US009258765B1

(12) **United States Patent**
daCosta

(10) **Patent No.: US 9,258,765 B1**
(45) **Date of Patent: *Feb. 9, 2016**

(54) **CHIRP NETWORKS**

(71) Applicant: **Francis daCosta**, Santa Clara, CA (US)

(72) Inventor: **Francis daCosta**, Santa Clara, CA (US)

(73) Assignee: **Dynamic Mesh Networks, Inc.**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/269,014**

(22) Filed: **May 2, 2014**

Related U.S. Application Data

(63) Continuation-in-part of application No. 13/541,446, filed on Jul. 3, 2012, and a continuation-in-part of application No. 12/696,947, filed on Jan. 29, 2010, now Pat. No. 8,520,691, which is a
(Continued)

(51) **Int. Cl.**

H04H 20/71	(2008.01)
H04W 48/12	(2009.01)

(Continued)

(52) **U.S. Cl.**
CPC ***H04W 48/12*** (2013.01); ***H04L 41/12*** (2013.01); ***H04L 45/04*** (2013.01); ***H04W 8/005*** (2013.01);
(Continued)

(58) **Field of Classification Search**
CPC H04L 12/2858; H04L 12/46; H04L 12/66; H04L 41/12; H04L 45/02; H04L 45/04; H04W 8/005; H04W 40/24; H04W 40/246; H04W 40/248; H04W 84/005; H04W 84/02; H04W 84/18; H04W 84/20; H04W 84/22; H04W 92/04; H04W 92/045
USPC 370/312, 315, 321, 328, 327, 340, 370/389–390, 400–401, 445–448, 537
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,226,045 A *	7/1993	Chuang	....	370/332
5,428,636 A *	6/1995	Meier	....	375/132

(Continued)

Primary Examiner — Peter Cheng
(74) *Attorney, Agent, or Firm* — Cherskov Flaynik & Gurda, LLC

(57) **ABSTRACT**

A wire-less/Wired mesh network is described, using a multi-slot modular mesh node to house diverse transceiver elements (e.g. IR, Wi-Fi, Powerline). A radio agnostic tree based mesh network is formed, based on what type of wire-less links are formed on the uplink and downlink of the backhaul and what type of radios etc are used for the Access Points AP, see FIG. **17**,**23**.

In addition to servicing IP based clients (e.g. Wi-Fi, WiMax, Bluetooth), the modular mesh nodes APs may also serve as receivers/collectors for low cost chirp devices. These devices are not "agile" and therefore contentious. APs, servicing these devices, alleviate potential contention by multiple means including: sending out a "incoming" CTS, efficient delivery through container based schedulable bus deliveries, and its reverse (moving chirp transmission times to be sequential).

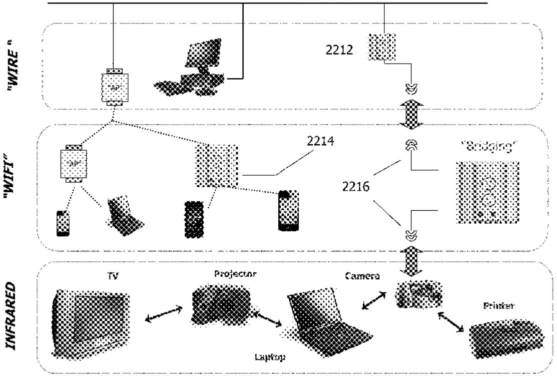
Multiple approaches for "pollen" to reach the intended "flower" in a timely manner are described. The method of transport is standard IP based packets yet security is inherent in this pollen-flower based system: only mesh nodes are privy to the routing tables that indicate that packet addresses are not IP. Multiple options to further obfuscate packet flow are presented.

A multi-agent based approach driving business process flow is described. Such agents can also provide specialized mesh network routing e.g. navigation agents for chirp devices.

Tree based routing and logical radio abstractions are revisited.

An organic approach to providing category/class based form of data type identification is proposed, to efficiently match publishers and subscribers, based on the type of data being sought. A private, secure and natively Publish/Subscribe M2M community is engendered at the edge. It has loosely, dynamic and ad hoc couplings to big data servers, also operating on their own private exchange/market place, using a real time publish/subscribe infrastructure with content categories used as part of pollen identification.

2 Claims, 67 Drawing Sheets



Related U.S. Application Data

continuation-in-part of application No. 11/084,330, filed on Mar. 17, 2005, now abandoned, which is a continuation-in-part of application No. 10/434,948, filed on May 8, 2003, now Pat. No. 7,420,952, said application No. 14/269,014 is a continuation of application No. 12/352,457, filed on Jan. 12, 2009, now Pat. No. 8,477,762, which is a continuation-in-part of application No. 11/266,884, filed on Nov. 4, 2005, now Pat. No. 7,583,648, said application No. 14/269,014 is a continuation of application No. 13/571,294, filed on Aug. 9, 2012, and a continuation-in-part of application No. 13/764,008, filed on Feb. 11, 2013, which is a continuation-in-part of application No. 13/627,883, filed on Sep. 26, 2012, now Pat. No. 8,923,186, said application No. 14/269,014 is a continuation-in-part of application No. 12/625,365, filed on Nov. 24, 2009, now Pat. No. 8,514,852.

(60) Provisional application No. 61/148,803, filed on Jan. 30, 2009, provisional application No. 61/117,502, filed on Nov. 24, 2008.

(51) **Int. Cl.**

H04W 40/02	(2009.01)
H04W 84/18	(2009.01)
H04L 12/715	(2013.01)
H04W 40/24	(2009.01)
H04W 84/00	(2009.01)
H04L 12/24	(2006.01)
H04W 84/22	(2009.01)
H04W 8/00	(2009.01)
H04W 84/02	(2009.01)
H04L 12/28	(2006.01)
H04L 12/751	(2013.01)
H04W 92/04	(2009.01)
H04L 12/66	(2006.01)
H04W 84/20	(2009.01)
H04L 12/46	(2006.01)

(52) **U.S. Cl.**

CPC ***H04W 40/02*** (2013.01); ***H04W 40/24*** (2013.01); ***H04W 40/246*** (2013.01); ***H04W 40/248*** (2013.01); ***H04W 84/005*** (2013.01); ***H04W 84/18*** (2013.01); ***H04W 84/22*** (2013.01); *H04L 12/2858* (2013.01); *H04L 12/46* (2013.01); *H04L 12/66* (2013.01); *H04L 45/02* (2013.01); *H04W 84/02* (2013.01); *H04W 84/20* (2013.01); *H04W 92/04* (2013.01); *H04W 92/045* (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,567,383	B1 *	5/2003	Bohnke	370/280
7,420,952	B2	9/2008	daCosta	
7,500,004	B1 *	3/2009	Homer	709/228
7,583,648	B2	9/2009	daCosta	
7,649,852	B2	1/2010	Thubert et al.	
7,738,402	B2	6/2010	Feldman et al.	
7,835,265	B2 *	11/2010	Wang et al.	370/216
2002/0118682	A1 *	8/2002	Choe	370/395.31
2002/0137459	A1	9/2002	Elbata et al.	
2002/0191573	A1 *	12/2002	Whitehill et al.	370/338
2003/0115282	A1	6/2003	Rose	
2004/0042435	A1 *	3/2004	Soomro et al.	370/338
2004/0095900	A1 *	5/2004	Siegel	370/328
2004/0100929	A1 *	5/2004	Garcia-Luna-Aceves	370/338
2006/0119508	A1 *	6/2006	Miller	342/357.17
2007/0183346	A1 *	8/2007	Thubert et al.	370/254
2009/0304381	A1	12/2009	Muppidi et al.	

* cited by examiner

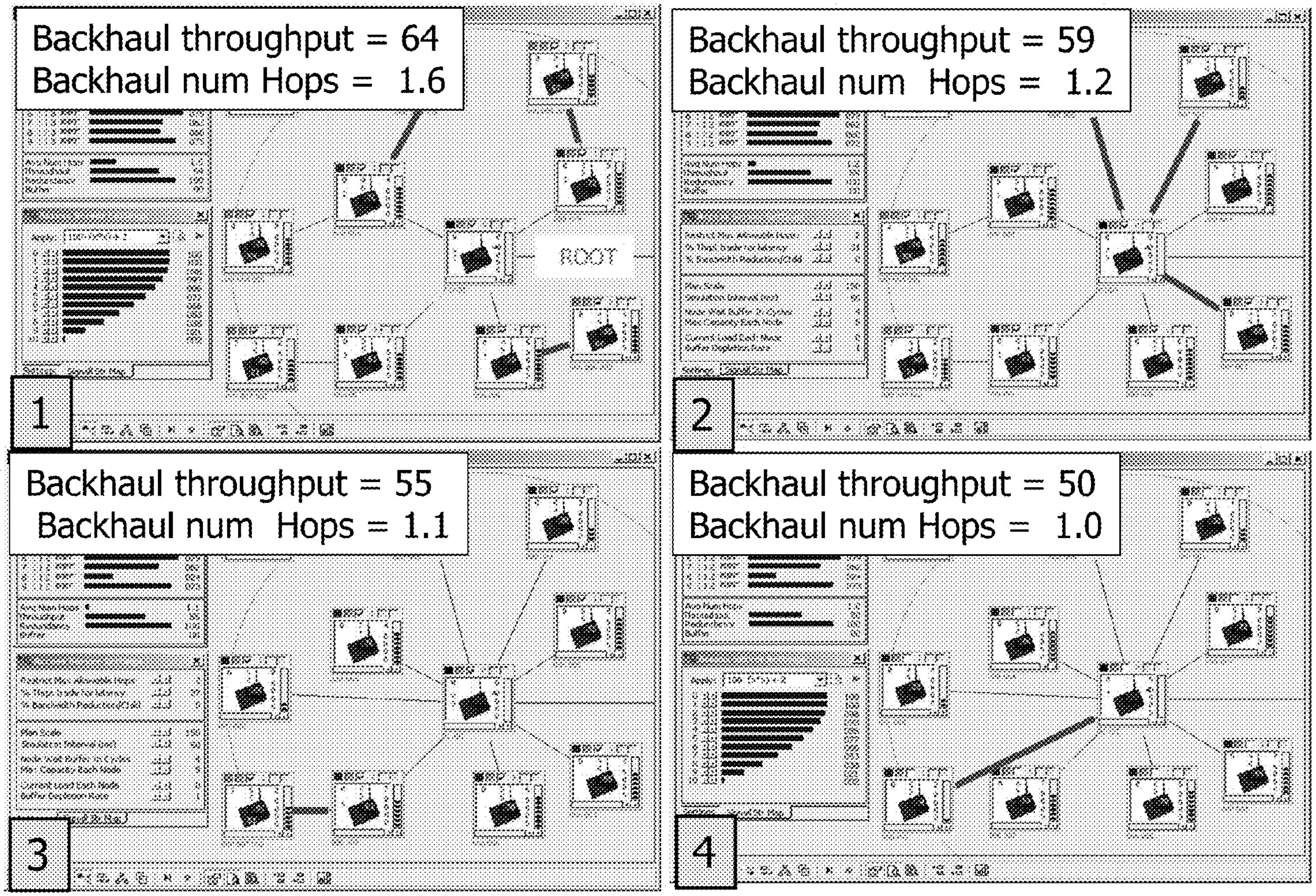

Backhaul throughput = 64
Backhaul num Hops = 1.6
ROOT
1
Backhaul throughput = 59
Backhaul num Hops = 1.2
2
Backhaul throughput = 55
Backhaul num Hops = 1.1
3
Backhaul throughput = 50
Backhaul num Hops = 1.0
4

Figure 1

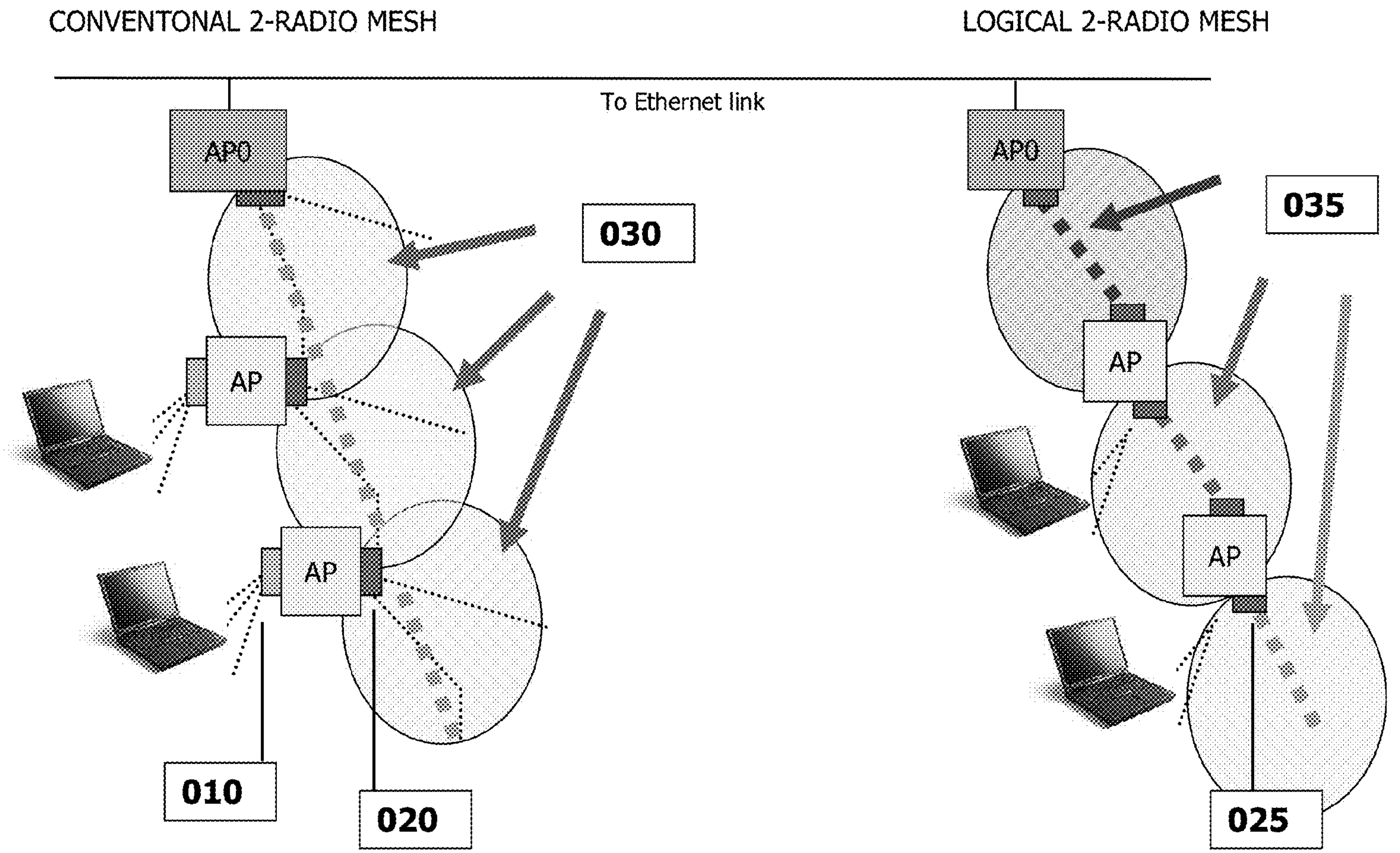

CONVENTONAL 2-RADIO MESH
LOGICAL 2-RADIO MESH
To Ethernet link
AP0
AP
AP
030
010
020
AP0
AP
AP
035
025

Figure 2

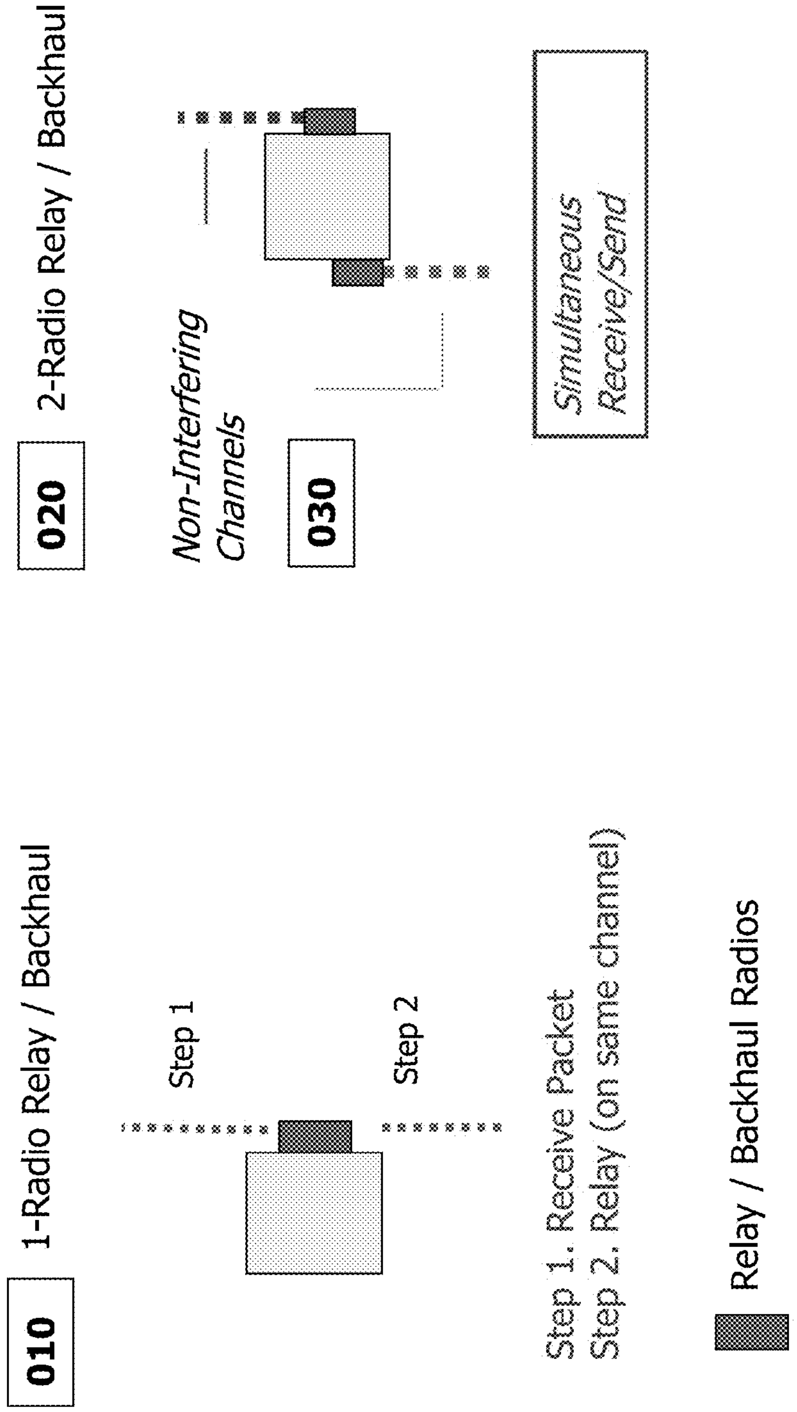

010
1-Radio Relay / Backhaul
Step 1
Step 2
Step 1. Receive Packet
Step 2. Relay (on same channel)
▨ Relay / Backhaul Radios
020
2-Radio Relay / Backhaul
Non-Interfering Channels
030
Simultaneous Receive/Send

Figure 3

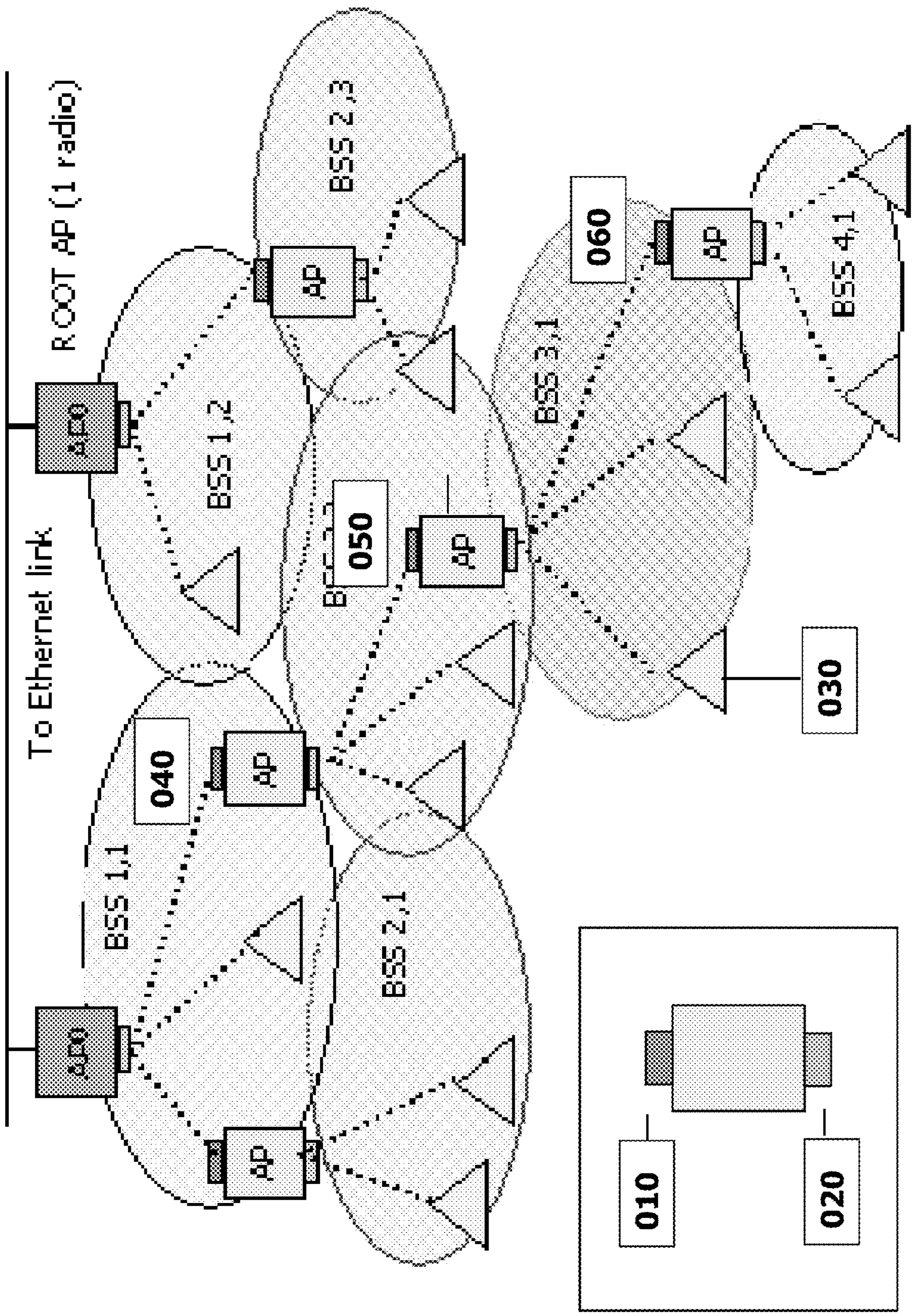

To Ethernet link
ROOT AP (1 radio)
RAP
RAP
AP
AP
AP
AP
AP
BSS 1,1
BSS 1,2
BSS 2,1
BSS 2,3
BSS 3,1
BSS 4,1
040
050
060
030
010
020

