furthermore, for activity recognition, any button press packets are ignored, as these are out of sequence time-wise and result in incorrect activity data. Since the activity data is an accumulation since the last transmission, and button presses can occur at any time, the actual activity is not representative. Although the time since last packet arrival is known, and the actual activity level could be backed out, button presses occur so infrequently that this added complexity is unnecessary. Also, the high forces during the actual button press action make these data of little value.

[0096] After this windowing process, the Portable Module checks for activity start or continue conditions. The activity start conditions have much higher values than the activity continue conditions, to help reduce false positives under the assumption that once activity is detected, it is more probable that the user is still active rather than not. This keeps users in the system through periods of low activity, when the activity sensor is returning low values. Start conditions include windowed mean and variance above certain thresholds, and continue conditions include windowed mean and variance and temperature above certain thresholds. There is also a timeout on the continue conditions, excluding temperature. Temperature is excluded to eliminate thermal lock-up conditions. If no continue conditions are detected before the timeout period of six minutes, the system assumes the user is no longer active, and requires that a start condition be detected before re-establishing him or her in the system. The user is only flagged as active if a continue condition is reached, even if the timeout has not yet expired. The time of last activity is then logged for other modules to access in their decision processes. A time stamp is chosen as the pass variable instead of an activity flag, as a user could leave the system while active and leave the activity flag set, which could not be reset as no new user data would be arriving.

[0097] In an illustrative deployment of the Portable Module of this prototype, many very low activity states occur during normal usage by the users in the system. Many of the tasks the users perform on a daily basis include reading and thinking, neither of which require much physical motion. As each portable node, and the user of the node, is slightly different, it is preferable that the thresholds of activity were different as well. These thresholds are picked manually for the users from visual evaluation of six weeks of user data. As the data is unlabeled, estimations are made of the actual active times, and the algorithms were modified to match these estimates as best as possible. After the activity is sensed, a timeout is placed on any use of the data for fifteen minutes. This is done to help alleviate erroneous control signals due to the temperature and humidity sensors not having acclimated. It would be more stable for the system to wait even longer, but this would also reduce the responsiveness of the system. If a user is not active for more than 20 minutes, the algorithm is restarted and the user must wait another 15 minutes for his or her data to become valid again. This 20 minute window time allows for short excursions out of the system and compensates for short term false negatives in the activity algorithm.

[0098] If a button is pressed, these data are processed by the comfort algorithm, which stores the last nine button presses each of hot, cold, and neutral. For the same reasons employed by the activity algorithm, the comfort algorithm only allows button presses from users who have been active for at least 15 minutes. The comfort algorithm is updated and stored in a back-up file. As each subsequent data packet arrives, a comfort distance metric is calculated based upon this updated algorithm, and made available to the Room Module for temperature control calculations.

[0099] FIG. 13 shows a high level flowchart for the Room Module, in the control system of this prototype. The Room Module is one of the most influential modules in the network, even though it only has one sensor data stream of its own that it uses: the passive infrared (PIR) motion sensor. In addition to its own sensor, the Room Module references many other modules in the network before making its decisions. It determines what format of control the room should be set to (normal, setback, or comfort), and performs the comfort control itself.

[0100] The reason for this is that the room nodes are fixed in location, and always active, so they can be relied upon to maintain the system state. The room nodes also have 30 s data updates, making them more responsive than the rest of the nodes in the network, which have at least 60 s updates.

[0101] The Room Module first processes its activity information. Although the PIR sensor returns a count which increases with the amount of motion in the room, all levels greater than zero are counted as a level of one. This is done to even out the algorithm results, as only the presence of activity is desired to be known, not the level of activity. The algorithm performs two windowed means on the incoming data, one over the past 70 samples, and the other over the past 12 samples. Two different averaging times are used to drive two different outputs. The first is a determinant of whether the room is occupied, which requires a fast response time. The second is a determinant of the first arrival time of users in a day, which requires high accuracy but response time is irrelevant. Both algorithms use a higher threshold for initiating an occupancy state than they use for maintaining an occupancy state. This limits false positives and excessive oscillation between states throughout the day.