Figure 4

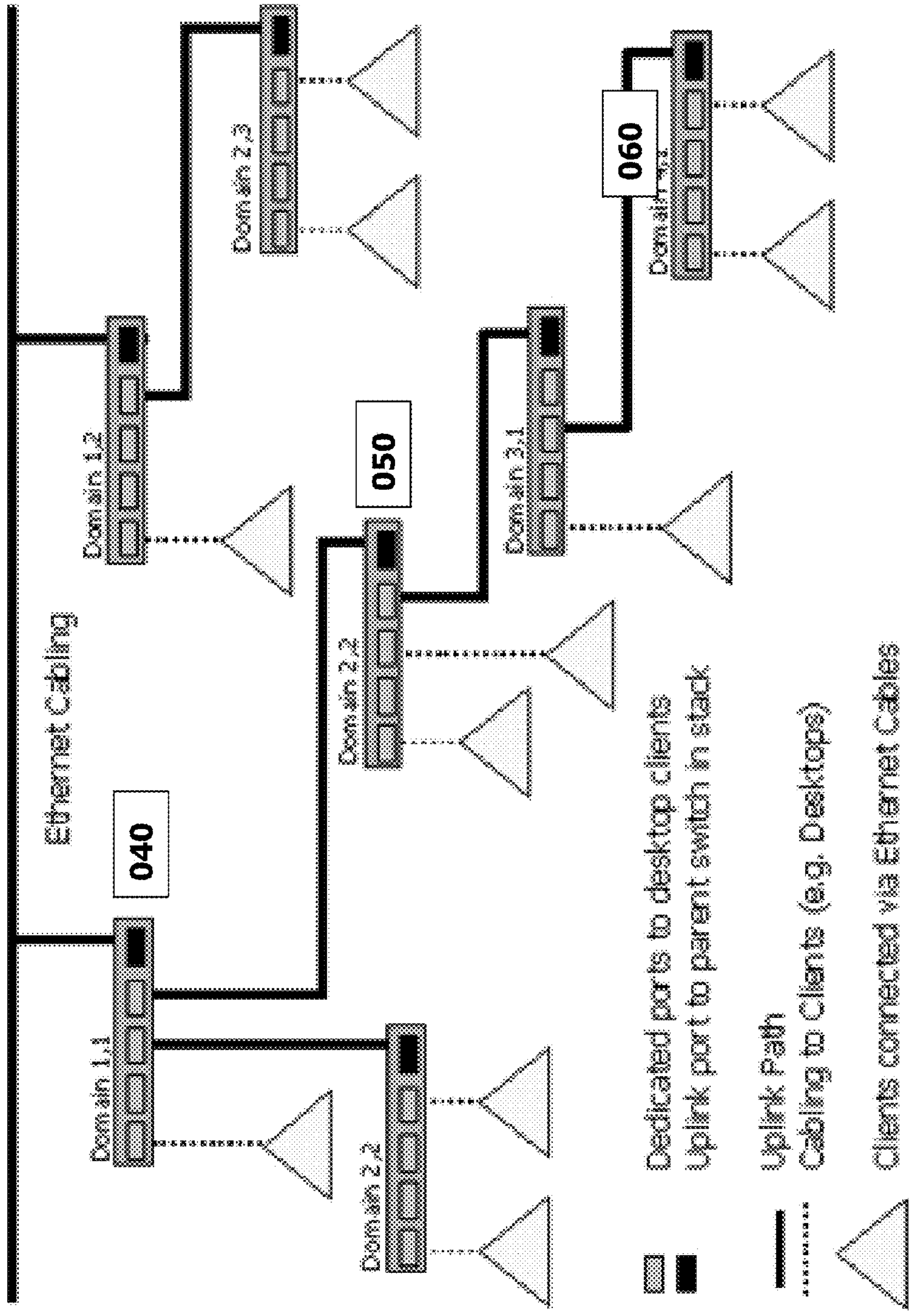

Ethernet Cabling
040
050
060
Domain 1.1
Domain 1.2
Domain 2.2
Domain 2.2
Domain 2.3
Domain 3.1
Domain
Dedicated ports to desktop clients
Uplink port to parent switch in stack
Uplink Path
Cabling to Clients (e.g. Desktops)
Clients connected via Ethernet Cables

Figure 5

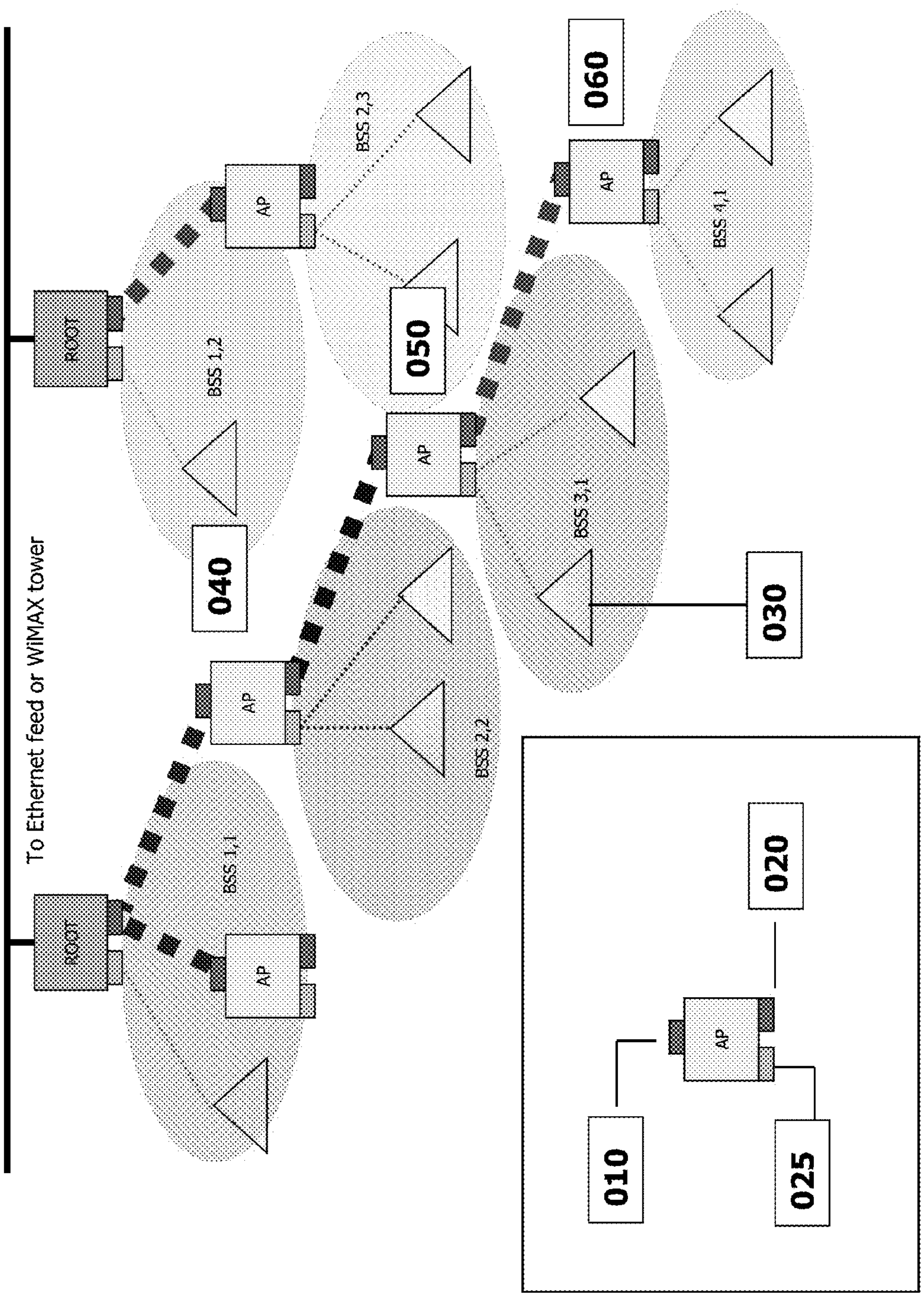

To Ethernet feed or WiMAX tower
ROOT
ROOT
AP
AP
AP
AP
AP
AP
BSS 1,1
BSS 1,2
BSS 2,2
BSS 2,3
BSS 3,1
BSS 4,1
040
050
060
030
010
020
025

Figure 6

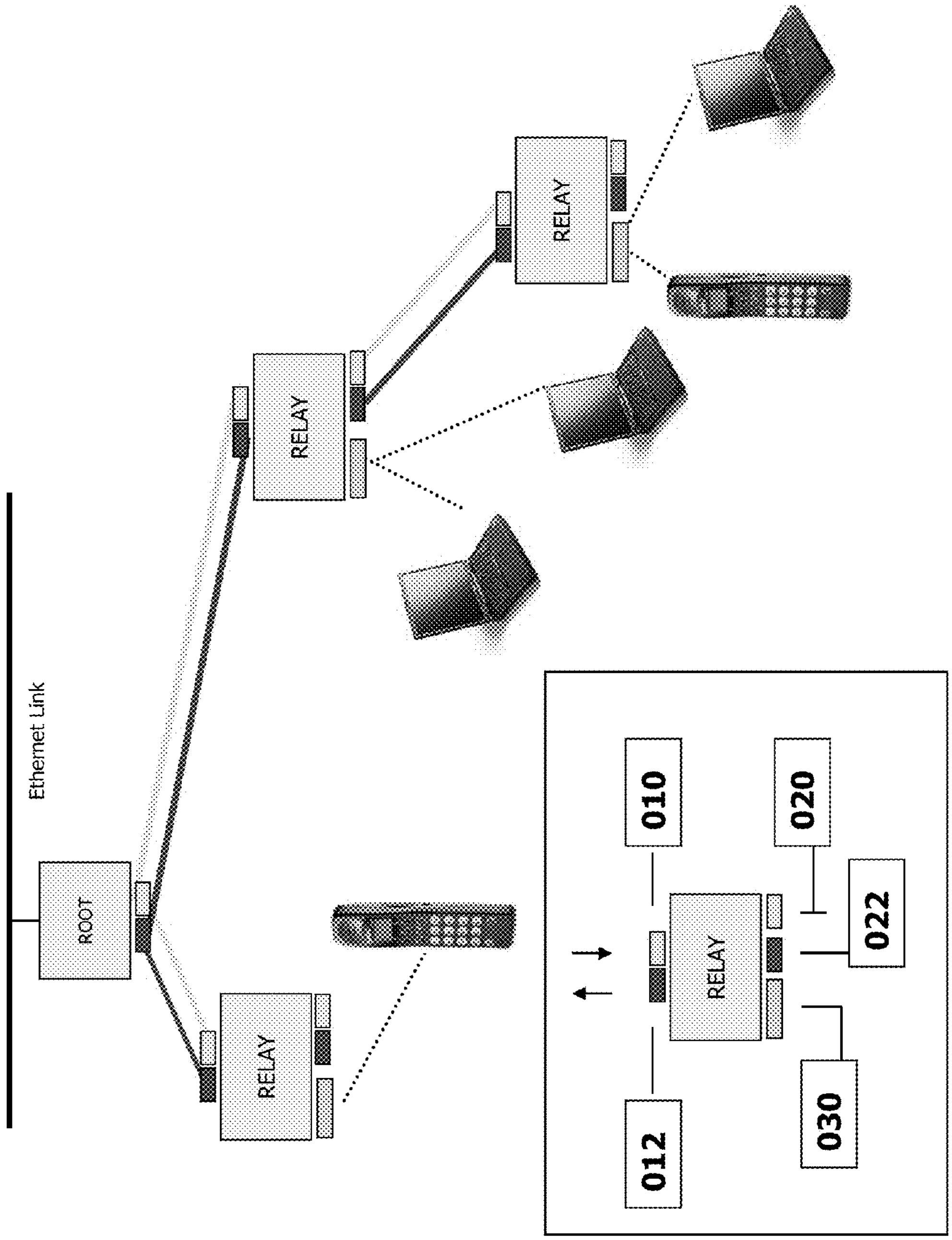

Ethernet Link
ROOT
RELAY
RELAY
RELAY
RELAY
010
012
020
022
030

Figure 7

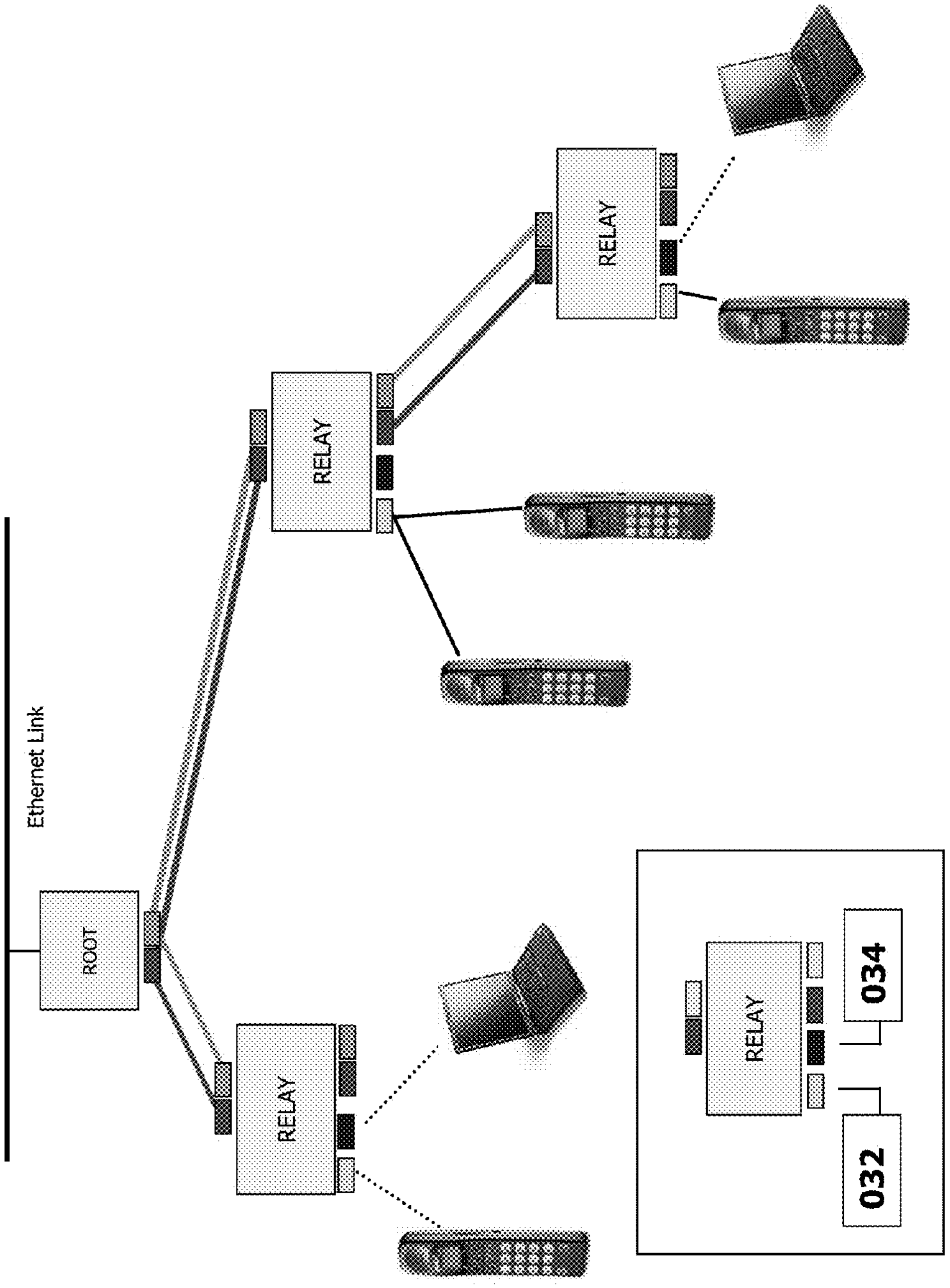


Figure 8

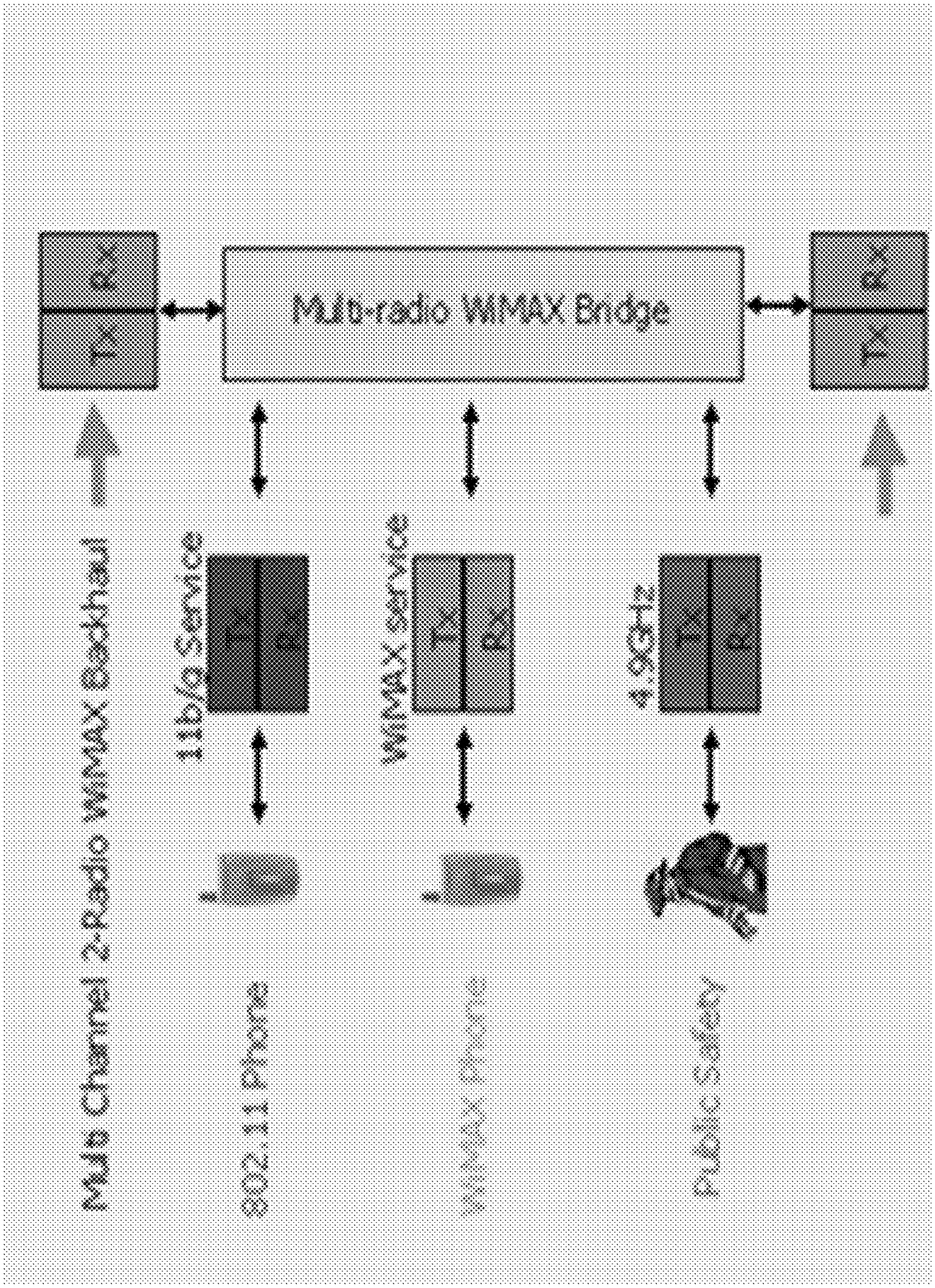

Multi Channel 2-Radio WiMAX Backhaul
TX
RX
Multi-radio WiMAX Bridge
TX
RX
11b/g Service
TX
RX
WiMAX service
TX
RX
4.9GHz
TX
RX
802.11 Phone
WiMAX Phone
Public Safety

Figure 9

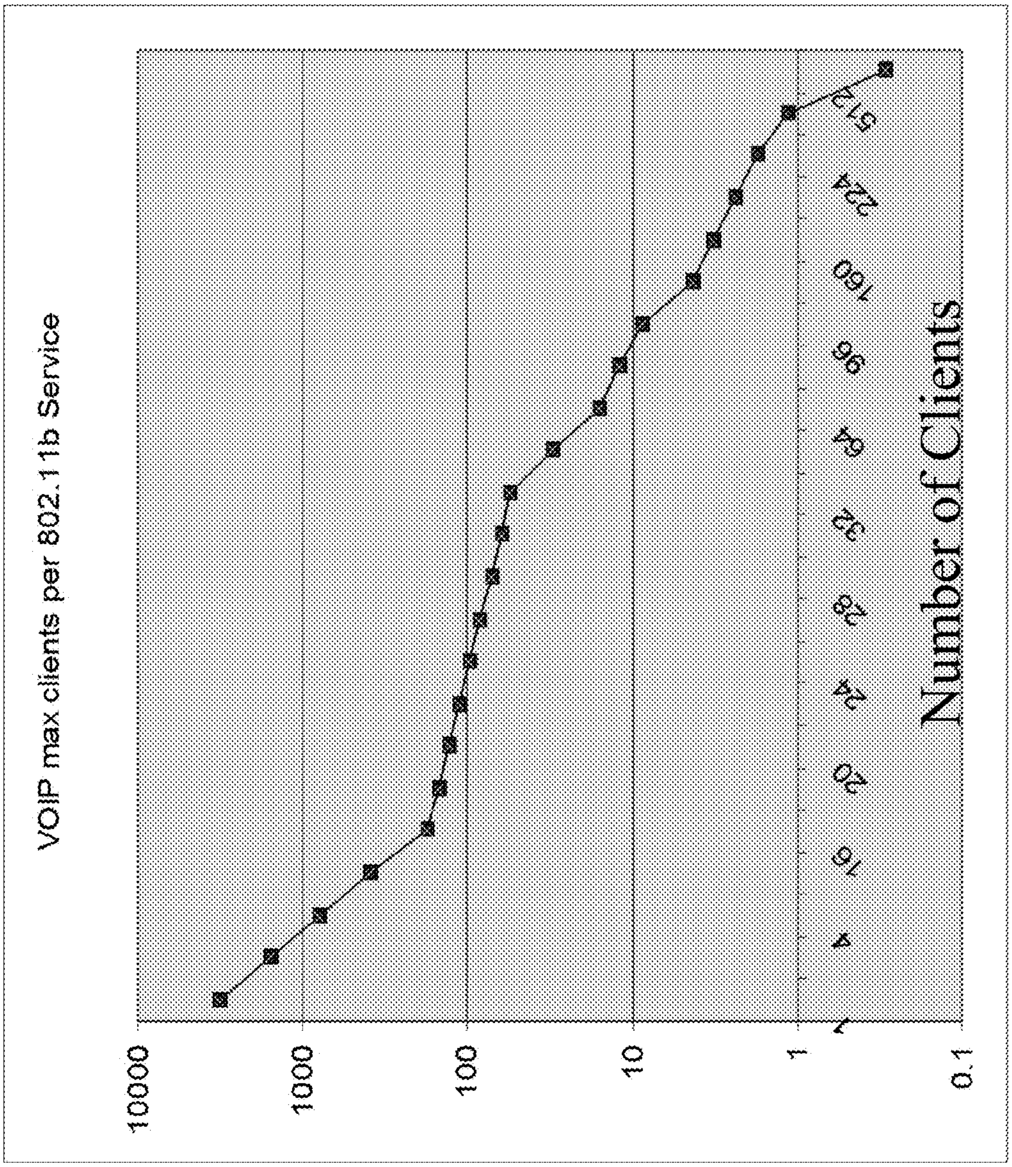

VOIP max clients per 802.11b Service
10000
1000
100
10
1
0.1
1
4
16
20
24
28
32
64
96
160
224
512
Number of Clients

Figure 10

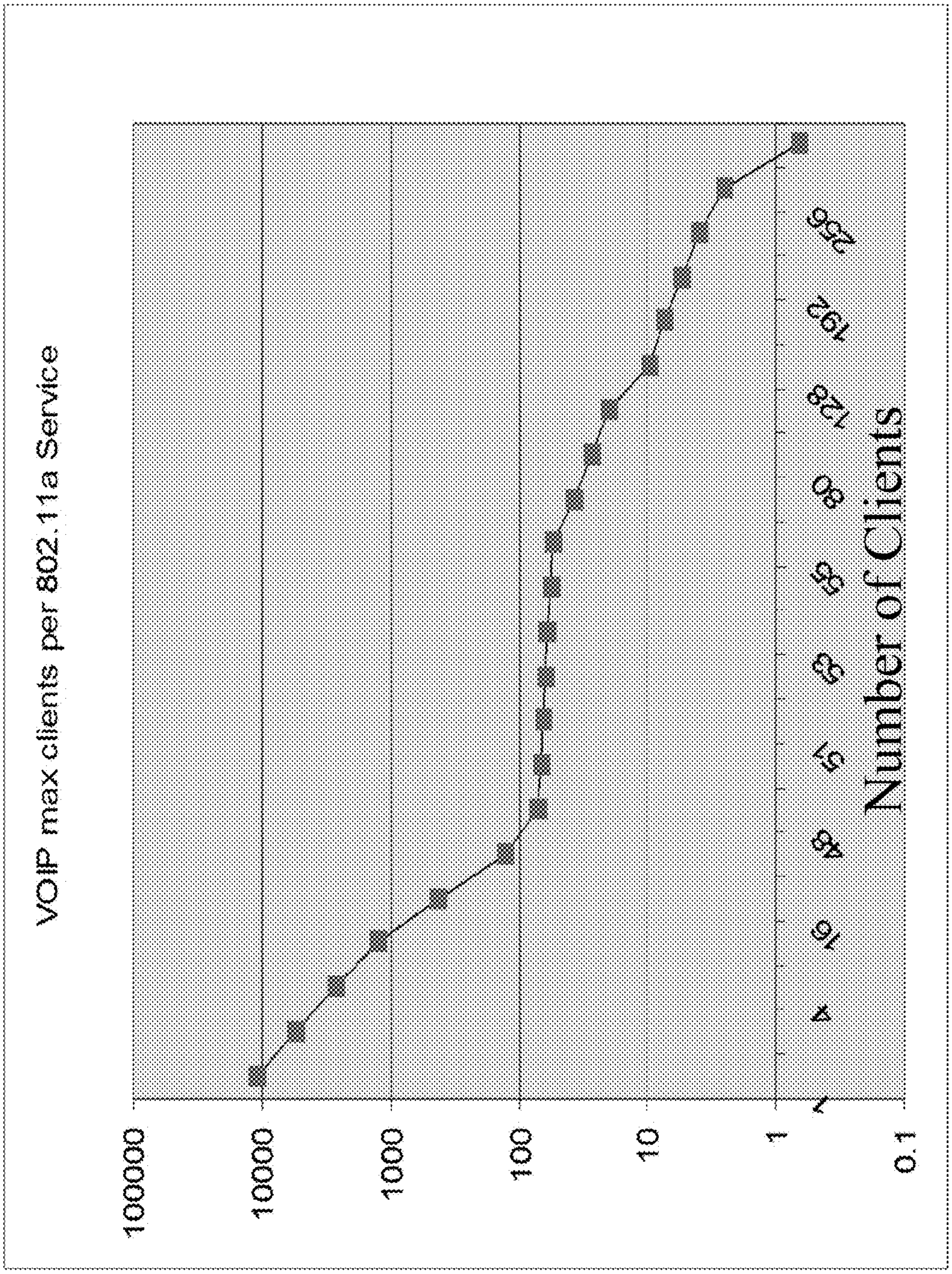

VOIP max clients per 802.11a Service
100000
10000
1000
100
10
1
0.1
1
4
16
48
57
63
65
80
128
192
256
Number of Clients

Figure 11

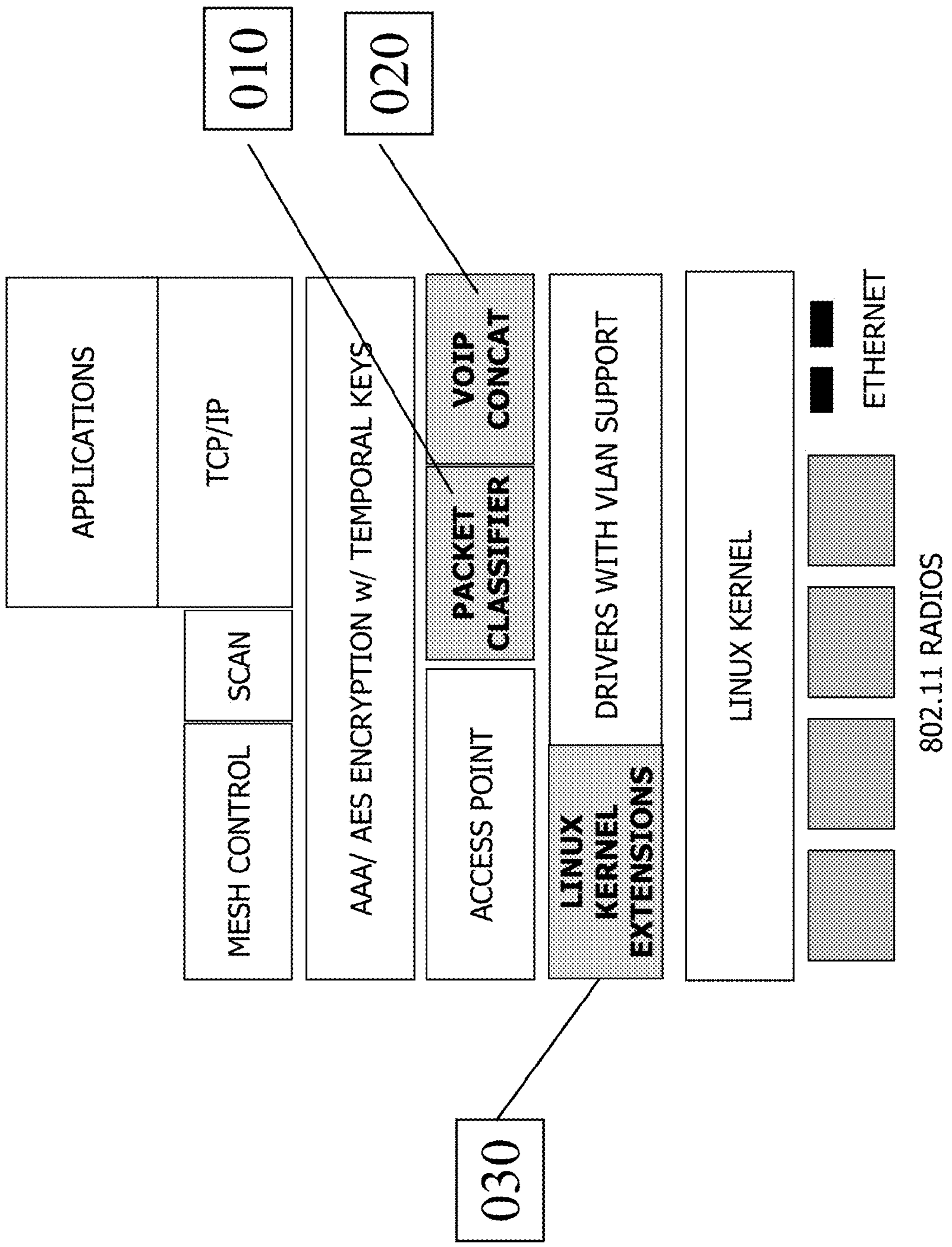

010
020
030
APPLICATIONS
TCP/IP
MESH CONTROL
SCAN
AAA/ AES ENCRYPTION w/ TEMPORAL KEYS
ACCESS POINT
PACKET CLASSIFIER
VOIP CONCAT
LINUX KERNEL EXTENSIONS
DRIVERS WITH VLAN SUPPORT
LINUX KERNEL
802.11 RADIOS
ETHERNET

Figure 12

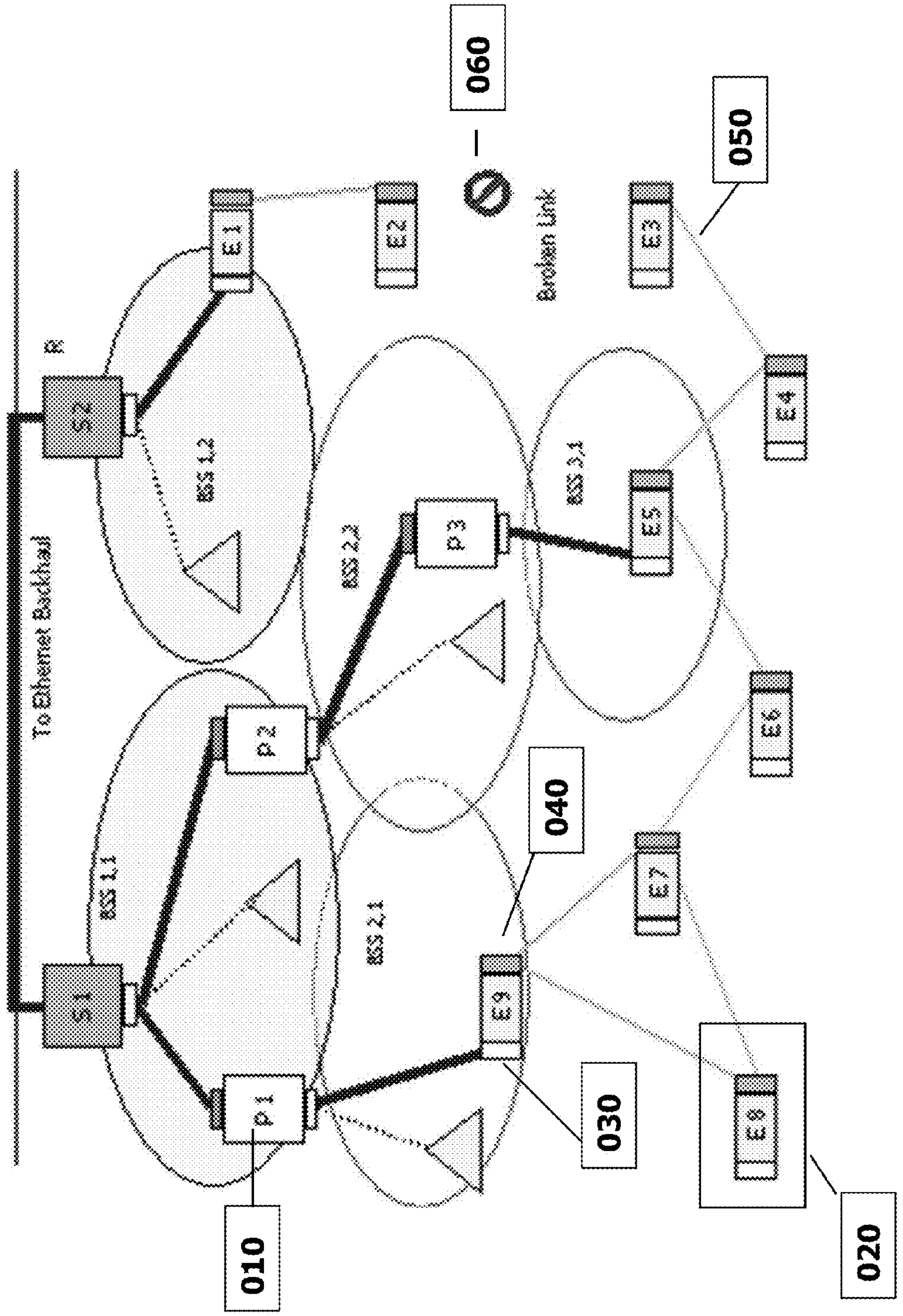

To Ethernet Backhaul
R
S1
S2
P1
P2
P3
E1
E2
E3
E4
E5
E6
E7
E8
E9
BSS 1.1
BSS 1.2
BSS 2.1
BSS 2.2
BSS 3.1
Broken Link
010
020
030
040
050
060

FIGURE 13

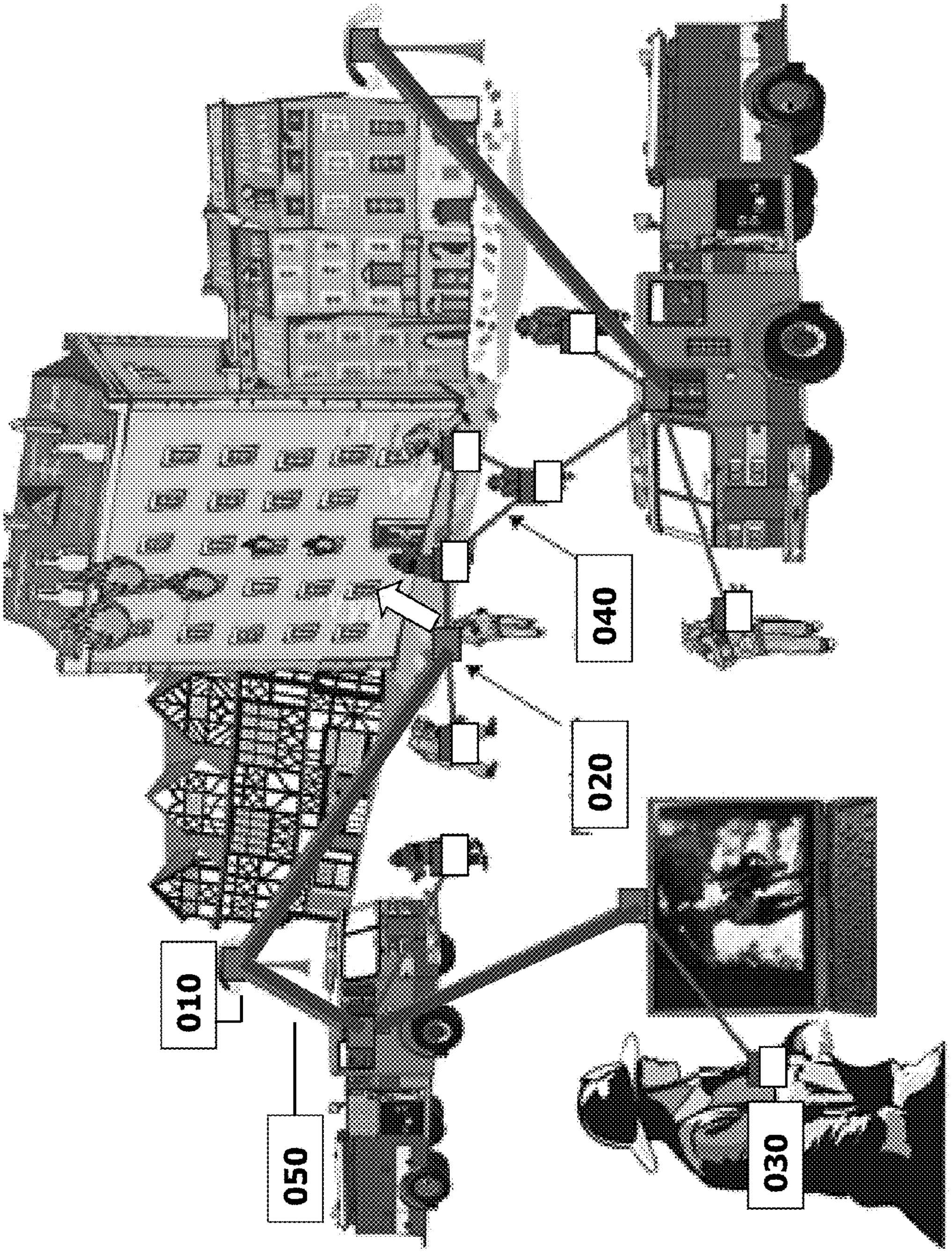


Figure 14

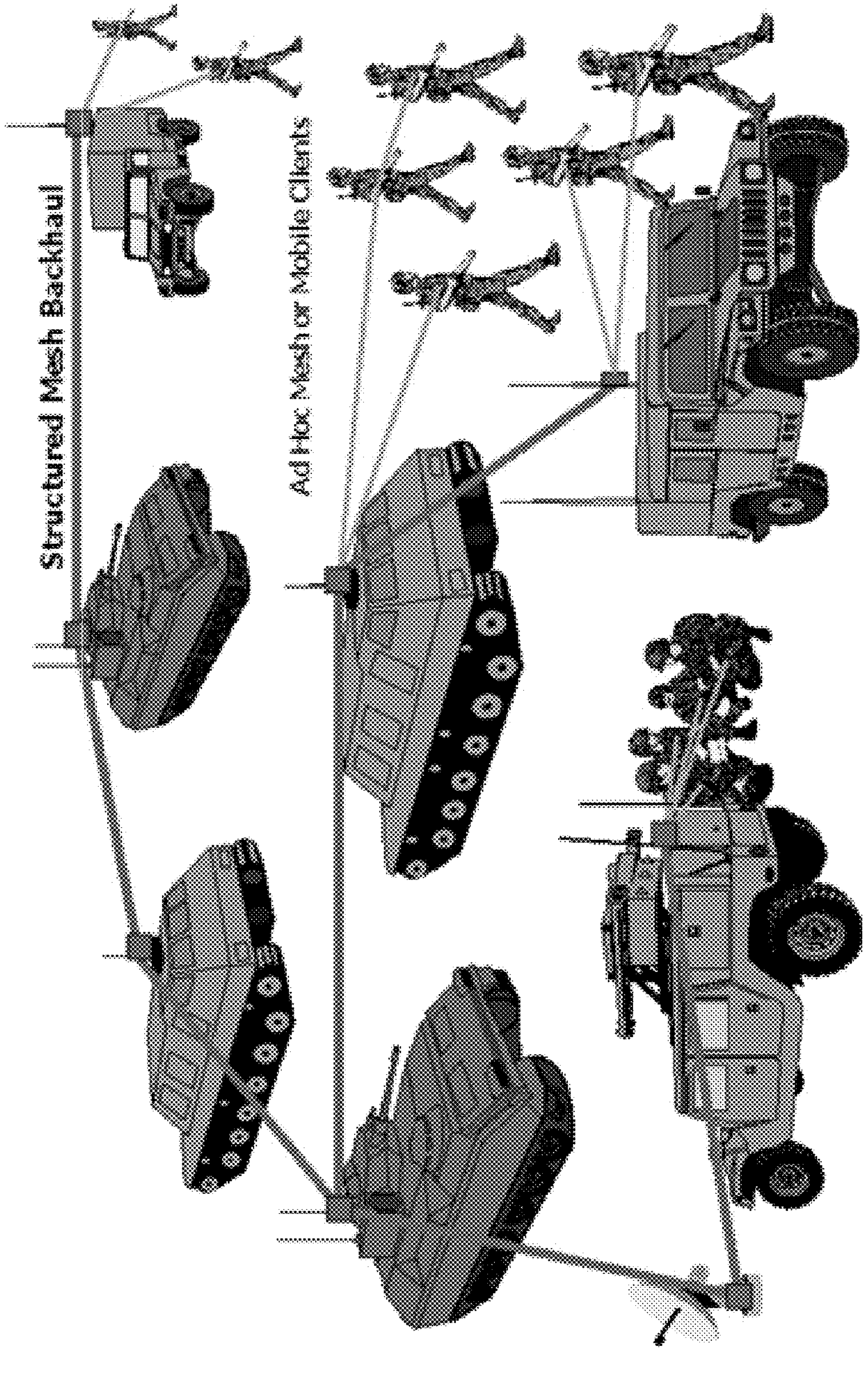


Figure 15

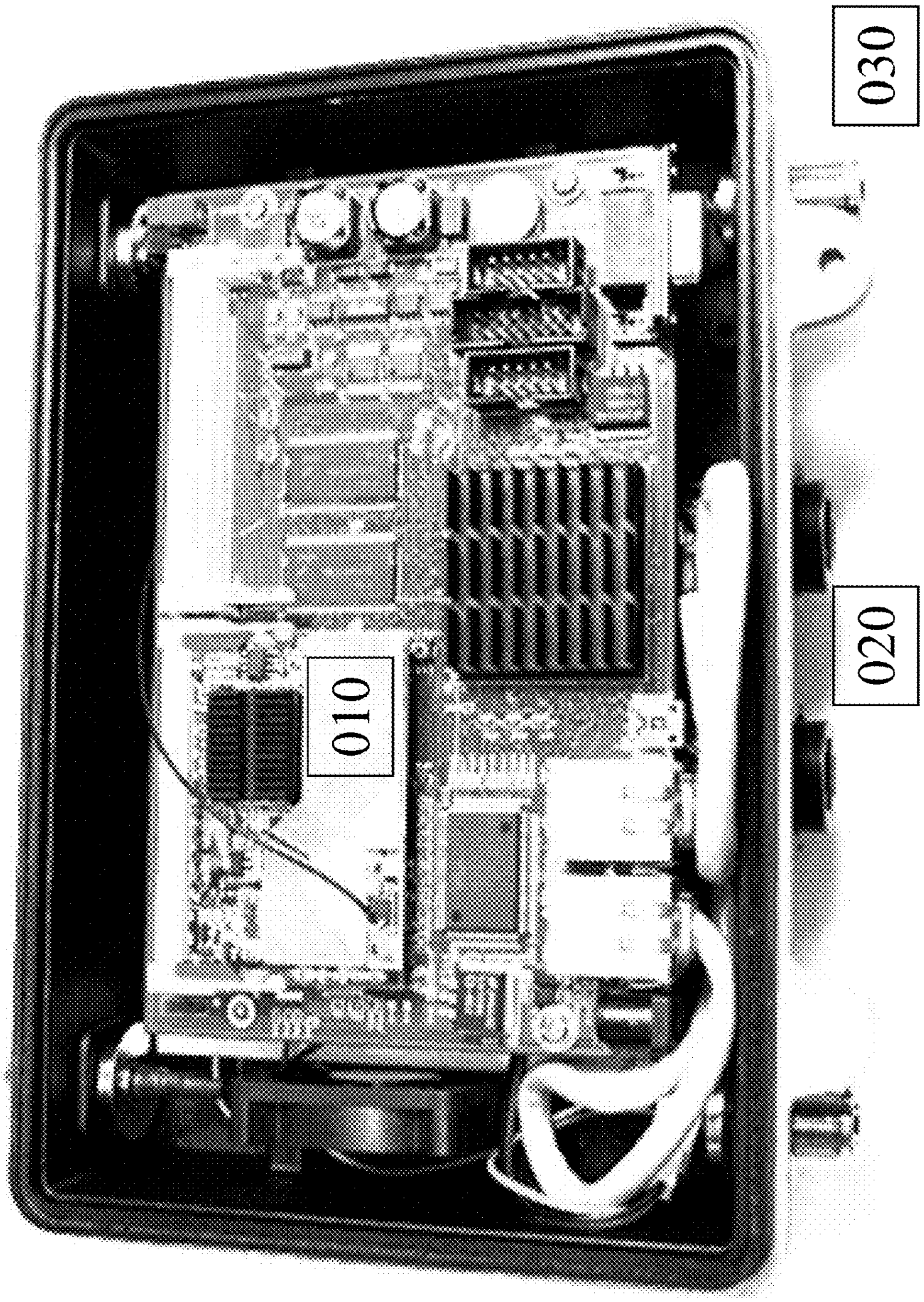


Figure 16

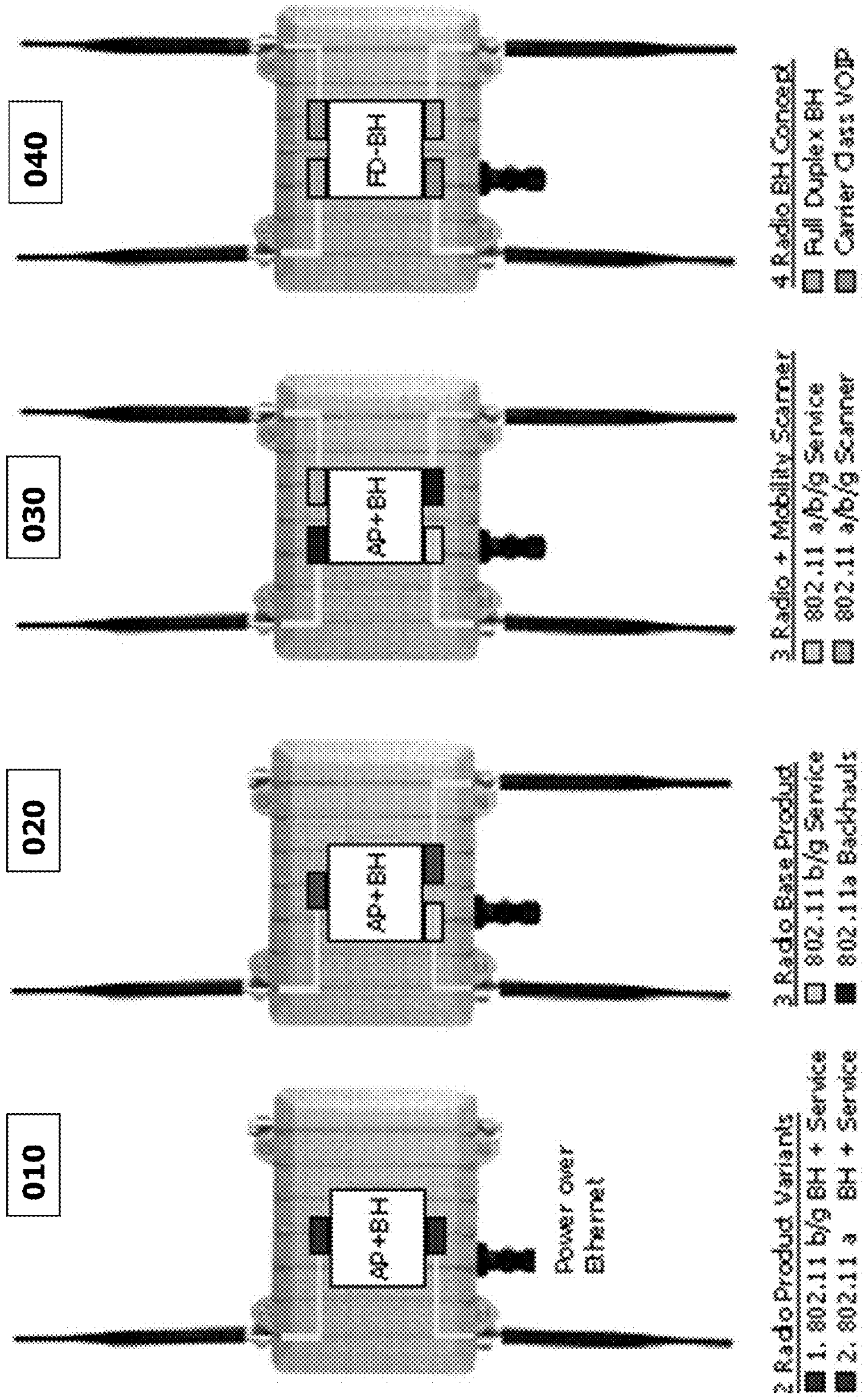

010
AP+BH
Power over Ethernet
2 Radio Product Variants
■ 1. 802.11 b/g BH + Service
■ 2. 802.11 a BH + Service
020
AP+BH
3 Radio Base Product
□ 802.11 b/g Service
■ 802.11a Backhauls
030
AP+BH
3 Radio + Mobility Scanner
□ 802.11 a/b/g Service
▨ 802.11 a/b/g Scanner
040
FD-BH
4 Radio BH Concept
▨ Full Duplex BH
▨ Carrier Class VOIP