[0102] In the Room Module, the main role of the occupancy algorithm is to determine what level of cooling the room needs. For this reason, a large amount of lag is placed in the transitions, in order to keep the air-conditioning system from cycling too heavily and wasting energy. Once occupancy is detected, the system maintains an occupied state for three hours. This was done to bridge large gaps in the day when users leave to go to lunch or meetings. These happen quite frequently, and a three hour window covers most scenarios. This does, however, place a high penalty on false positives, as it will cool a room excessively for three hours. The window averaging time of 12 samples, which equates to six minutes, attempts to strike a balance between fast activation and accuracy. This gives an average delay of three minutes for new occupant entries. Although there is a long time window for this global occupancy variable, the system also keeps track of immediate occupancy for use in determining what cooling actions are required.

[0103] If no occupancy is detected, the system checks if a user is expected to arrive soon, in order to ensure that the room is at an appropriate temperature when he or she enters. This is done by checking the arrival times stored in a buffer over the past week. If a user arrived any time in the past week within two hours of the current time, the system assumes he or she will most likely arrive again, and it prepares the room accordingly. The window time is much larger for this algorithm, with an average lag of 17.5 minutes, to ensure that any person who entered the room in the past week actually stayed long enough to make it worth cooling the room down for him or her. This algorithm also employs a three hour time-out, in order to bridge gaps in the day, and return only one entry point for the day. If a room is unoccupied for more than three hours, the next entry will be stored in a buffer for future reference.

[0104] Once the Room Module has calculated its occupancy state, it then performs control actions. These control actions are performed at one minute intervals, even though data is updated at 30 s intervals. This is done to synchronize with the other controllers which are limited to one minute updates. Doubling the control action time would be the equivalent of doubling all the PI control gains, causing system instability. If no occupancy is detected, and it is not within two hours of an expected entry, the Room Module sets the control state to setback and performs no actions. If occupancy is detected, or expected to occur in the next two hours, but there are no users wearing portable nodes in the room, the Room Module sets the state to normal and performs no control actions. Under both setback and normal states, the Thermostat Module controls the VAV dampers to regulate the room temperature. If the room has been occupied by a user with a portable node in the past three minutes, the Room Module checks the comfort level of all users present in the room and averages them to create a control scalar. It then turns off both normal and setback control, and acts upon the control scalar to minimize the average room discomfort. Once users have been detected there is a six minute timeout before either setback or normal mode can be entered, in order to maintain system state during periods of false location responses and short trips out of the room.

[0105] When determining the control scalar for a room, the Room Module only considers those occupants who are normally situated in the environment. The test setting incorporates four closed offices, and one large common space which is divided into two sections. For the four offices, only the occupants of those offices are allowed to control the comfort

setting. In the large common space, all users are averaged together when determining the setpoint. Although room occupant information was pre-assigned in this case, it could be replaced by a weighting algorithm that determines how much time each user spends in a particular space. This would produce similar results as the pre-assigned knowledge, as office occupants would make up the highest percentage for their respective offices, but it would also increase the amount of control workers in the public space had over their area.

[0106] In the control system of this prototype, indications of discomfort inputted by users during an initial transition period (15-30 minutes) after the users enter a room are ignored (on the expectation that this discomfort will pass after the system responds to their presence). To mitigate the problem of initial discomfort, the system used a conservative setback control technique, which aimed to ensure that rooms were always within the realm of comfort before the user entered.

[0107] In this prototype, the control platform was stable, with room occupancy detectors ensuring that the temperature was at a reasonable level, regardless of whether the occupant had a portable node. The window actuators had hardware over-ride buttons that allowed occupants to open and close them when desired.