Figure 17

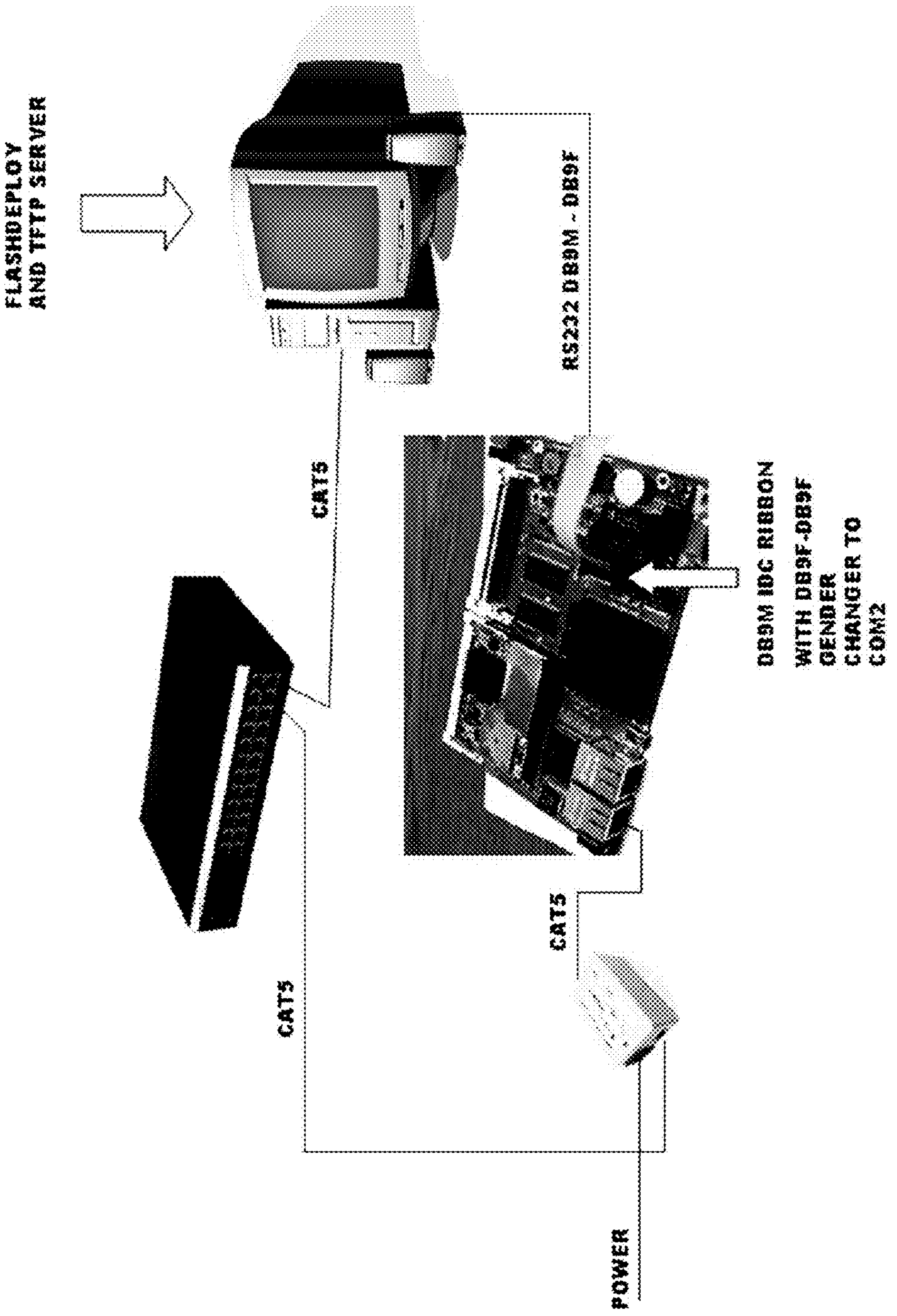
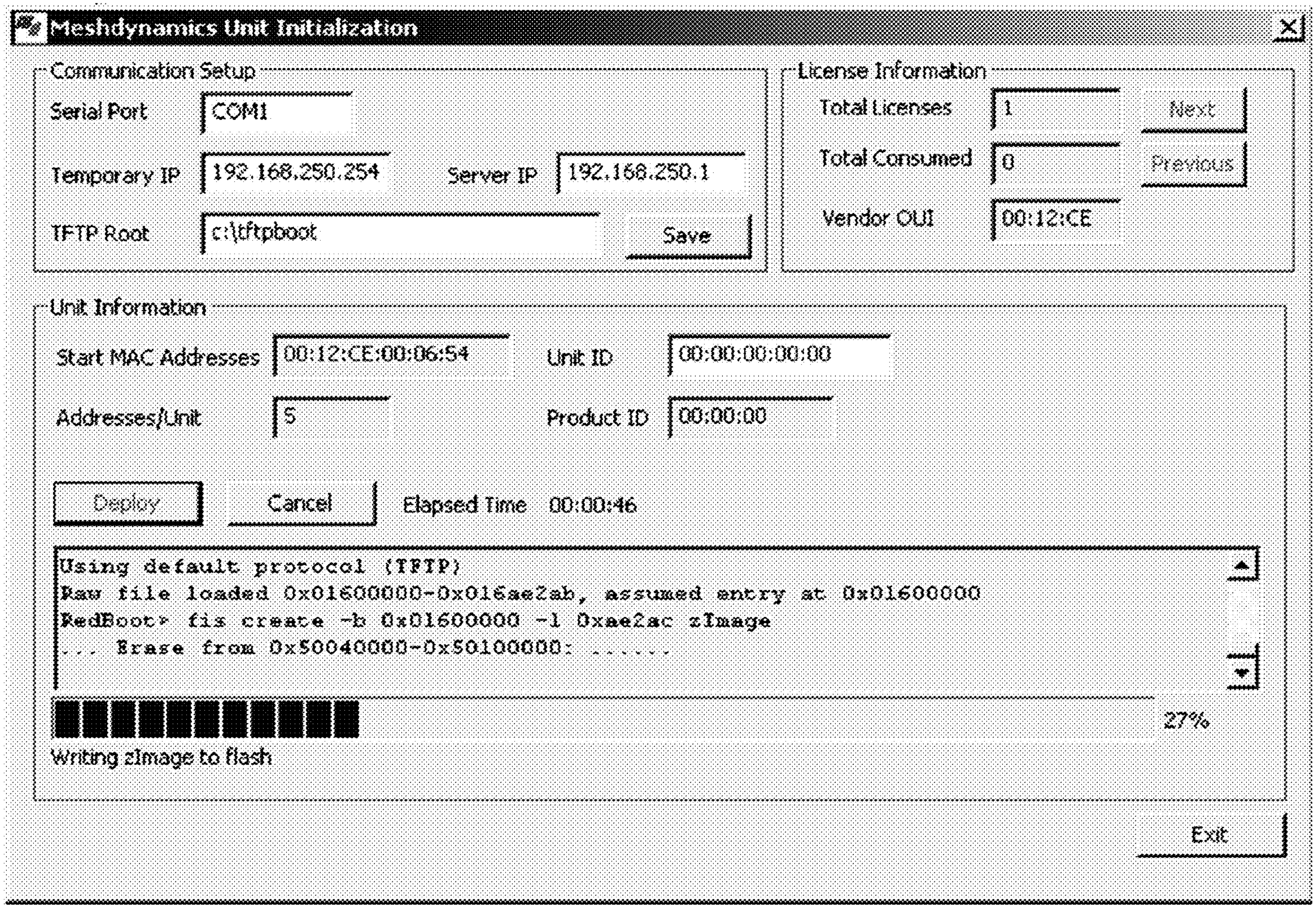

FLASHDEPLOY
AND TFTP SERVER
RS232 DB9M - DB9F
CAT5
DB9M IDC RIBBON
WITH DB9F-DB9F
GENDER
CHANGER TO
COM2
CAT5
CAT5
POWER

Figure 18



Serial Port: COM1

Temporary IP: 192.168.250.254

Server IP: 192.168.250.1

TFTP Root: c:\tftpboot

Total Licenses: 1

Total Consumed: 0

Vendor OUI: 00:12:CE

Start MAC Addresses: 00:12:CE:00:06:54

Unit ID: 00:00:00:00:00

Addresses/Unit: 5

Product ID: 00:00:00

Elapsed Time 00:00:46

```
Using default protocol (TFTP)
Raw file loaded 0x01600000-0x016ae2ab, assumed entry at 0x01600000
RedBoot> fis create -b 0x01600000 -l 0xae2ac zImage
... Erase from 0x50040000-0x50100000: ......
```

27%

Writing zImage to flash

Figure 19

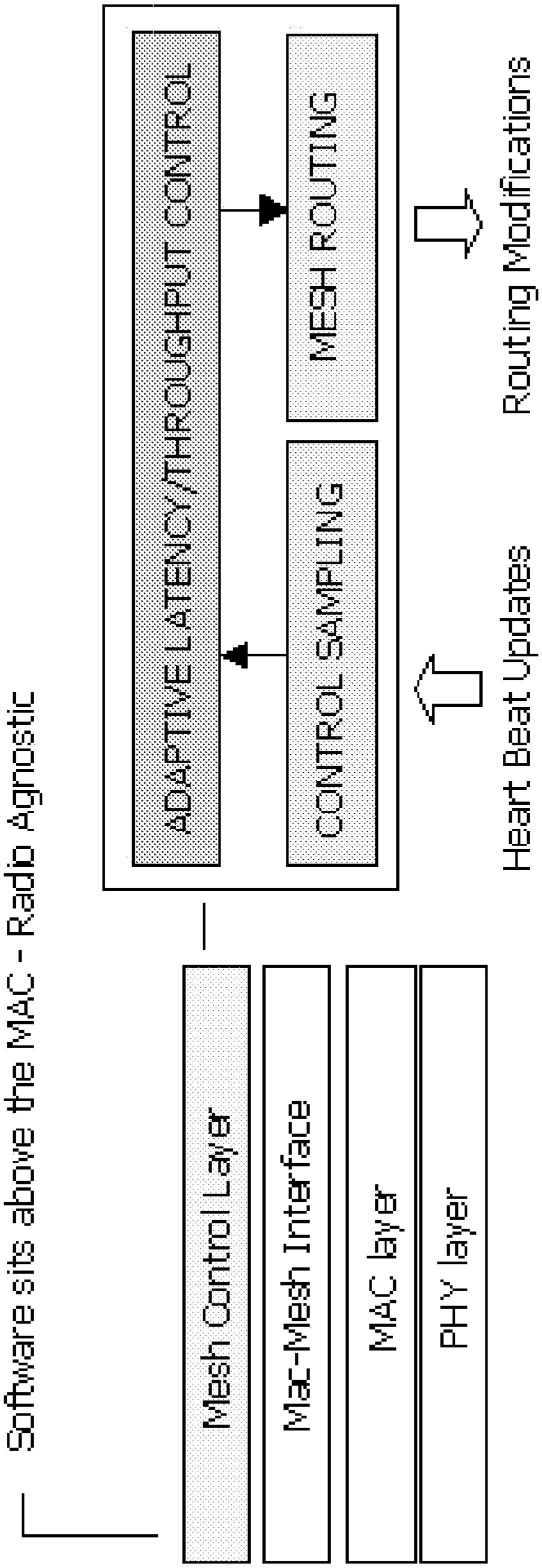

Software sits above the MAC - Radio Agnostic
Mesh Control Layer
Mac-Mesh Interface
MAC layer
PHY layer
ADAPTIVE LATENCY/THROUGHPUT CONTROL
CONTROL SAMPLING
MESH ROUTING
Heart Beat Updates
Routing Modifications

Figure 20

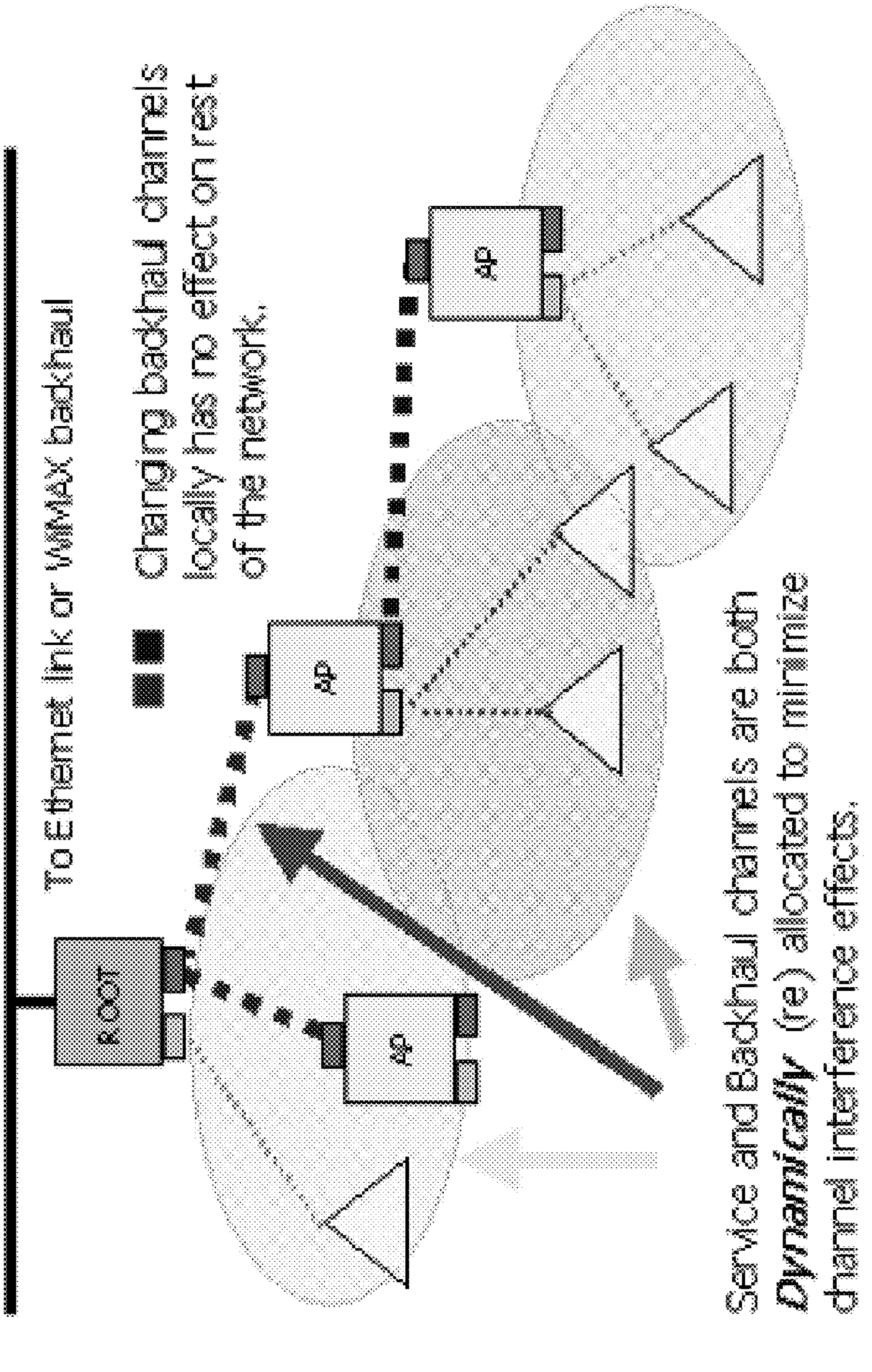

To Ethernet link or WiMAX backhaul
Changing backhaul channels locally has no effect on rest of the network.
ROOT
AP
AP
AP
Service and Backhaul channels are both *Dynamically* (re) allocated to minimize channel interference effects.

Figure 21

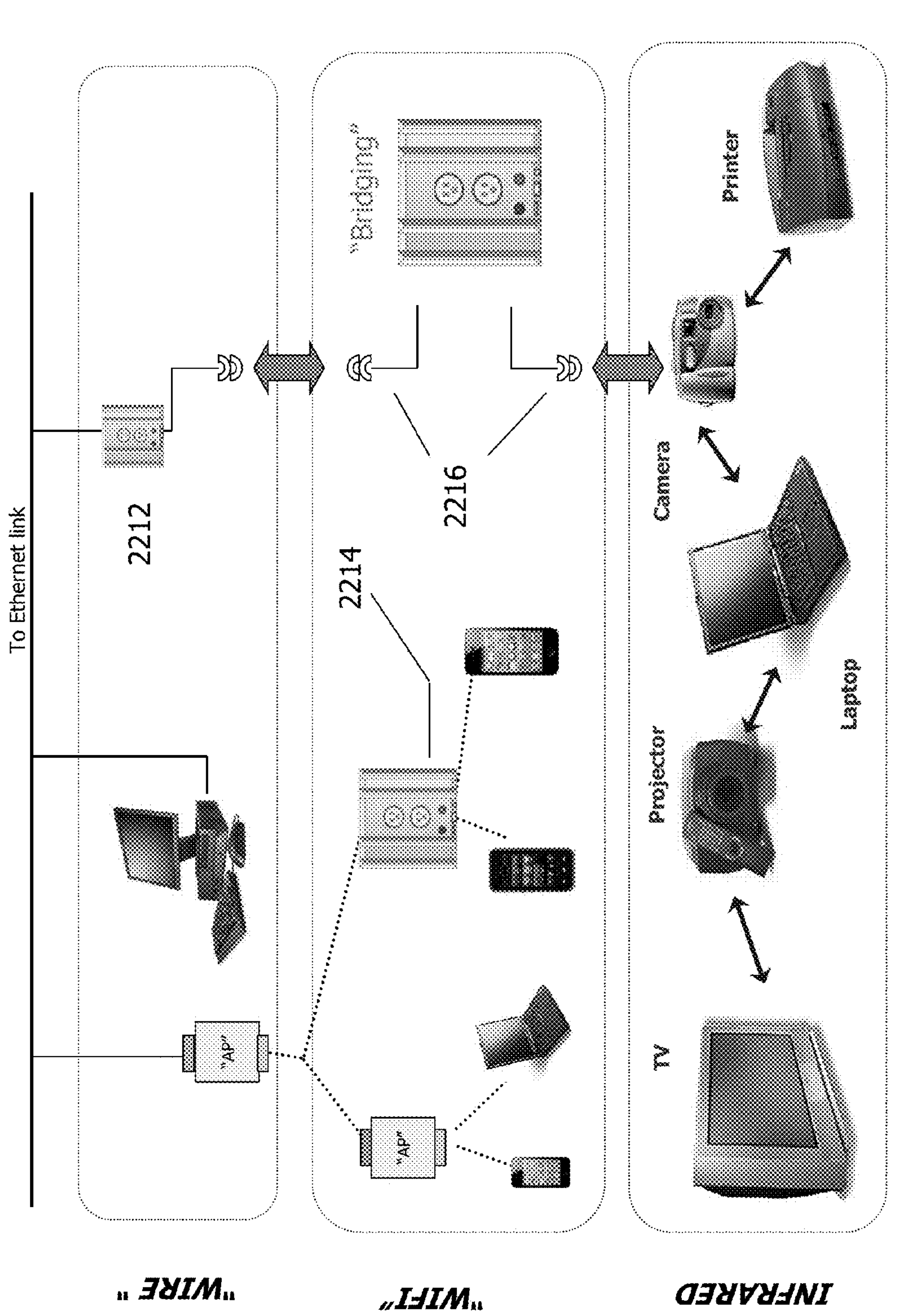
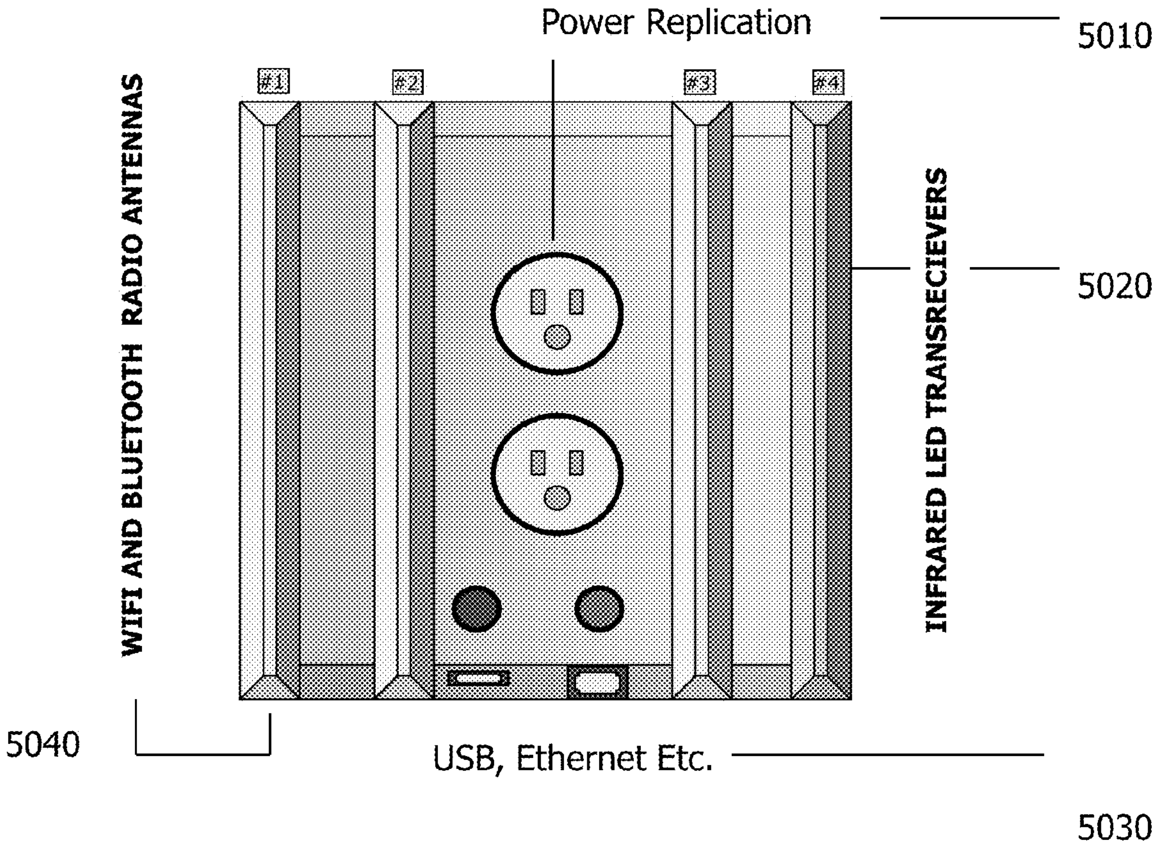