[0108] In this prototype, a comfort algorithm measures the user's "comfort distance". The term "comfort distance" means how far away the user is from being comfortable—specifically, how far away the user is from being at a comfortable temperature. The comfort algorithm not only calculates values of the comfort metric, but also drives a control loop.

[0109] In this prototype, it is preferable that the comfort algorithm satisfy a number of constraints. Preferably, it has a monotonic structure to avoid instabilities and local minima. These could be accounted for through a non-linear control structure, but this is avoided in this prototype in order to minimize the number of variables involved in determining system stability and performance. Preferably, the algorithm may be used with a limited set of on-body comfort indices. In this prototype, only temperature and humidity are measured on-body. This problem is compounded by the limited labeling of the acquired data. Users of this prototype have only three choices to indicate comfort state: hot, cold, or neutral. This means that, in this prototype, there is no way of knowing exactly how hot or cold the users are at the instant a button is pressed. Instead, in this prototype, this metric must be inferred from the distribution of the received data.

[0110] In this prototype, not only are the labeled data points ambiguous as to their level of discomfort, but they are also very sparse in their occurrence. The users are not required to press any buttons, and are only asked to do so if they feel uncomfortable, limiting the amount of labeled data points to an average of about one per person per day. Ideally, if the system were to function flawlessly, this number would be even lower, as the users would be comfortable more often. Another issue faced in this reduced data set is the lack of an even distribution of hot and cold labels. For some users, the room never went cold enough to make them feel cold, so only hot data points exist, giving no information by which to determine a lower limit on comfort.

[0111] In this prototype, the comfort algorithm preferably is robust to insufficient, inaccurate, and indeterminate data. For example, if one user has his or her sensor in a pocket, where the temperature is much warmer than its usual location,

the entire system preferably is not allowed to drift too cold in order to compensate. Preferably, some pre-assigned knowledge is included which incorporates a rational view of comfort boundaries. This favors more general approaches, which may not be as effective for each individual, but will reduce the problems associated with over-fitting of the data (e.g., the system attempting to cool a room to match an accidental button press). Also, preferably, the comfort algorithm runs and updates in real-time, to affect the comfort of the users when it is needed.

[0112] In this prototype, many features are available to help determine comfort. Examples of these include on-body environmental conditions, room environmental conditions, outdoor environmental conditions, location, time of day, day of week, and local VAV air flow. Even without the creation of any subfeatures, the system in this prototype logs 24 different data indices each minute, for each user in the system. Given the extremely high dimensionality of the data set in comparison to the number of labeled data points (an average of one per person per day), the number of indices used to determine comfort is preferably kept to a minimum. Any algorithm trained on a small data set, with too many features, will classify those particular points well, but is unlikely to be representative of the function as a whole. However, preferably the training should be much larger than the feature set. Preferably, this limits the scope for these data to two or three features.

[0113] In the comfort algorithm used in this prototype, the only sensor data used to determine comfort are the portable node's temperature and humidity data.

[0114] (It must be stressed that this prototype, and this invention more generally, can be implemented using other or additional parameters to determine comfort. For example, temperature and humidity measured in room nodes and thermostat nodes can be used advantageously as parameters to help determine user comfort. Indeed, in a test deployment of this prototype, the main parameters that returned consistent results among all users, were the room and on-body environmental conditions.)

[0115] Two different comfort algorithms were used for this prototype: a K-Nearest-Neighbor ("KNN") algorithm was used in an early deployment of this prototype, and a Fisher Linear Discriminant ("Fisher Discriminant") was used in a later deployment of this prototype.

[0116] In an early version of this prototype, a KNN algorithm was used to determine comfort. KNN is well-suited to determine a level of discomfort, since distance measurement is an intrinsic aspect of locating nearest neighbors. In this KNN algorithm, "comfort distance" is measured as the distance away from a known labeled point, and an average of these comfort distances is used to help reduce the effects of outliers.

[0117] In this KNN algorithm, although it is possible to accurately measure the Cartesian distance between any two data points, it is unclear how this distance relates to comfort. For this reason, outside knowledge is employed to help train the algorithm. It is well known that increasing temperature and humidity levels lead to increases in heated thermal discomfort. To determine the appropriate weightings of temperature and humidity for this situation, the slopes of the 'comfort lines' are measured from the received data from the portable nodes. These comfort lines are the boundaries

between the hot and cold labeled data points, essentially enforcing a comfort metric based upon the orthogonal distance to this line of comfort.

[0118] This KNN algorithm has at least four advantages. First, it incorporates expert knowledge of the system, allowing the system to respond as expected, and constraining the output to apply negative feedback to the control system. It is preferable that the decreases in temperature always result in decreases in heated discomfort, as this is the only method by which the VAV damper motors can affect thermal sensation. Furthermore, this comfort line can be determined off-line by visual inspection, which can achieve closer fits due to a human's better understanding of the difference between representative data and outliers than the computer's. Second, the KNN algorithm can easily incorporate the neutral class into its determination, giving a better average of how the user might be feeling at a particular point. This also allows for morphing of the comfort space to match an individual's experience, as local clusters can pull away from the enforced distance metric, if reinforcement is high enough. Third, the KNN algorithm is robust to insufficient data. In fact, it has the ability to work in the complete absence of "cold" data points. (In a deployment of this prototype, some users never became cold enough to input an indication that they were cold.) Fourth, the KNN algorithm creates a flexible boundary, close to the labeled discomfort points.

[0119] In this KNN algorithm, to determine the comfort line, an average of all comfort boundaries is used. A linear boundary is assumed to limit the ambiguity of the weightings of temperature and humidity at a given point, by restricting it to a constant for the entire space. An average is also taken to make the system more representative of the preferences of the entire group. Individual comfort slopes could be used for each user, but these slopes are based on relatively sparse data, and an average tends to suffer less from the problem of over-fitting.

In this KNN algorithm:

$$D_d = \frac{1}{n} \sum_{i=1}^n \left(\frac{T_d - T_i}{G_T} - \frac{RH_d - RH_i}{G_{RH}} \right) \times G + C_i$$

where D_d is the level of thermal discomfort at point d, for n nearest neighbors, T is the temperature in degrees Fahrenheit, RH is the relative humidity in percent, and $G_T/G_{RH} = -3/5$ is the temperature to humidity ratio. C_i is the class label of the i^{th} neighbor, with -1 representing cold, 0 representing neutral, and +1 representing hot. The relative weighting of the enforced linear distance metric is set by G, with lower values of G allowing the KNN algorithm to adapt to local data more readily, at the cost of greater overall coherence.

[0120] In a later deployment of this prototype, the Fisher Linear Discriminant ("Fisher Discriminant") was used to determine comfort. The Fisher Discriminant is used to fit a linear boundary between labeled data and measure the distance from this boundary. The Fisher Discriminant creates a clearly defined boundary from which a positive or negative distance may be measured.

[0121] The Fisher Discriminant seeks the most effective rotation matrix for the given data set, to produce a projection on a lower dimensional space with high class separation. It takes a statistical approach of finding the greatest between-class scatter for the lowest within-class scatter.

[0122] In this later deployment of this prototype, the Fisher Discriminant reduces a two dimensional space to one (the two dimensions being the temperature and humidity measured by the portable node). The Fisher Discriminant also chooses a decision boundary. The decision is based upon discriminating between points which represent the boundary conditions. In this case, the most representative training points are assumed to be those with the most extreme values (e.g., “hot” data with the lowest temperature values), and a separating line is created at the mean of these data. The comfort distance can then be simply computed as the distance to this boundary.

[0123] In this implementation of the Fisher Discriminant, in order to accommodate the updating and adaptation of the comfort algorithm, a limited set of data points is used in creating the decision boundary. Nine points each of hot and cold are used, which allows a complete update in two to three weeks (users press buttons on average once a day). In cases where nine data points are not available, as many as are present are used. If less than two data points exist, two points are selected which create a reasonable line in comparison to other users. One point each generally returned more favorable results, but two are used in order to limit problems associated with outliers.