To Ethernet link
2212
"AP"
"WIRE"
2214
2216
"Bridging"
"AP"
"WIFI"
TV
Projector
Laptop
Camera
Printer
INFRARED

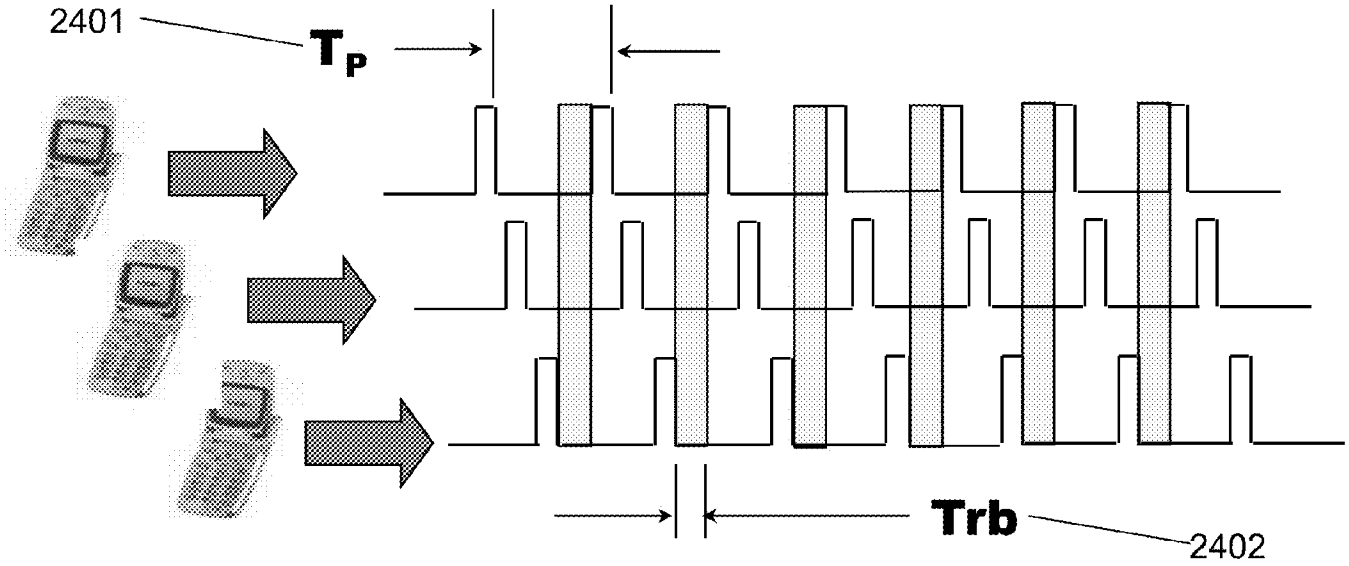
Figure 22


Power Replication
5010
#1
#2
#3
#4
WIFI AND BLUETOOTH RADIO ANTENNAS
INFRARED LED TRANSRECIEVERS
5020
5040
USB, Ethernet Etc.
5030

N- Logical "Radio" Modular Mesh aware Bridge

Figure 23

Receive Block: Time for Bulk VOIP Transmissions



▨ BLOCK OF TIME RESERVED FOR RECEVE PACKETS

- T_P : PACKETIZATION TIME: 20 ms for G.711 and G.729 CODECS
- T_{rb} : RECEIVE BLOCK: For all receive packets (from Access Point to Devices)

Figure 24

Separate Voice and Data Service

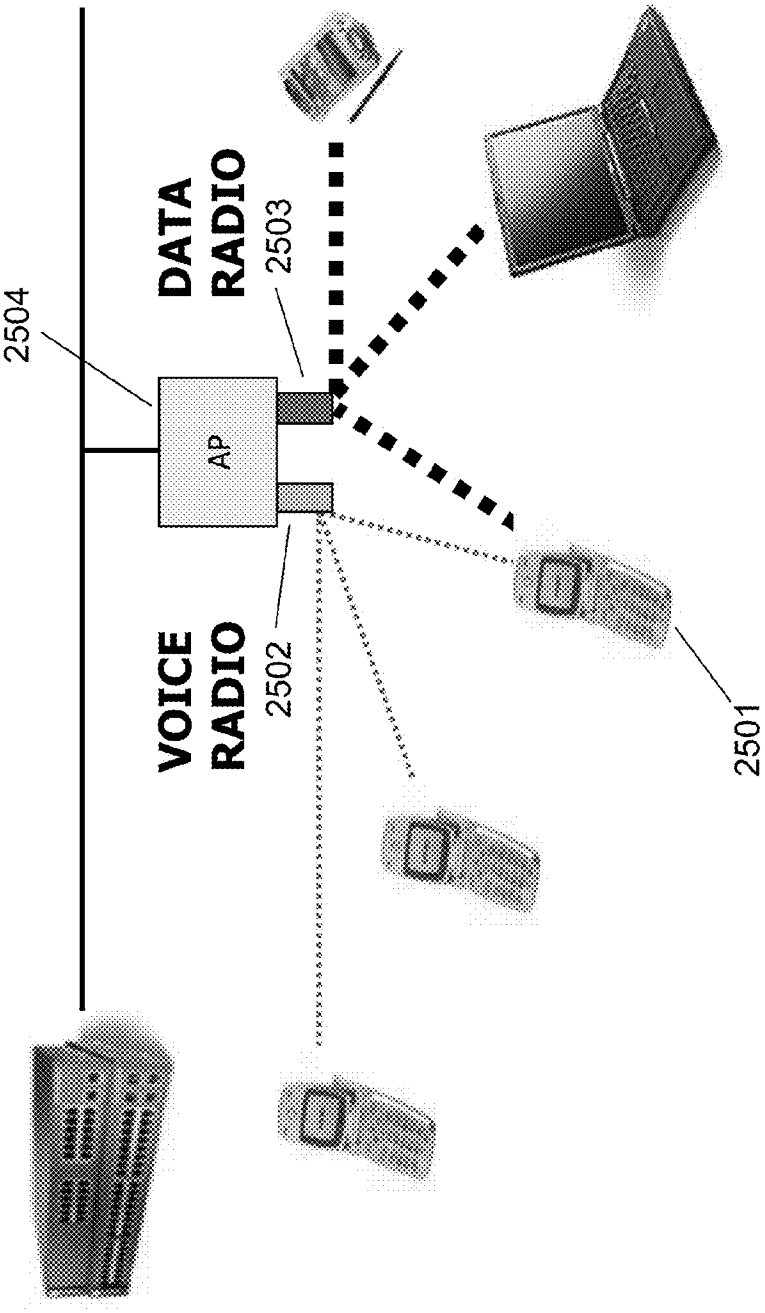


Figure 25

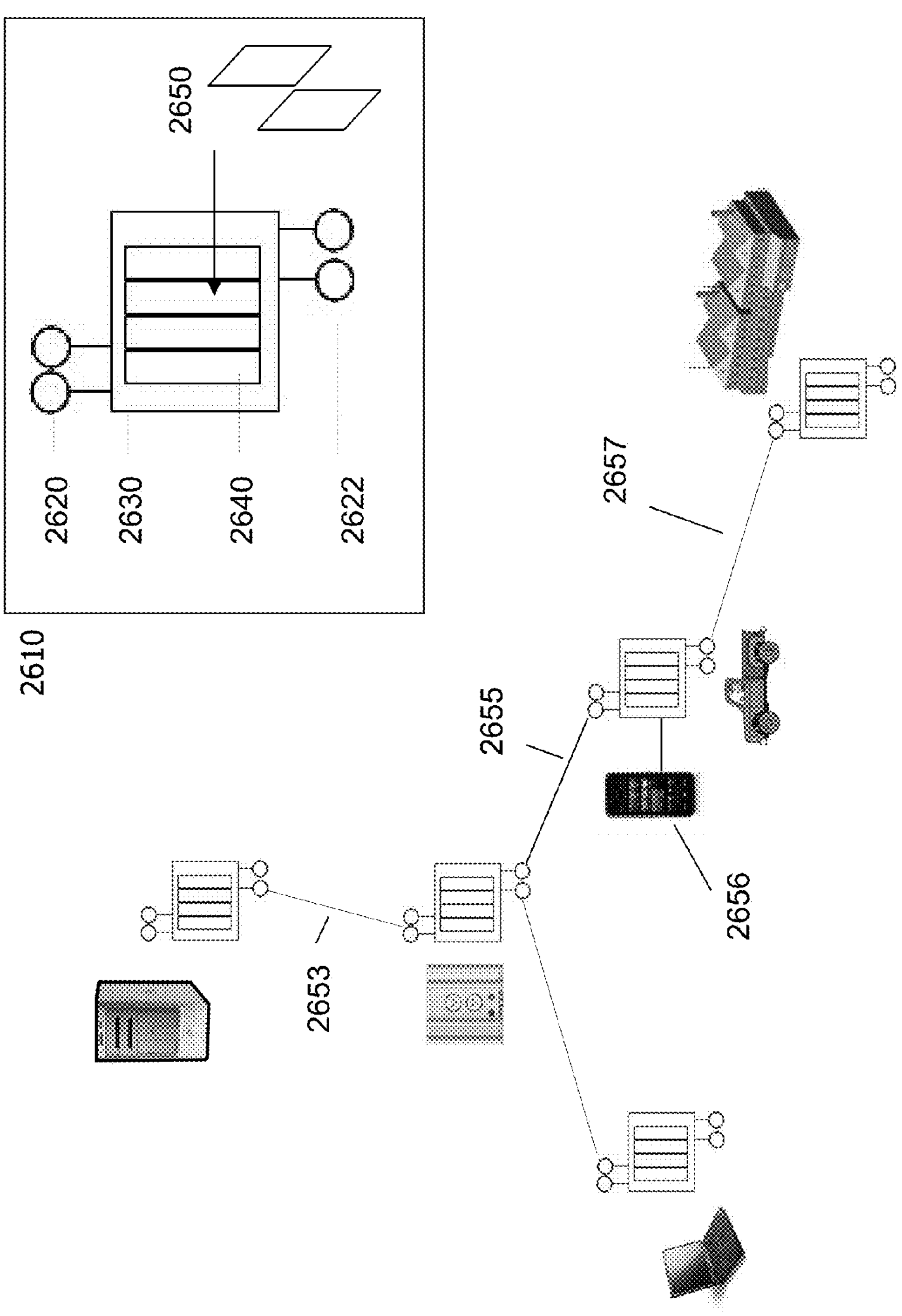


Figure 26

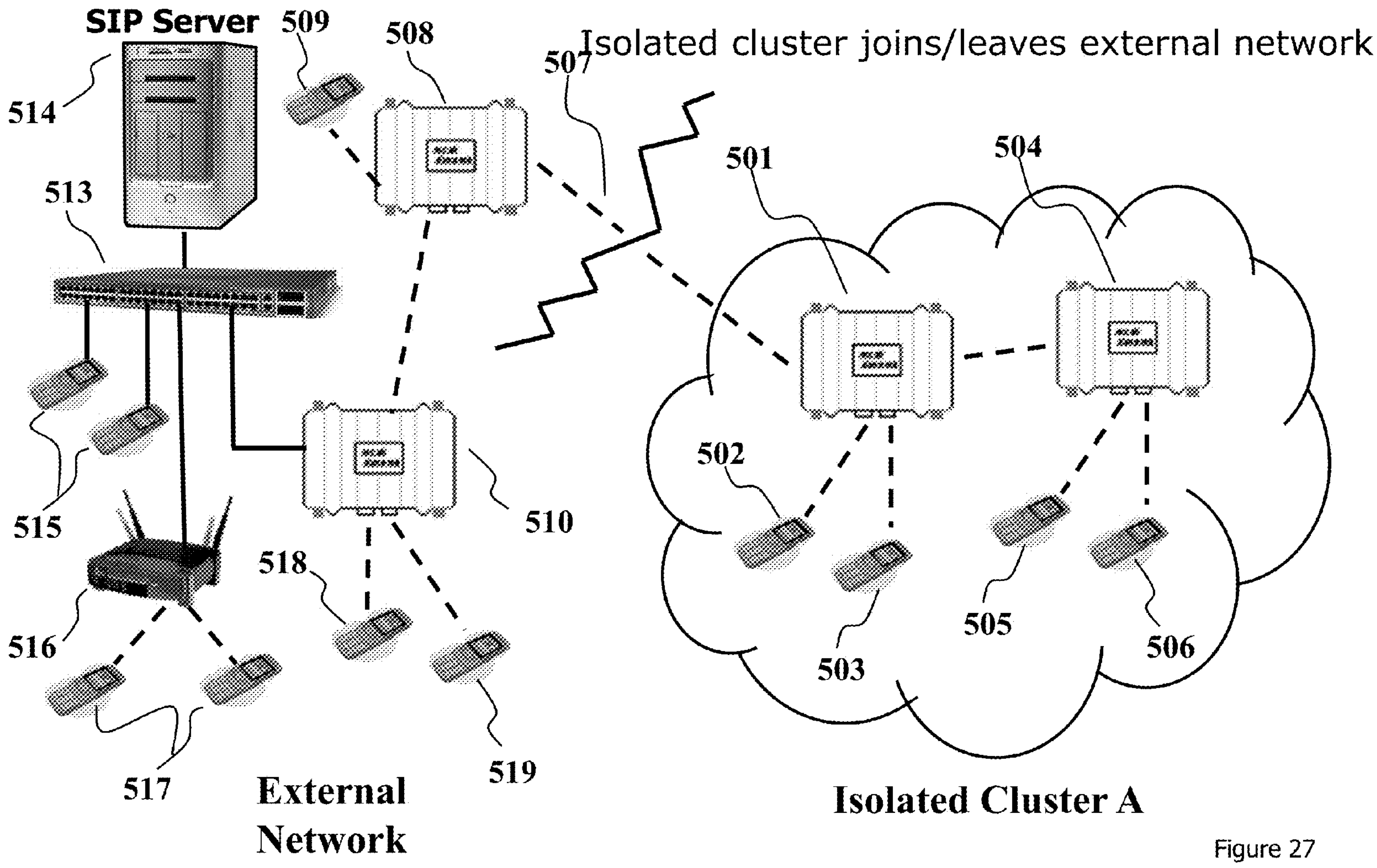
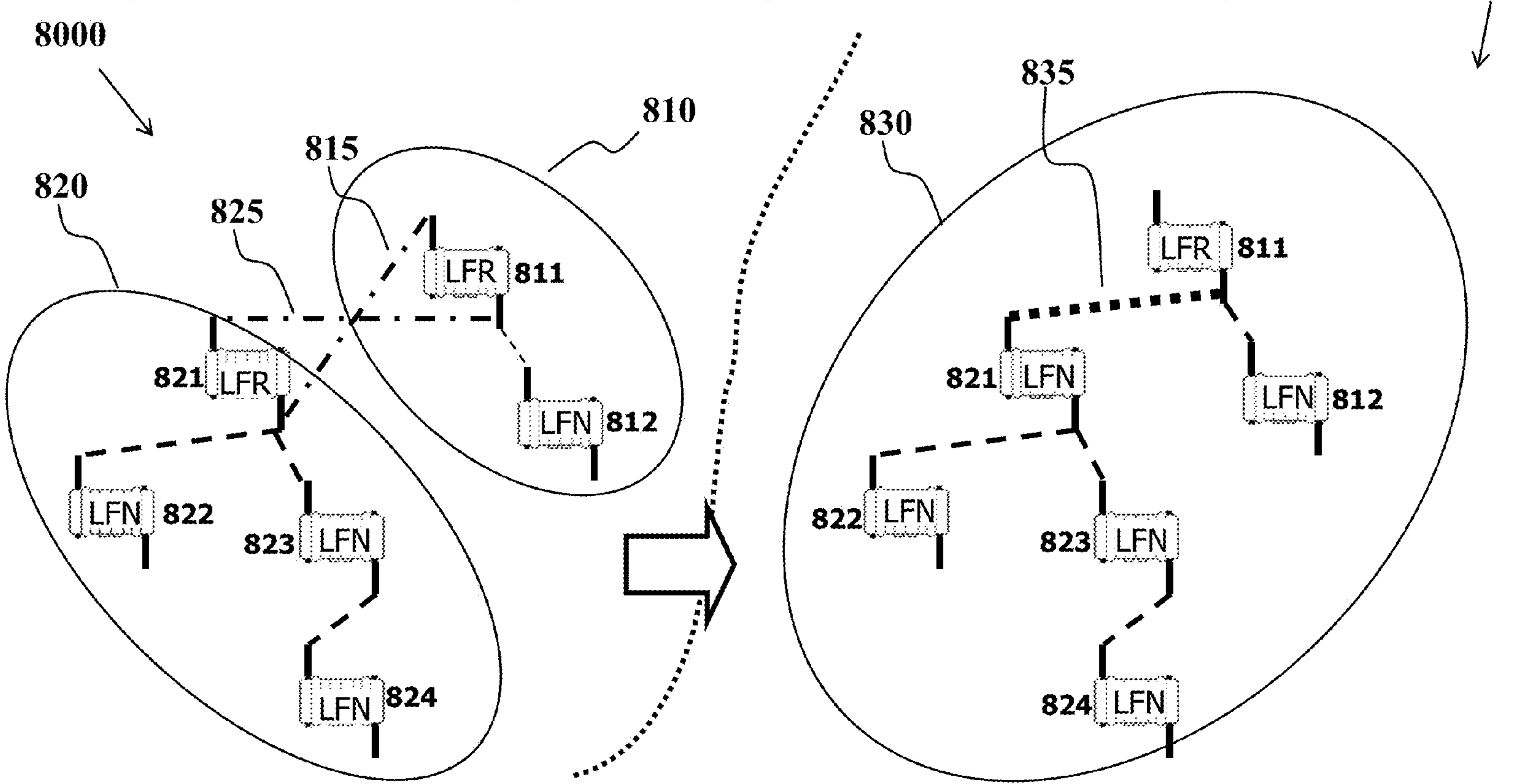

Isolated cluster joins/leaves external network
SIP Server
509
508
507
514
513
501
504
515
510
502
518
516
503
505
506
517
519
External Network
Isolated Cluster A

Figure 27

Floating Network Sub Tree Merges with Floating Network Sub Tree Per Policy Directives



- When two LFRs see each other, one becomes LFN and joins the other
- A Tie Breaker Function (TBF) decides which one becomes an LFN

Figure 28

DISTRIBUTED DHCP SERVICE WITH INHERENT CONFLICT RESOLUTION

ASSUME A FIXED SUBNET MASK OF *255.0.0.0* IS USED

- **IP ADDRESSES OF THE FORM A.*x.y.k* where**
- **A defines the *CUSTOMER-NETWORK-ID* 16 bit (0 -255)**
- **k is *CLIENT BASED ID* 16 bit (0-255)**
- **x, y are each 16 bit : 2^{32} unique networks each with up 255 clients possible**

POLICY DIRECTIVE: SPLIT UP ADDRESS SPACE BETWEEN STATIC and MOBILE networks

- **Assign 15 bits to distributed DHCP services for floating networks**
- **Let NODES CHOOSE A RANDOM 15-bit NUMBER '*R*'AT STARTUP and**

1. **LET '*M*'BE THE DECIMAL EQUIVALENT OF THE 7-MSBs OF '*R*'**
2. **LET '*N*'THE DECIMAL EQUIVALENT OF THE 8-LSBs OF '*R*'**

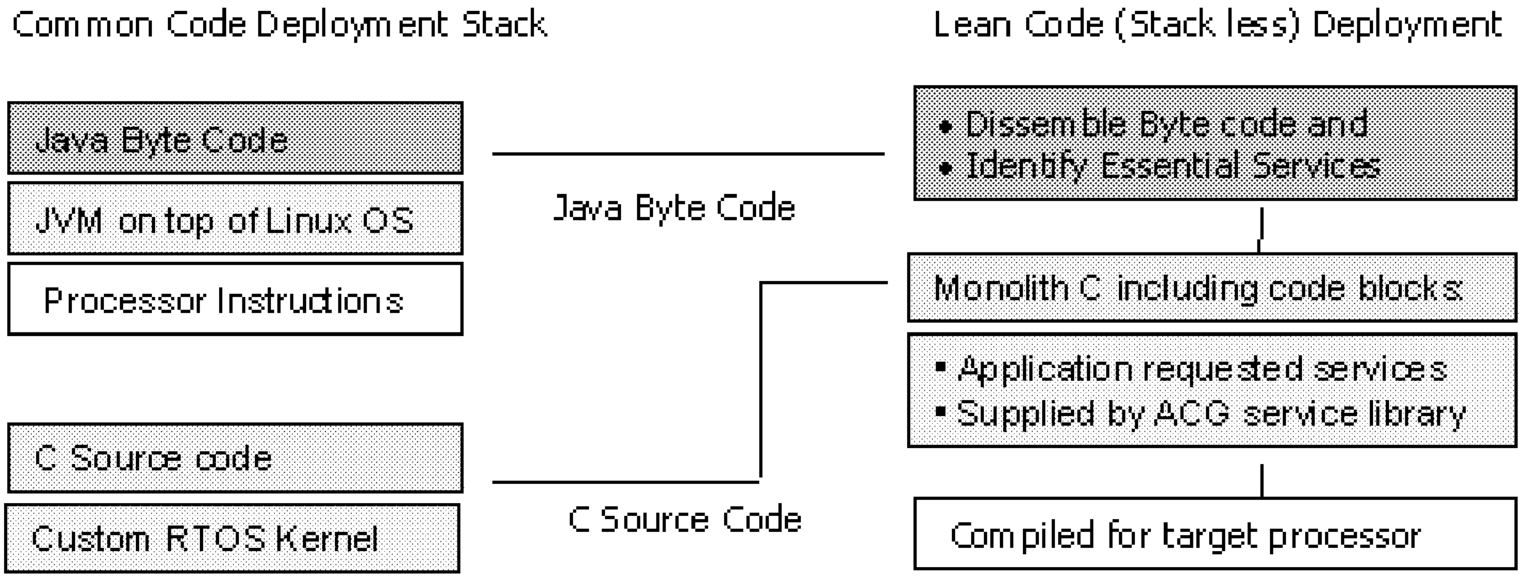
THE DHCP ADDRESS SPACE OF THE NODE WOULD BE