[0124] In this implementation of the Fisher Discriminant, the distance between the ‘hot’ and ‘cold’ labeled points varies greatly for the different users. Accordingly, the calculated comfort distance would (but for normalization) also vary greatly between users. To normalize this reported comfort distance so the control system can effectively arbitrate between users, the mean distance of “hot” and “cold” points from the decision boundary is calculated. As new temperature and humidity data are collected by a user’s portable node, the final comfort output is computed as the comfort distance divided by this mean distance. In this way, each user is equally uncomfortable for a given comfort value.

[0125] FIG. 14 is a high level flowchart for a comfort algorithm in this prototype, which employs a Fisher Discriminant.

[0126] The prototype described above is not the only way that this invention may be implemented.

[0127] In an alternate implementation, sensors are densely distributed throughout the rooms, so that a user has a sensor nearby, regardless of the user’s position. For example, dozens of temperature sensors may be positioned in each room. These temperature sensors may be solar-powered and have low power consumption. Advantageously, a dense distribution of temperature sensors can reduce the occurrence of false readings from nearby, warm electronic components could be reduced via outlier elimination and sensor data fusion. Also, advantageously, it can liberate the user from the need for cumbersome, on-body temperature sensing. Other types of sensors can also be densely distributed, such as sensors for detecting humidity or light level.

[0128] If a dense distribution of sensors is used, the sensor data can be related to the user based on the user’s position.

[0129] In some alternate implementations, sensors may be embedded in a cellphone, smart phone, other mobile device or wristwatch, allowing such a device to perform portable node functions, such as assessing occupant identity, location, activity level, and comfort status. Advantageously, this may allow the user to maintain at least some of this information on his or her mobile device, for privacy reasons. Also, advantageously, this may eliminate the need for the dedicated portable nodes (wrist-worn or lanyard mounted) used in the prototype described above. For example, a software app may be

installed on a conventional cellphone, smart phone or other mobile device to allow the device to perform portable node functions. In that case, data (such as data indicative of identity, comfort status, or sensor measurements) may be transmitted from that device. Such data may be used, for example, when calculating position values indicative of the position of the user of that device or when calculating comfort values.

[0130] In some implementations, users can input their preferences regarding a trade-off between comfort and energy efficiency. For example, a user may input his or her willingness to accept temporary discomfort in order to save on energy.

[0131] In some implementations, robustness to RF interference is improved (as compared to the prototype described above) by using a channel hopping protocol and better packet identification.

[0132] In some implementations, both the window motors and damper motors have hardware over-ride buttons that allow occupants to open and close them when desired (as compared to the prototype described above, where only the window motors have this feature.)

[0133] In the prototype described above, room usage schedules are automatically created by the system. In alternate implementations, room usage schedules are augmented with personal information from online calendars or location information from cell phones or other mobile devices.

[0134] In alternate implementations, user intent is communicated by means other than or in addition to button presses. For example, a user may communicate intent by manual interaction with a device (such as a rocker switch, dial, mouse or keyboard), by interacting with a graphical user interface on a computer screen, or by voice or other auditory commands. Alternately, for some implementations, the control system infers user intent from other actions customarily taken to modify the environment (such as closing or opening a shade, window or vent). The user intent being communicated may be an instruction, a comfort state or other data.

[0135] In some implementations, a user can input data that conveys the magnitude of the user’s discomfort. For example, magnitude may be indicated by the duration or pressure of the button press. (This is in contrast to the prototype described above, in which the user can input whether he or she feels hot, cold or neutral, but cannot input the magnitude of his or her discomfort).

[0136] In some implementations, the control system provides perceptible feedback to the users. For example, this feedback may indicate the system’s belief about the user’s current comfort, the action that is being taken, or that comfort is intentionally being compromised for energy reasons. Such feedback may, for example, be visual (e.g., given by changing the intensity of one or more light sources, which light sources may differ in color, or be given via a graphical user interface on a computer screen, or by email or other text message), auditory (such as by transducers that output beeps, or with an IVR system) or haptic. The control system may allow the user to input instructions regarding the type, timing and level of detail of the feedback. For example, a user who usually does not want feedback may want immediate, detailed feedback if he or she is hot and feels that the system is not responding.

[0137] In alternate implementations, other or additional types of hardware may be used. For example, other types of sensors may be used (such as accelerometers or inertial mea-

surement units that are embedded in portable nodes to detect motion of a user). Or, for example, other network topologies and connections may be used.

[0138] The control system software is not limited to the particular configuration used in the prototype. The modules may be implemented in different ways, and additional and different modules may be employed. Other firmware may be employed.

[0139] Other comfort algorithms may be employed in this invention, instead of the Fisher Discriminant and KNN algorithms. For example, a hybrid KNN/Fisher Discriminant algorithm may be employed that uses a Cartesian distance from the KNN determined decision boundary.

[0140] In some implementations, a larger data set is used for the comfort algorithm (rather than the nine or less most recent data points used in the prototype described above). Advantageously, use of a larger data set makes it much easier to detect outlier data, and permits use of a larger feature space. For example, a time-weighted metric, which leaves many more data points in the set, but gives preference to more recent events, may be used. Also, for example, the user may be granted inputs to communicate which scenarios are most relevant.

[0141] In some implementations, software can determine user comfort from sensor data and from user input, and can also do one or more of the following: (a) infer location of users in an area, (b) arbitrate between conflicting comfort demands within a given space, (c) make decisions regarding tradeoffs between comfort and energy, (d) keep users posted of current state of their comfort and energy usage, (e) balance demand between various locations within a building, (f) make predictive control decisions based upon past information, (g) locate malfunctioning areas of the control system, and (f) turn off the system when users are not present.

[0142] In some implementations, the system may contain a wide range of fixed position sensors to augment the effectiveness of the algorithms. For example, one or more of the following fixed position sensors may be employed: (a) outdoor temperature sensor, (b) outdoor humidity sensor, (c) outdoor air flow sensor, (d) outdoor rain fall sensor, (e) outdoor light sensor, (f) incident radiation sensors on windows, (g) motion sensors in rooms, (h) light sensors in rooms, (i) door open/closed sensors in rooms, (j) air flow sensors on the hot/cold air outlet ducts, (k) air flow sensors near windows, (l) temperature sensors at various locations indoors, (m) humidity sensors at various locations indoors, (n) power sensors on the hot/cold air actuators, (o) power sensors on the window actuators, (p) power sensors on the various HVAC components (e.g., compressors, fans), (q) power sensors on the lights, (r) light sensors on or adjacent to the lights, and (s) passive IR sensors in a PIR motion detector.

[0143] The climate-controlling output of this invention is not limited to moving HVAC dampers and opening and closing windows. For example, in some implementations, the control system can do one or more of the following: (1) control fans; (2) control motors to open or close window shades; (3) control light switches or light dimmers (to affect thermal load); (4) control power supplied by electrical outlets (for example, to turn on or off lamps or other energy sinks); or (5) adjust temperature setpoints for zones in an HVAC system.

[0144] In exemplary implementations, the portable node includes a temperature sensor and wireless transmitter. It may also include one or more of the following: (a) a humidity

sensor, (b) occupant activity sensor, (c) ambient light sensor, (d) unique ID, (e) location engine, (f) RFID tag (which may be active or passive), (g) battery, (h) user input devices (e.g., buttons, touch screens), and (i) output devices (e.g., LEDs or LCD screens).