A.255-M.N.0 to A.255-M.N.254 where 0<=M<=127 and 0<=N<=255

Figure 29

Objective: Low foot print for low cost embedded system applications

Key Idea: Remove overhead of OS/Virtual Machine from the equation



Common Code Deployment Stack

Java Byte Code

JVM on top of Linux OS

Processor Instructions

C Source code

Custom RTOS Kernel

Processor Instructions

Java Byte Code

C Source Code

Lean Code (Stack less) Deployment

- Dissemble Byte code and
- Identify Essential Services

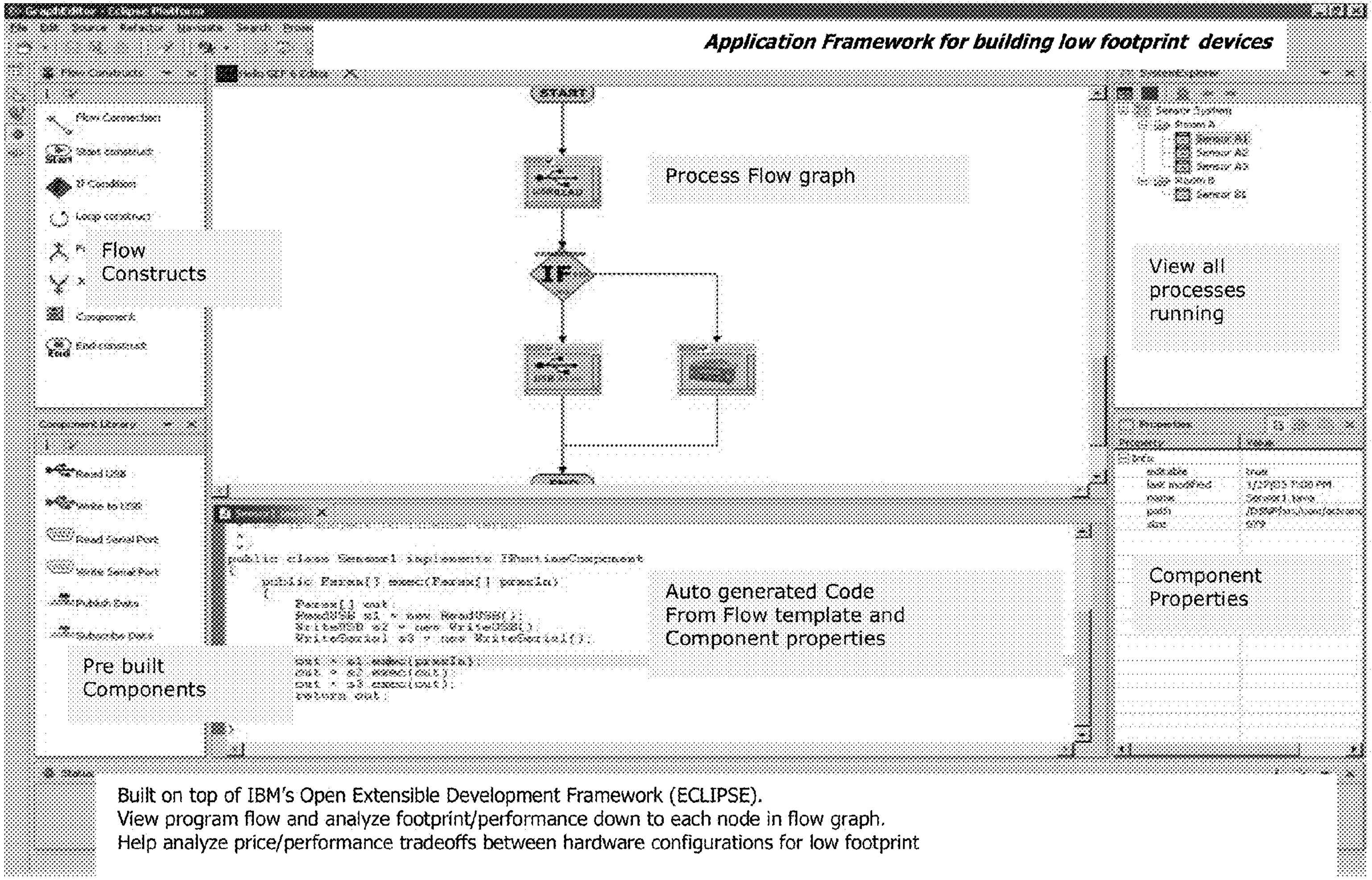
Monolith C including code blocks

- Application requested services
- Supplied by ACG service library

Compiled for target processor

Soft "chip": OS-less compact code

Figure 30



```
public class Sensor1 implements IRuntimeComponent
{
    public Param[] exec(Param[] params)
    {
        Param[] out;
        ReadUSB s1 = new ReadUSB();
        WriteUSB s2 = new WriteUSB();
        WriteSerial s3 = new WriteSerial();

        out = s1.exec(params);
        out = s2.exec(out);
        out = s3.exec(out);
        return out;
```

Property	Value
Info	
editable	true
last modified	[illegible]
name	Sensor1.java
path	[illegible]
size	[illegible]

Built on top of IBM's Open Extensible Development Framework (ECLIPSE).
View program flow and analyze footprint/performance down to each node in flow graph.
Help analyze price/performance tradeoffs between hardware configurations for low footprint

Figure 31

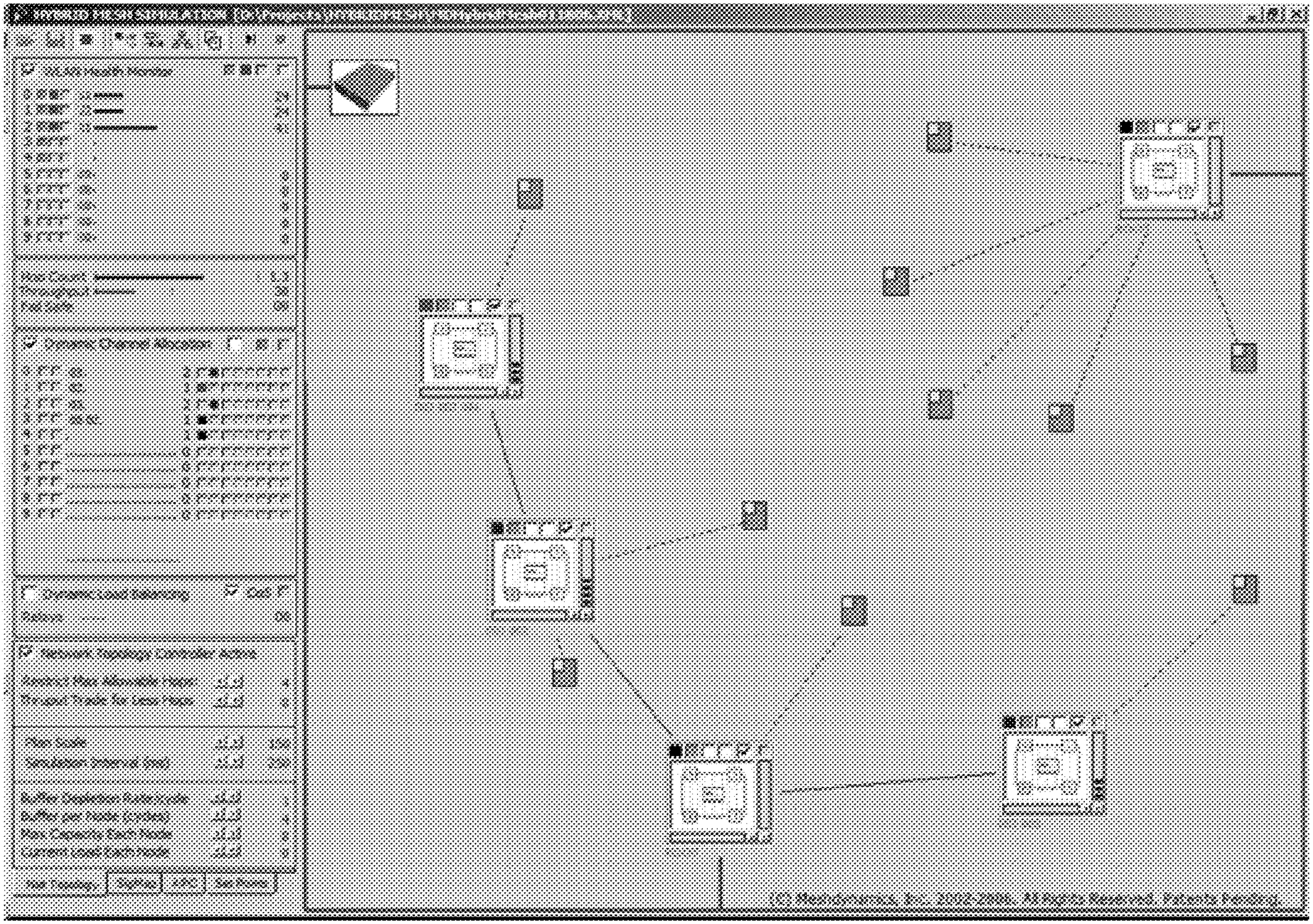

Dynamic Channel Allocation
Hop Count
Throughput
Dynamic Load Balancing
Network Topology Controller Active
Restrict Max Allowable Hops
Throput Trade for Less Hops
Plan Scale
Simulation Interval (sec)
Buffer Depletion Rate/cycle
Buffer per Node (cycles)
Max Capacity Each Node
Current Load Each Node
Net Topology
(C) Meshdynamics, Inc. 2002-2006. All Rights Reserved. Patents Pending.

Figure 32

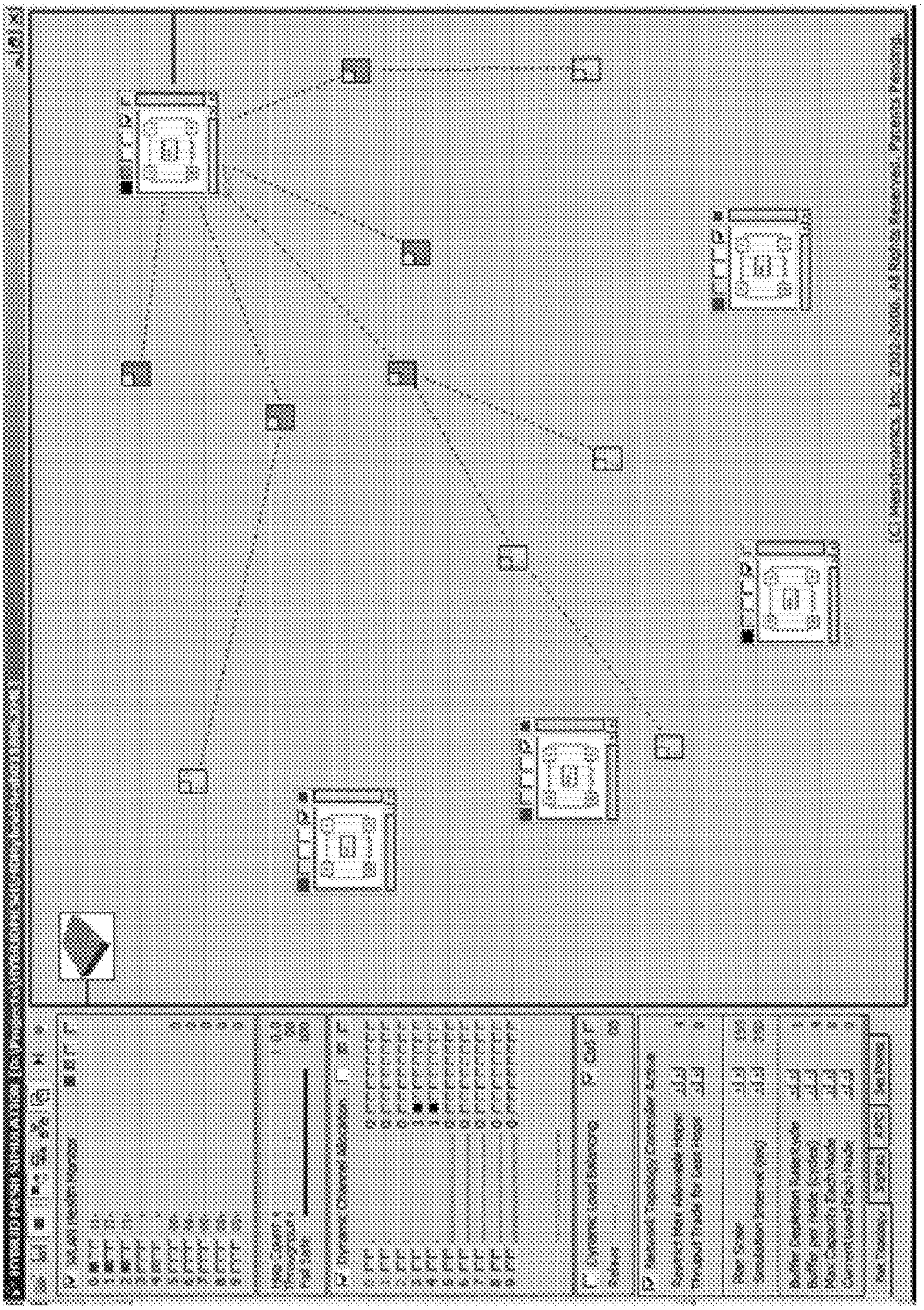


Figure 33

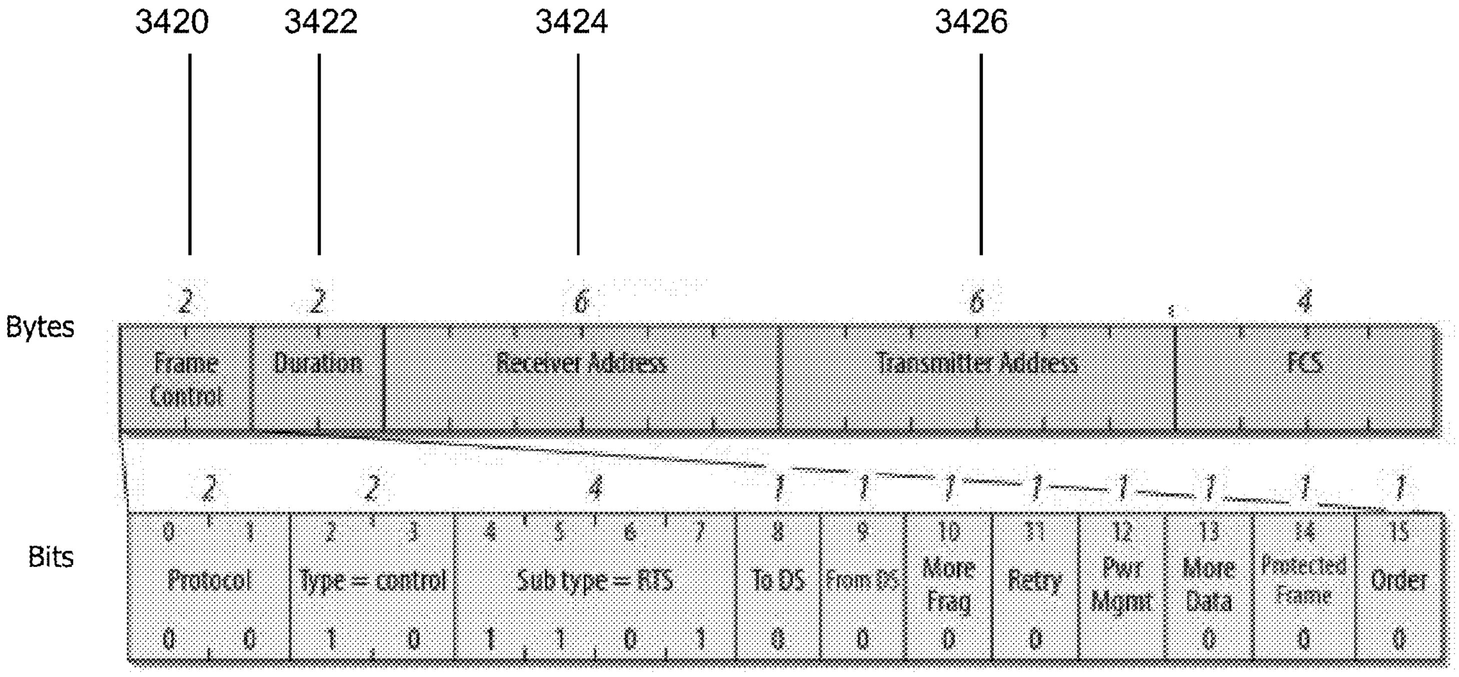

3420
3422
3424
3426
Bytes
2
2
6
6
4
Frame Control
Duration
Receiver Address
Transmitter Address
FCS
Bits
2
2
4
1
1
1
1
1
1
1
1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Protocol
Type = control
Sub type = RTS
To DS
From DS
More Frag
Retry
Pwr Mgmt
More Data
Protected Frame
Order
0
0
1
0
1
1
0
1
0
0
0
0
0
0
0

Figure 34

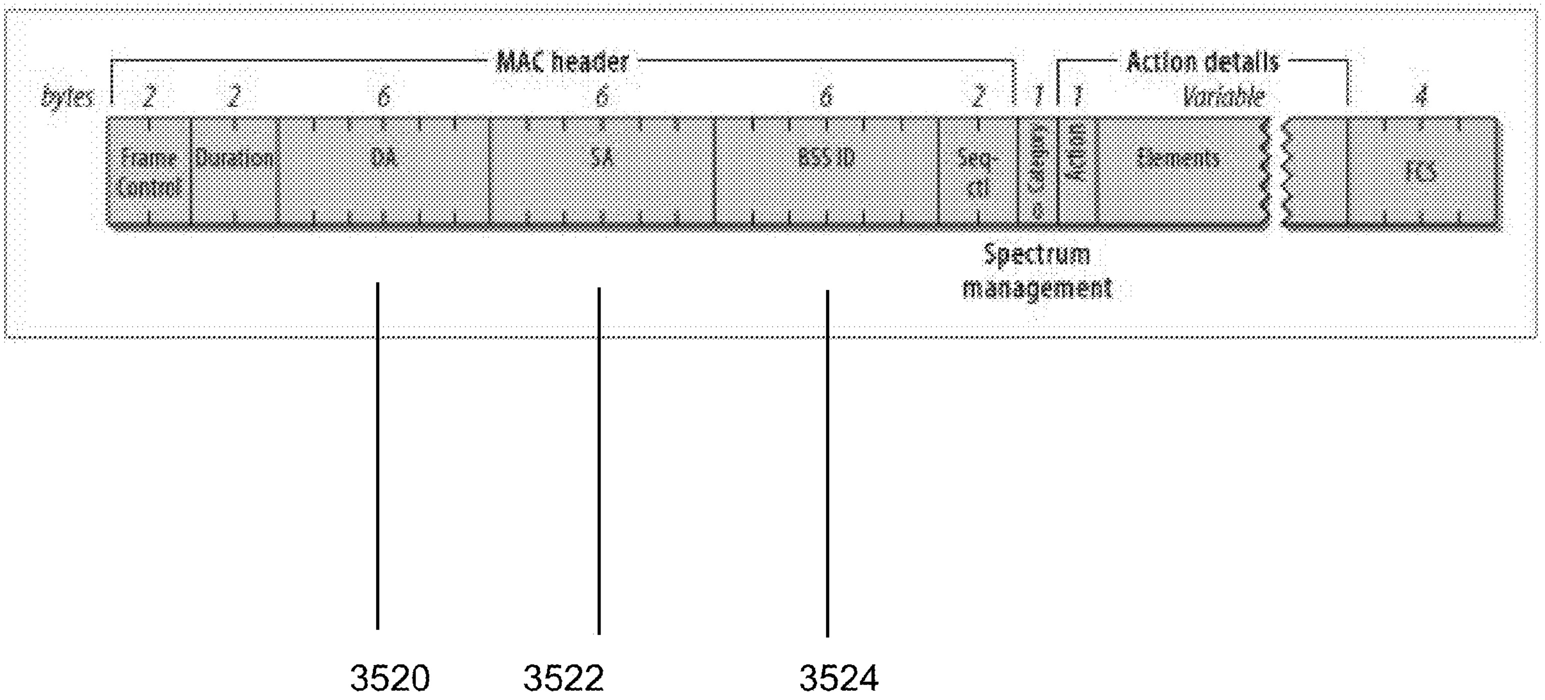

MAC header
Action details
bytes
2
2
6
6
6
2
1
1
Variable
4
Frame Control
Duration
DA
SA
BSS ID
Seq-ctl
Category = 0
Action
Elements
FCS
Spectrum management
3520
3522
3524

Figure 35

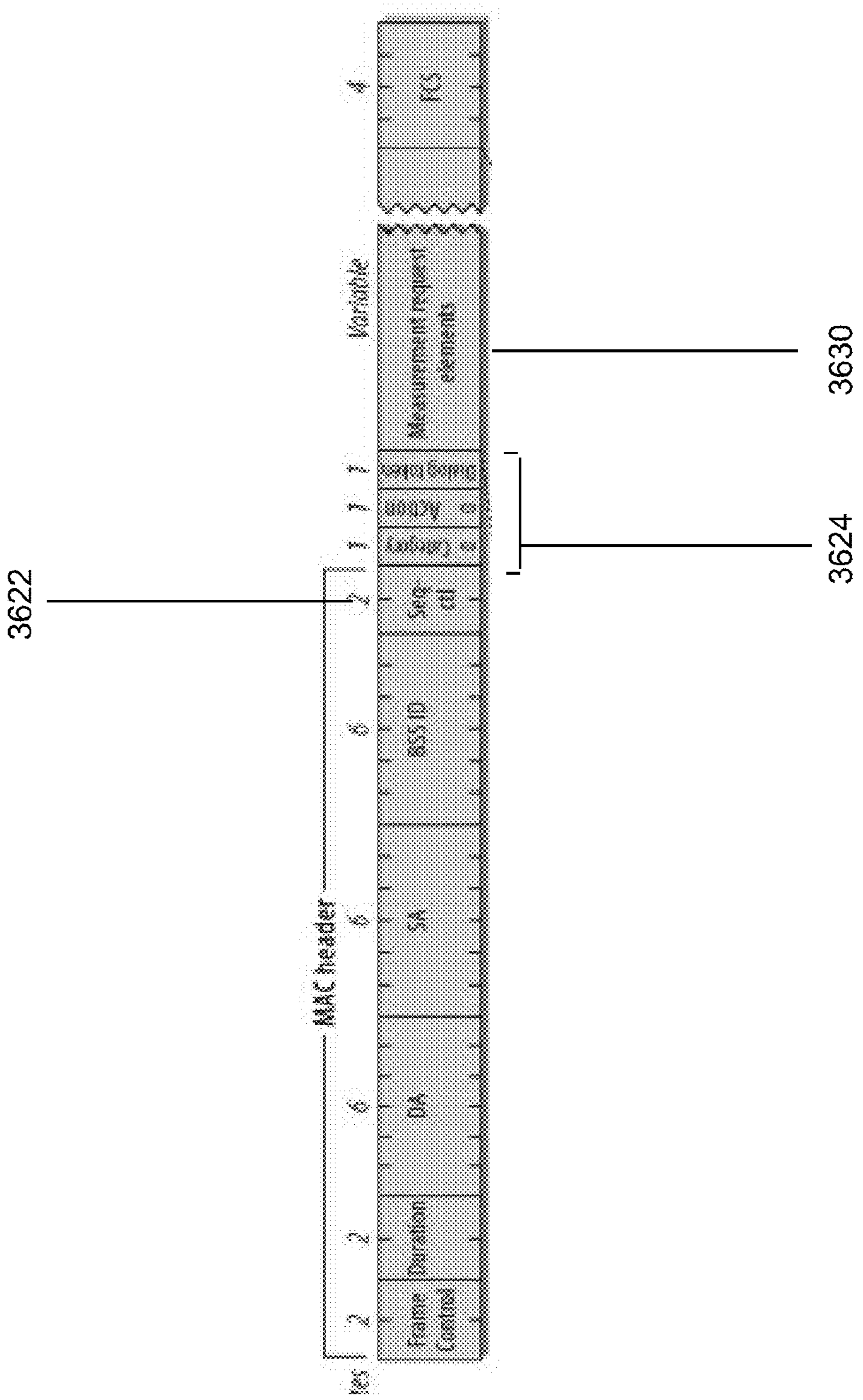

Bytes
MAC header
2
Frame Control
2
Duration
6
DA
6
SA
6
BSS ID
2
Seq ctl
1
Category
1
Action
1
Dialog token
Variable
Measurement request elements
4
FCS
3622
3624
3630