[0145] In exemplary implementations of this invention, computer processing (including processing of sensor data, implementation of modules in a control software system, implementation of a comfort algorithm, and processing of signals being received or transmitted) is carried out by one or more computer processors. This invention is not limited to any particular computational network. For example, depending on the particular implementation of this invention, any one or more of the following may vary: (a) the number and type of processors, (b) their physical locations, (c) how they are interconnected, (d) whether computation is centralized at a central server, distributed, or a hybrid thereof; and (e) the specific computing tasks assigned to the various processors.

[0146] A few definitions and clarifications of terms. As used herein:

[0147] “Calculation”, “calculate” and grammatical variations thereof include obtaining data from a lookup table.

[0148] A “comfort value” is a value that indicates the comfort of a user, as perceived by that user.

[0149] An “extreme value” is the maximum or minimum value in a set of sensor measurements by a sensor, which set of sensor measurements are associated with a particular comfort state of a particular individual. For example, the coldest temperature value at which an individual reports being “hot” is an extreme value for that individual.

[0150] The terms “include”, “includes”, “including” and grammatical variations thereof shall be construed broadly, as if followed by the words “without limitation”.

[0151] The term “or” is an inclusive disjunctive. For example “A or B” is true if A is true, or B is true, or both A or B are true.

[0152] A “position value” is a value that indicates position. For example, a position value may indicate the position of a user or a sensor.

[0153] This invention may be implemented in many different ways. Here are some examples:

[0154] This invention may be implemented as a system that comprises, in combination: (A) input devices for accepting inputs from a plurality of humans, (B) sensors for taking sensor measurements, the sensor measurements including measurements of temperature, (C) at least one processor for processing data indicative of the inputs and the sensor measurements, calculating position values, calculating comfort values indicative of comfort of at least some of the humans, and generating control signals, based at least in part on at least some of the comfort values, and (D) at least one actuator for mechanically moving, in accordance with the control signals, one or more objects to alter air flow. Furthermore: (1) the sensor measurements may also include measurements of humidity; (2) at least some of the sensors may be positioned in or on portable nodes, at least one sensor per portable node, which portable nodes may be adapted to be carried or worn; (3) at least some of the sensors may be located in fixed positions, (4) at least some of the sensors in fixed positions may be located inside of a building and at least some of the sensors in fixed positions may be located outside of the building, (5) the position values may be calculated based, at least in part, on data indicative of at least some of the sensor measurements; (6) the sensor measurements may include mea-

measurements of received signal strength of wireless radio transmissions from the portable nodes, and the position values may be calculated based, at least in part, on data indicative of the measurements of received signal strength; (7) the sensor measurements may include measurements of inertial activity of the portable nodes, and the position values may be calculated based, at least in part, on data indicative of the measurements of inertial activity; (8) at least some of the human inputs may be values indicative of human sensory perception; (9) at least some of values may be indicative of perceived temperature; (10) the sensor measurements may include measurements of ambient light level; (11) at least some of the sensors may be passive infrared sensors in PIR motion detectors; (12) the at least one processor may be adapted to generate comfort values by using an algorithm that employs linear discriminant analysis or a Fisher linear discriminant; (13) the at least one processor may be adapted to calculate a decision boundary, to calculate comfort values indicative of distance from that decision boundary, and to update calculations of the decision boundary based on updated measurements; (14) the decision boundary may be calculated as the mean of at least some extreme values; (15) at least some of the position values may be indicative of the position of at least some of the humans, respectively; (16) at least some of the position values may be indicative of the position of at least some of the portable nodes, respectively; (17) the at least one processor may be adapted to take into account, when calculating position values, data transmitted wirelessly by a cellphone, smart phone or other mobile computing device; and (18) the at least one actuator may comprise multiple actuators, at least some of which are each adapted to alter air flow through a VAV box. [0155] This invention may be implemented as a method that comprises, in combination: (A) using input devices to accept inputs from a plurality of humans, (B) using sensors to take sensor measurements, the sensor measurements including measurements of temperature, (C) using at least one processor to process data indicative of the inputs and the sensor measurements, to generate position values indicative of the position of at least some of the humans, to generate comfort values indicative of comfort of at least some of the humans, and to generate control signals, based at least in part on at least some of the comfort values, and (D) using at least one actuator to mechanically move, in accordance with the control signals, one or more objects to alter air flow.