Figure 36

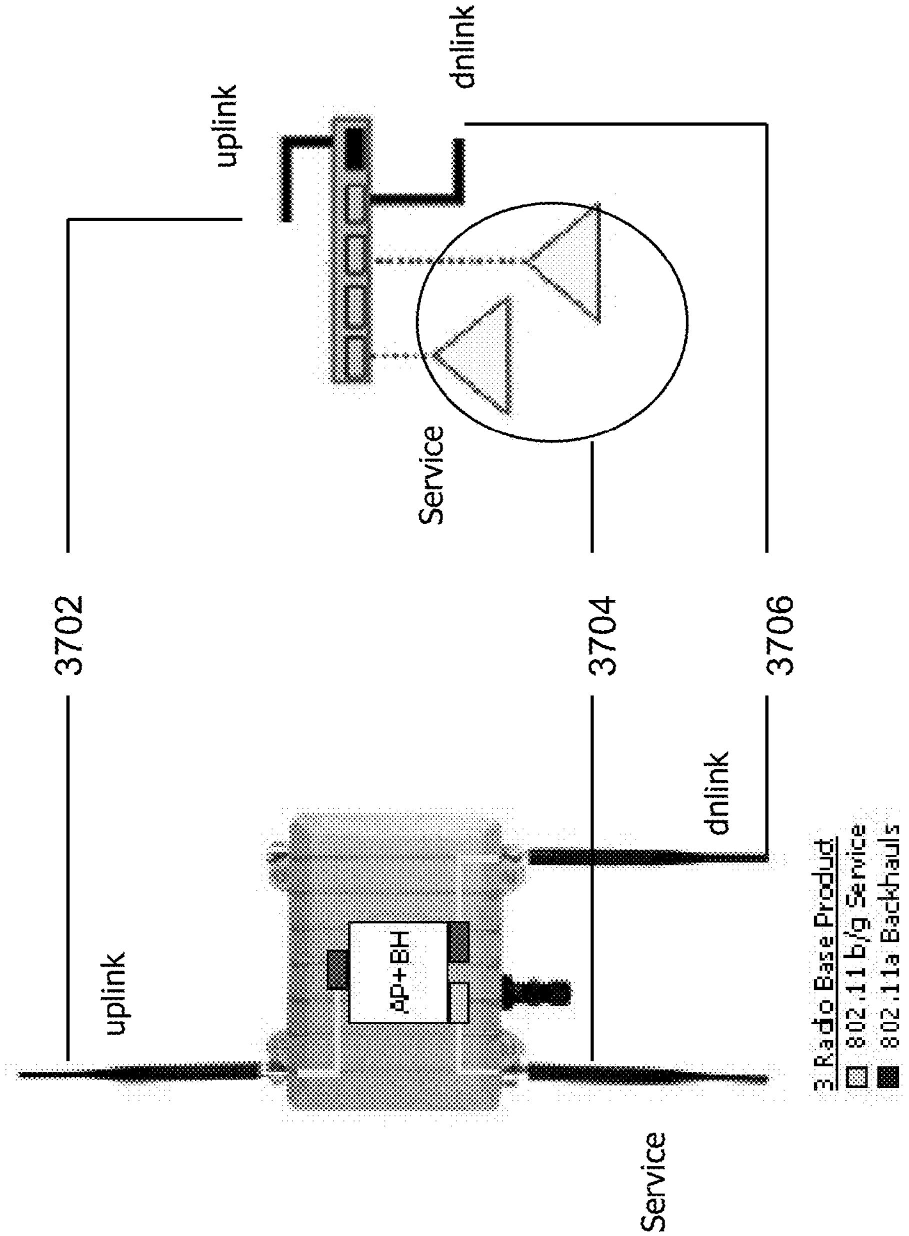

uplink
dnlink
Service
3702
3704
3706
uplink
dnlink
AP+BH
Service
3 Radio Base Product
□ 802.11 b/g Service
■ 802.11a Backhauls

Figure 37

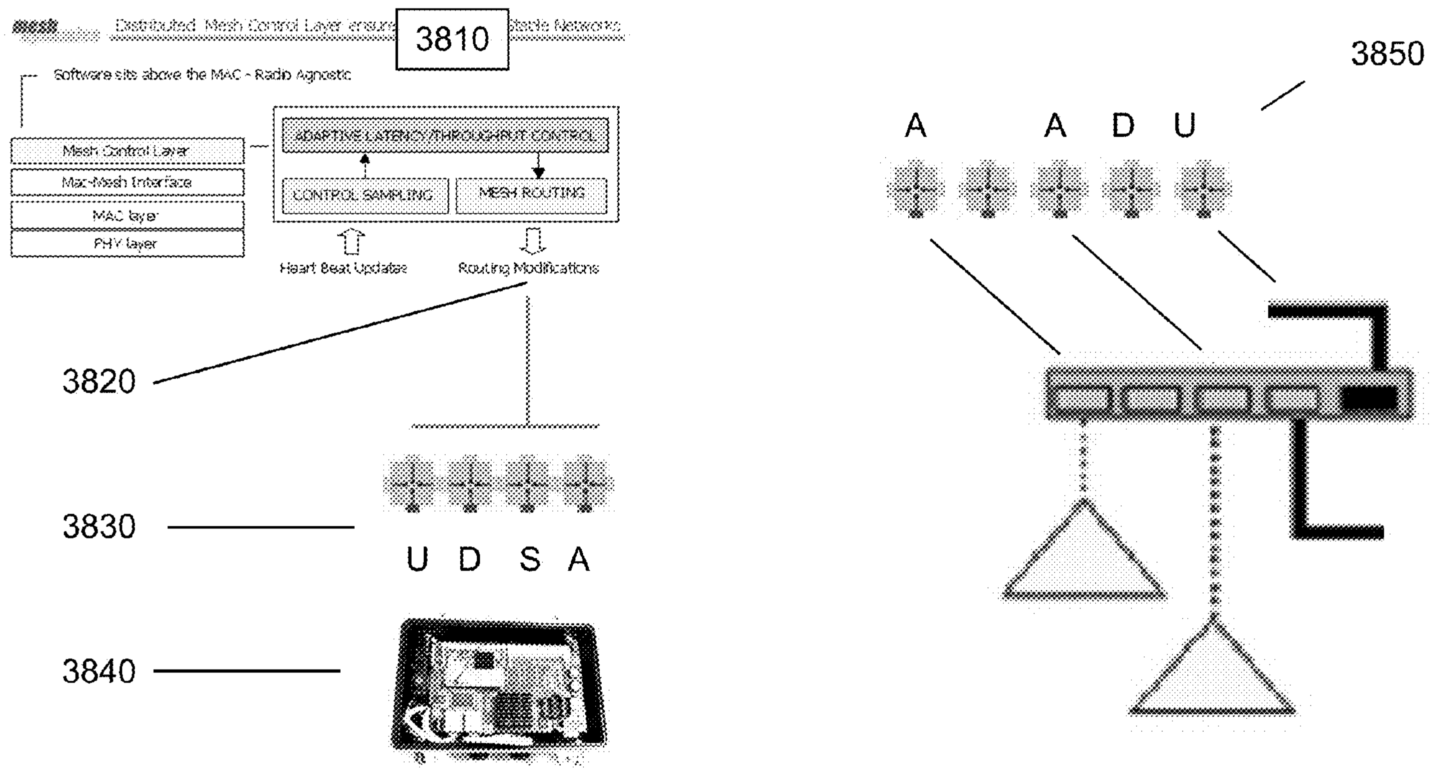

mesh
Distributed Mesh Control Layer ensure
3810
Stable Networks
Software sits above the MAC - Radio Agnostic
Mesh Control Layer
Mac-Mesh Interface
MAC layer
PHY layer
ADAPTIVE LATENCY/THROUGHPUT CONTROL
CONTROL SAMPLING
MESH ROUTING
Heart Beat Updates
Routing Modifications
3820
3830
U D S A
3840
3850
A
A D U

Figure 38

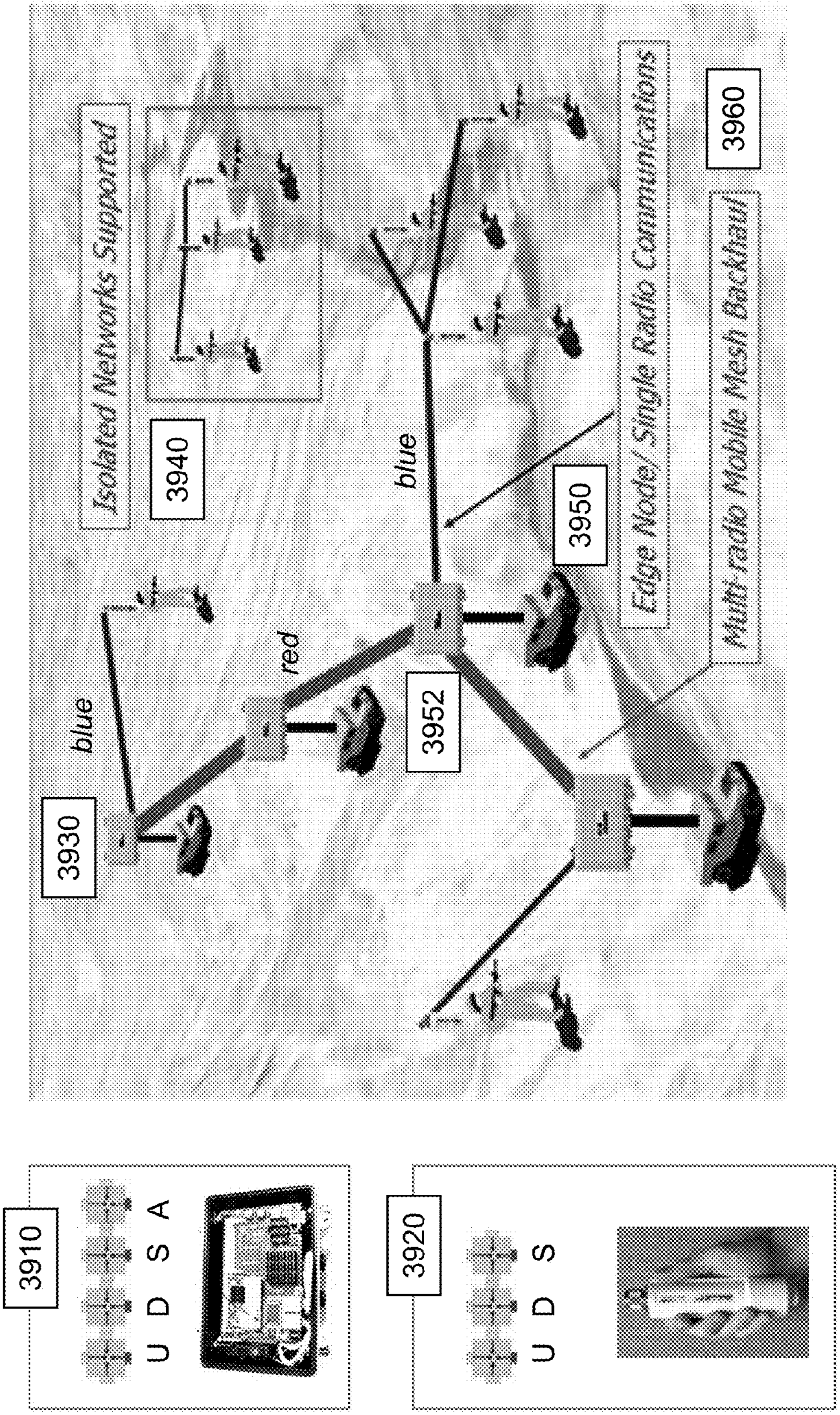

Isolated Networks Supported
3940
blue
red
blue
3930
3952
3950
Edge Node/ Single Radio Communications
Multi-radio Mobile Mesh Backhaul
3960
3910
U D S A
3920
U D S