CONCLUSION

[0156] It is to be understood that the methods and apparatus which have been described above are merely illustrative applications of the principles of the invention. Numerous modifications may be made by those skilled in the art without departing from the scope of the invention. The scope of the invention is not to be limited except by the claims that follow.

What is claimed is:

1. A system that comprises, in combination:
 - input devices for accepting inputs from a plurality of humans,
 - sensors for taking sensor measurements, the sensor measurements including measurements of temperature,
 - at least one processor for:
 - processing data indicative of the inputs and the sensor measurements,
 - calculating position values,
 - calculating comfort values indicative of comfort of at least some of the humans, and

- generating control signals, based at least in part on at least some of the comfort values, and
 - at least one actuator for mechanically moving, in accordance with the control signals, one or more objects to alter air flow.
2. The system of claim 1, wherein the sensor measurements also include measurements of humidity.
 3. The system of claim 1, wherein at least some of the sensors are positioned in or on portable nodes, at least one sensor per portable node, which portable nodes are adapted to be carried or worn.
 4. The system of claim 3, wherein at least some of the sensors are located in fixed positions.
 5. The system of claim 4, wherein at least some of the sensors in fixed positions are located inside of a building and at least some of the sensors in fixed positions are located outside of the building.
 6. The system of claim 1, wherein the position values are calculated based, at least in part, on data indicative of at least some of the sensor measurements.
 7. The system of claim 3, wherein the sensor measurements include measurements of received signal strength of wireless radio transmissions from the portable nodes, and the position values are calculated based, at least in part, on data indicative of the measurements of received signal strength.
 8. The system of claim 3, wherein the sensor measurements include measurements of inertial activity of the portable nodes, and the position values are calculated based, at least in part, on data indicative of the measurements of inertial activity.
 9. The system of claim 1, wherein at least some of the human inputs are values indicative of human sensory perception.
 10. The system of claim 9, wherein at least some of values are indicative of perceived temperature.
 11. The system of claim 1, wherein the sensor measurements include measurements of ambient light level.
 12. The system of claim 6, wherein at least some of the sensors are passive infrared sensors in PIR motion detectors.
 13. The system of claim 1, wherein the at least one processor is adapted to generate comfort values by using an algorithm that employs linear discriminant analysis or a Fisher linear discriminant.
 14. The system of claim 1, wherein the at least one processor is adapted to calculate a decision boundary, to calculate comfort values indicative of distance from that decision boundary, and to update calculations of the decision boundary based on updated measurements.
 15. The system of claim 14, wherein the decision boundary is calculated as the mean of at least some extreme values.
 16. The system of claim 1, wherein at least some of the position values are indicative of the position of at least some of the humans, respectively.
 17. The system of claim 3, wherein at least some of the position values are indicative of the position of at least some of the portable nodes, respectively.
 18. The system of claim 1, wherein the at least one processor, when calculating position values, is adapted to take into account data transmitted wirelessly by a cellphone, smart phone or other mobile computing device.
 19. The system of claim 1, wherein the at least one actuator comprises multiple actuators, at least some of which are each adapted to alter air flow through a VAV box.

20. A method that comprises, in combination:
using input devices to accept inputs from a plurality of humans,
using sensors to take sensor measurements, the sensor measurements including measurements of temperature,
using at least one processor:
to process data indicative of the inputs and the sensor measurements,
to generate position values indicative of the position of at least some of the humans,

to generate comfort values indicative of comfort of at least some of the humans, and
to generate control signals, based at least in part on at least some of the comfort values, and
using at least one actuator to cause, in accordance with the control signals, mechanical motion of one or more objects to alter air flow.

* * * * *