Figure 39

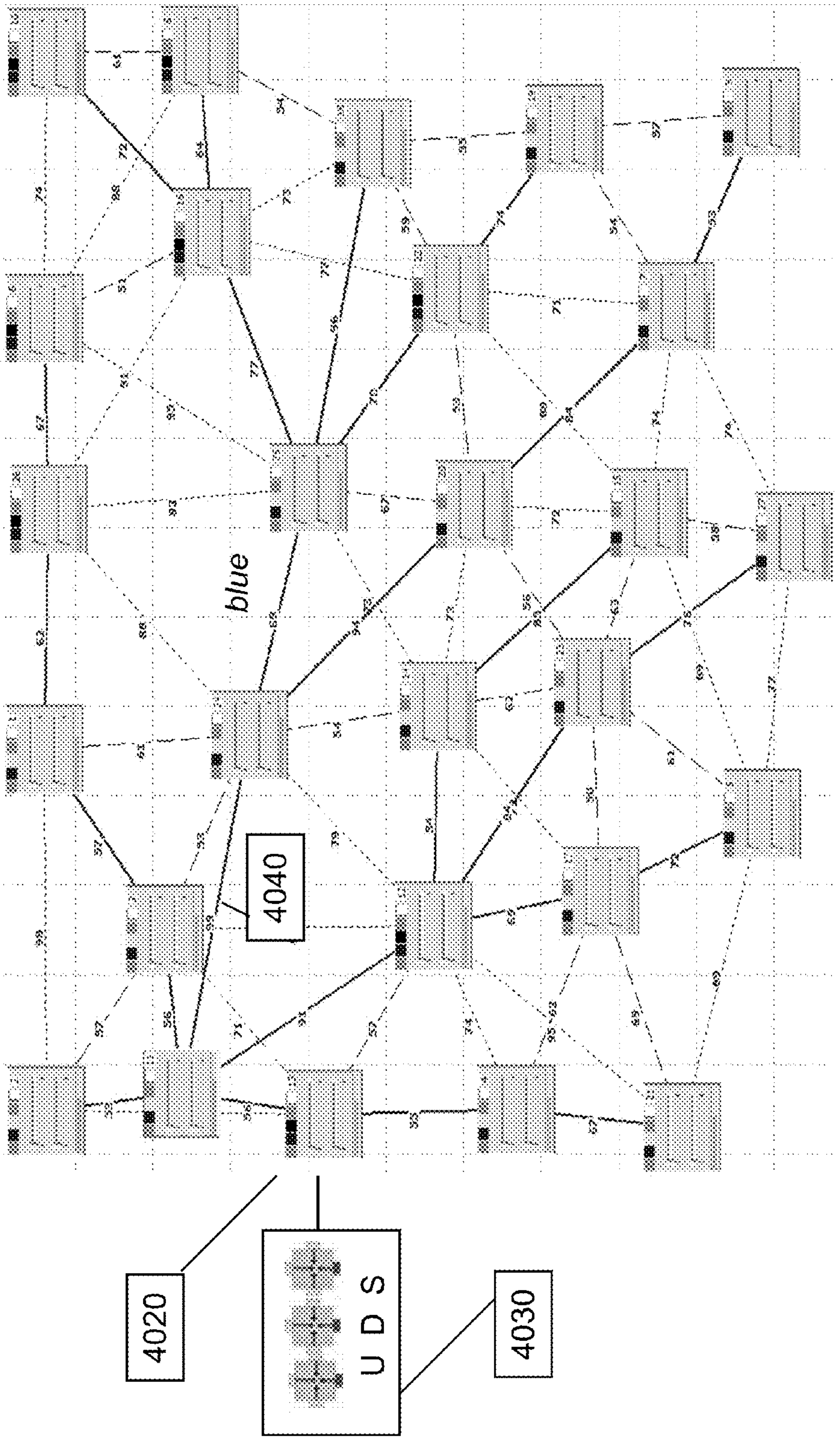


Figure 40

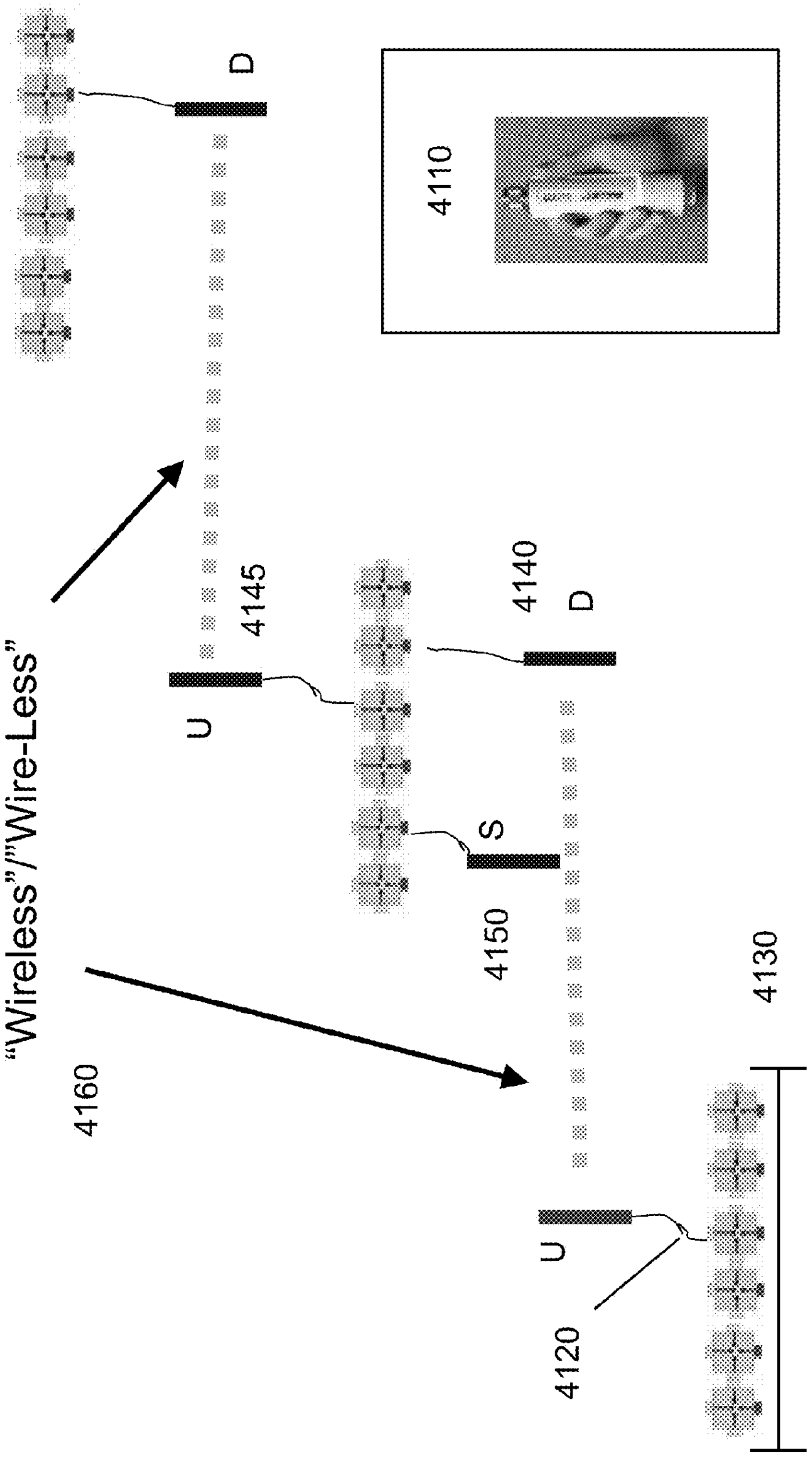


Figure 41

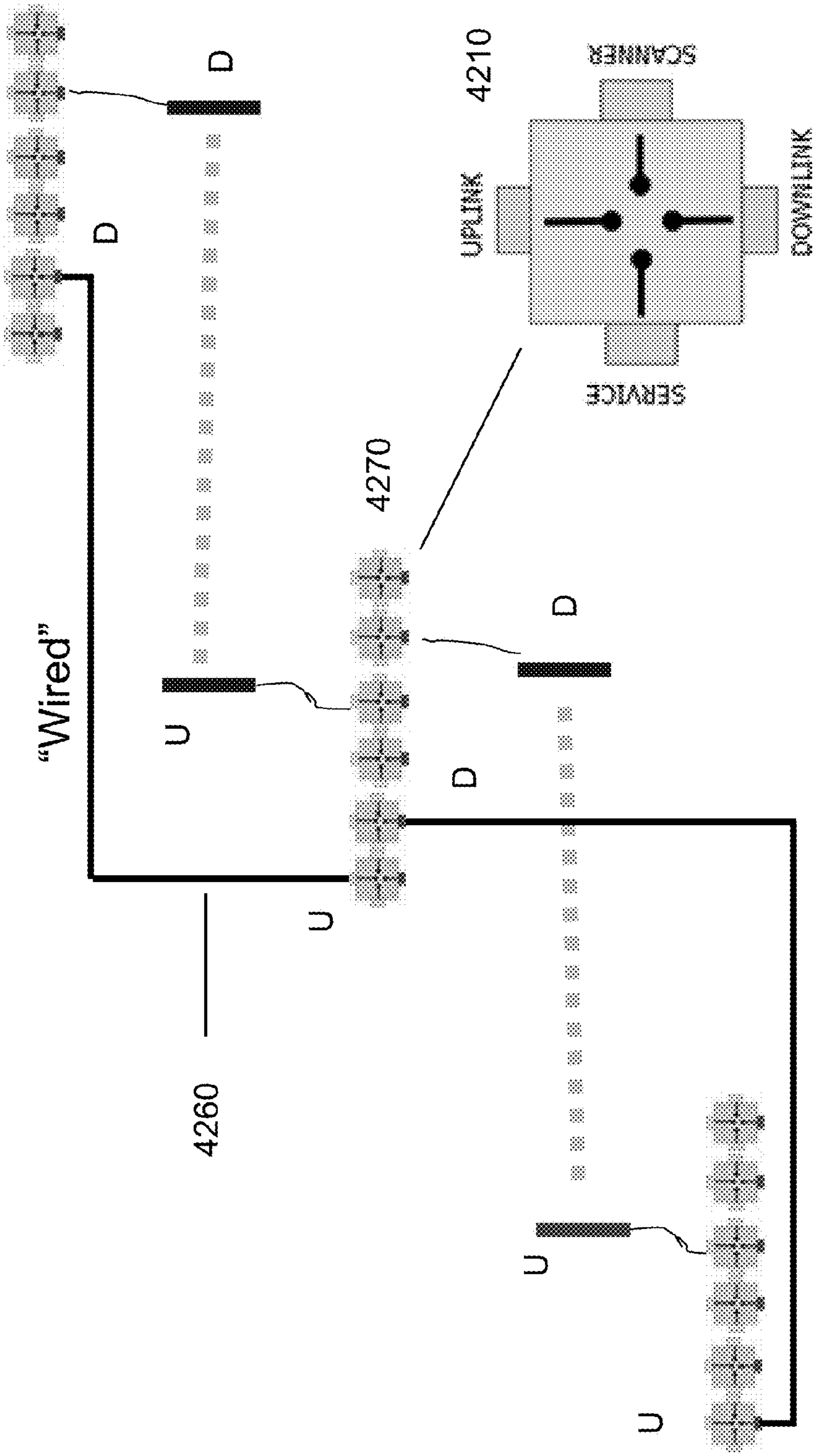

4210
SCANNER
UPLINK
DOWNLINK
SERVICE
4270
4260
"Wired"
U
D
U
D
U
D
U
D
U

Figure 42

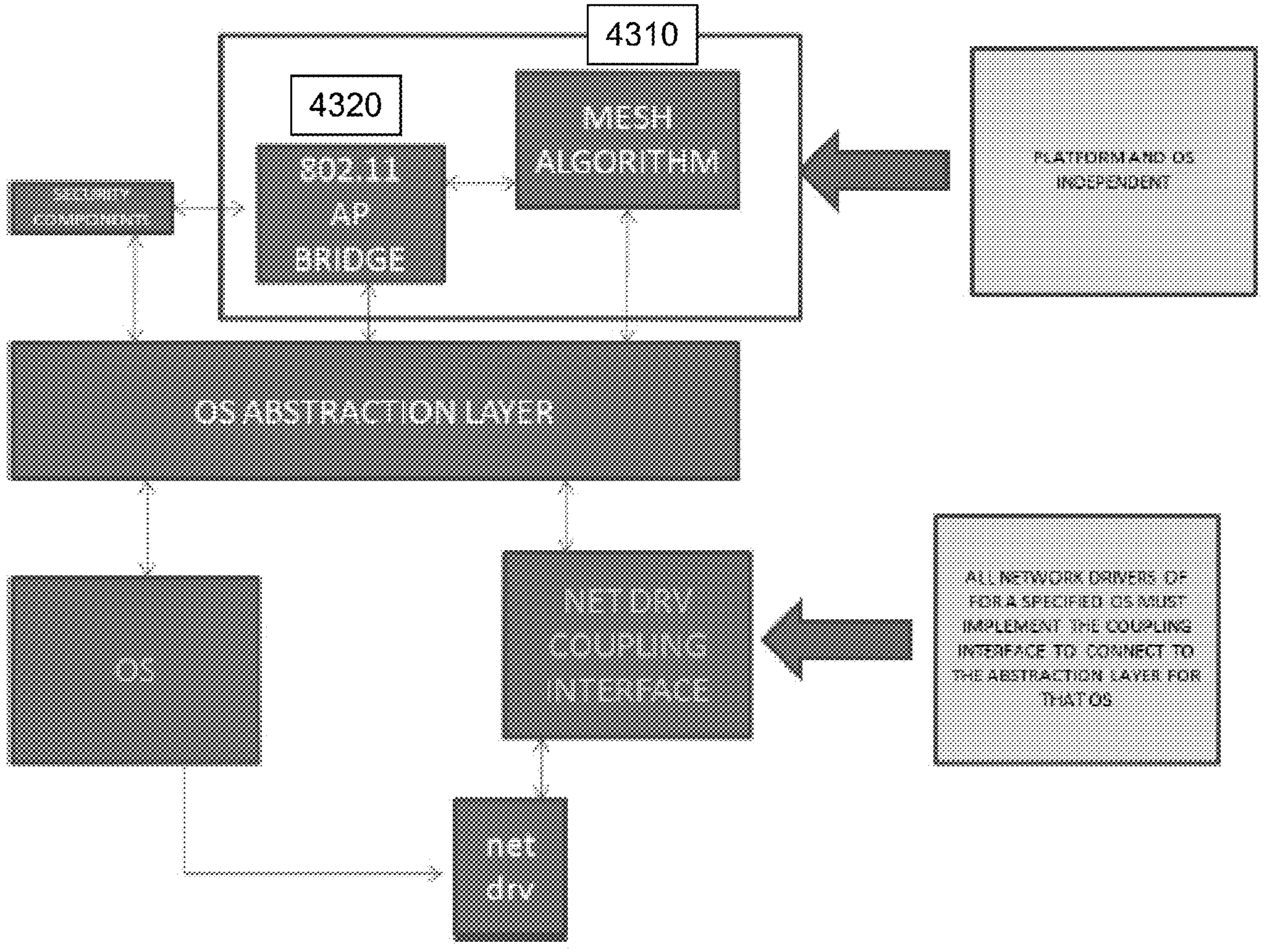

4310
4320
SECURITY COMPONENTS
802.11 AP BRIDGE
MESH ALGORITHM
PLATFORM AND OS INDEPENDENT
OS ABSTRACTION LAYER
OS
NET DRV COUPLING INTERFACE
ALL NETWORK DRIVERS OF FOR A SPECIFIED OS MUST IMPLEMENT THE COUPLING INTERFACE TO CONNECT TO THE ABSTRACTION LAYER FOR THAT OS
net drv

Figure 43

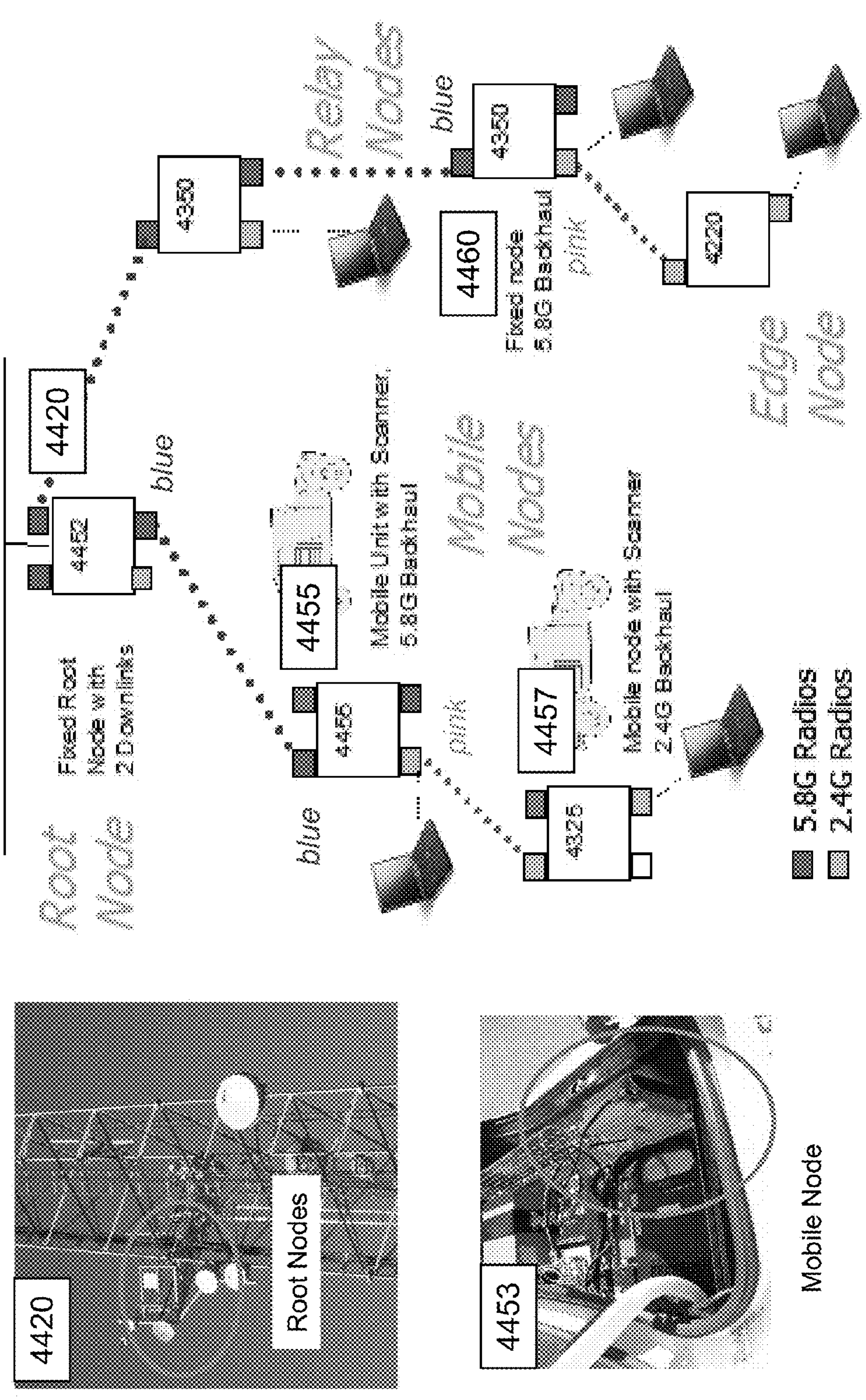

Root Node
Fixed Root Node with 2 Downlinks
4452
4420
blue
4350
Relay Nodes
blue
4350
4460
Fixed node 5.8G Backhaul
pink
4220
Edge Node
4455
Mobile Unit with Scanner. 5.8G Backhaul
Mobile Nodes
4456
pink
blue
4457
Mobile node with Scanner 2.4G Backhaul
4326
▨ 5.8G Radios
▨ 2.4G Radios
4420
Root Nodes
4453
Mobile Node

Figure 44

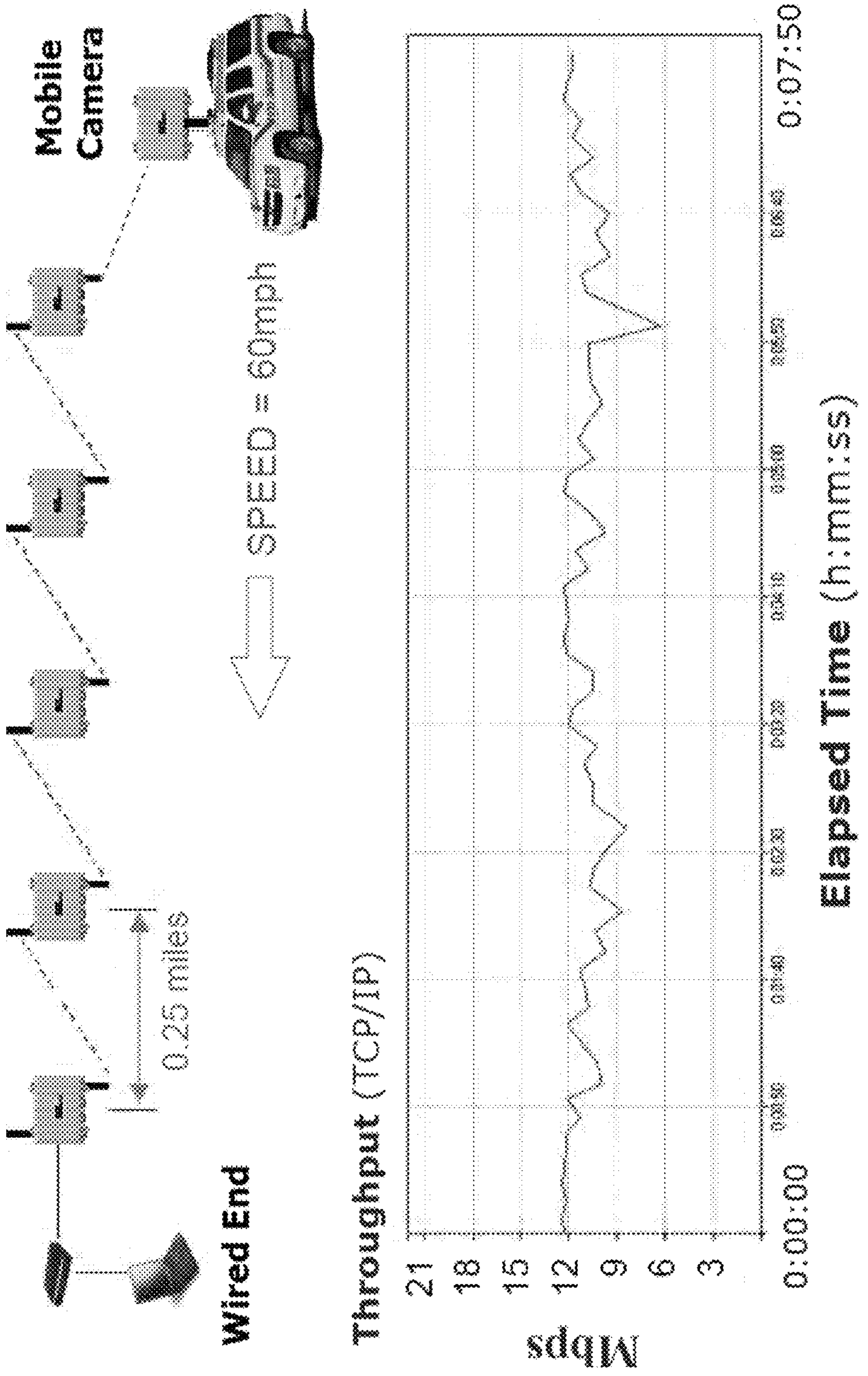

Mobile Camera
Wired End
0.25 miles
SPEED = 60mph
Throughput (TCP/IP)
Mbps
21
18
15
12
9
6
3
0:00:00
0:07:50
Elapsed Time (h:mm:ss)

Figure 45

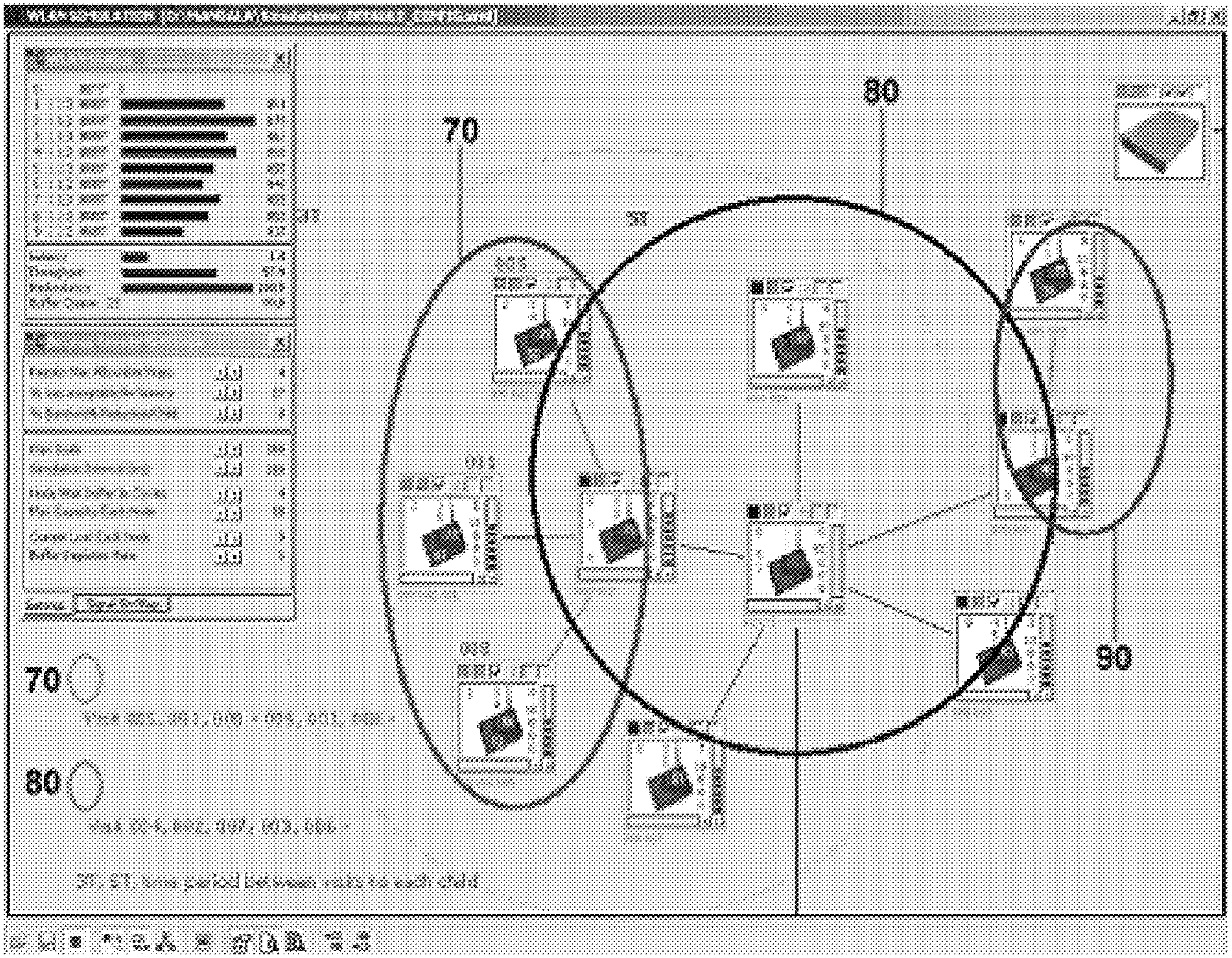
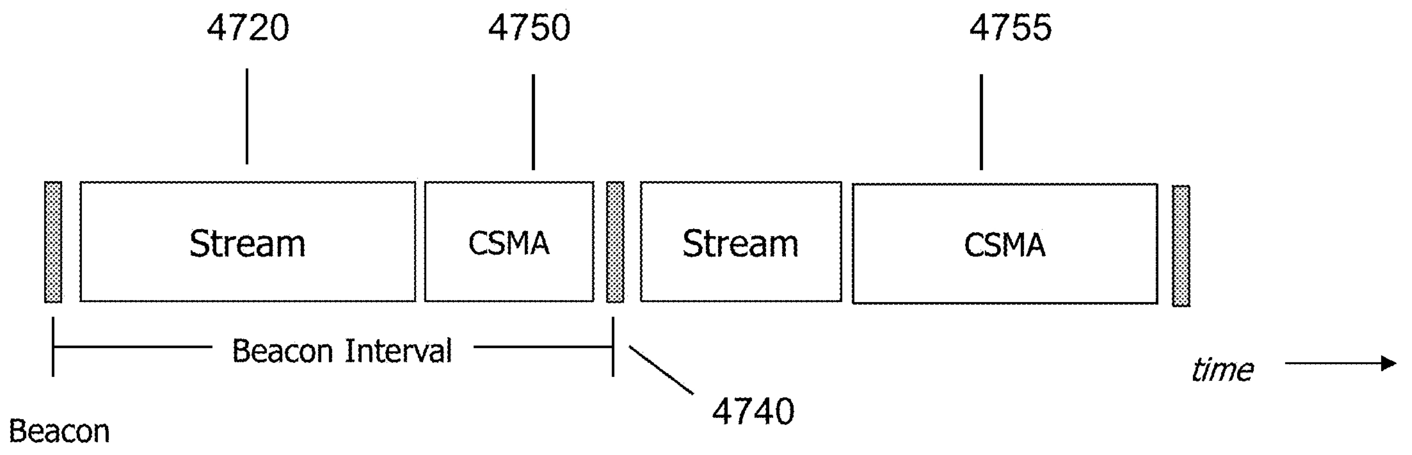

70
80
90
ST
70
Visit 005, 001, 009 + 008, 011, 006 >
80
Visit 004, 002, 007, 003, 006 >
ST, ST: time period between visits to each child

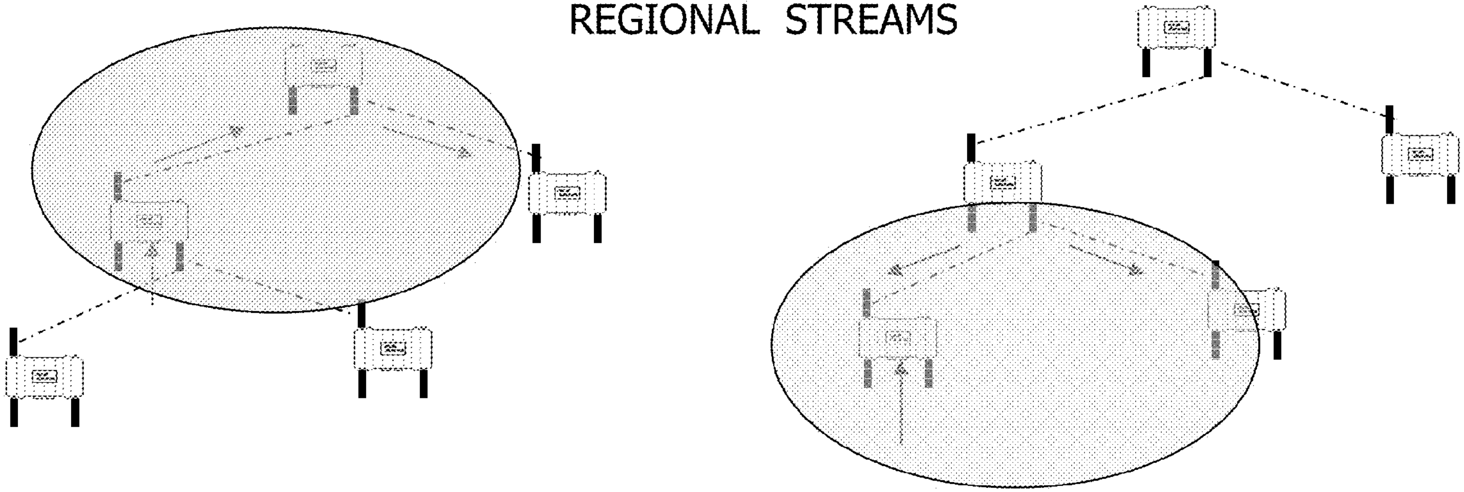
Figure 46



- CLIENTS SILENT IN ALLOCATED TIME SLOT FOR STREAMING.
- CSMA TIME SLOT IS ADJUSTABLE AFTER STREAM TIME SLOT ALLOCATIONN

Figure 47

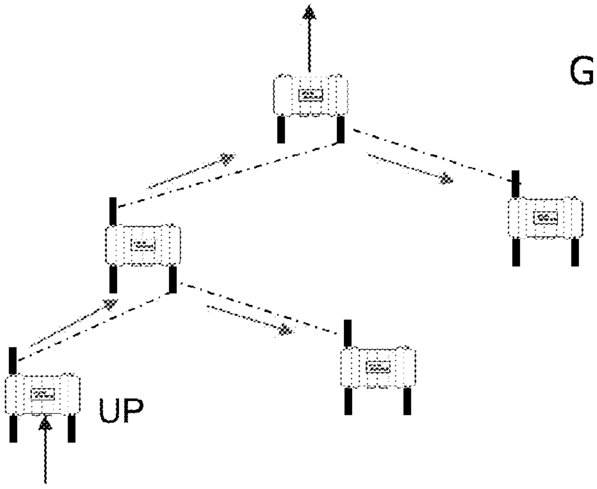
REGIONAL STREAMS

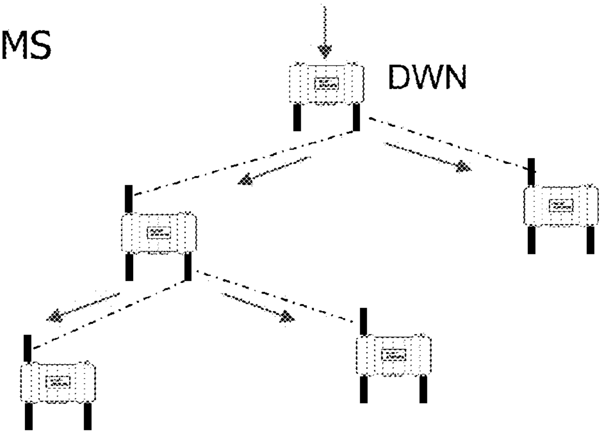


- REGIONAL STREAMS TRAVERSE ONLY THE SPECIFIED DIRECTIONS AND/OR REGION
  - REGIONS CAN BE GPS BOUNDARIES, OR NUMBER OF HOPS / SUBTREES
  - DIRECTIONS CAN BE a) UP OR DOWN OR b) TURN-BY-TURN etc.
  - RELEVANT FOR DATA IMPORTANT LOCALLY ONLY (e.g. Sibling Devices)
  - DOES NOT AFFECT BACKHAUL BANDWIDTH OUTSIDE SPECIFIED REGION

Figure 48

GLOBAL STREAMS





- GLOBALS STREAMS TRAVERSE THE ENTIRE NETWORK
  - AT THE MINIMUM GOES UP TO THE LOGICAL "ROOT"/PARENT/AGENT
  - THEY CAN ORIGINATE FROM ANY WHERE ON THE NETWORK
    - IF FROM ROOT: DOWNWARD ONLY (worst case).
    - IF FROM NODE : UPWARDS AND DOWNWARDS (worst case).
    - IF FROM CLIENT : UPWARDS AND DOWNWARDS, includes Siblings (worst case)

Figure 49

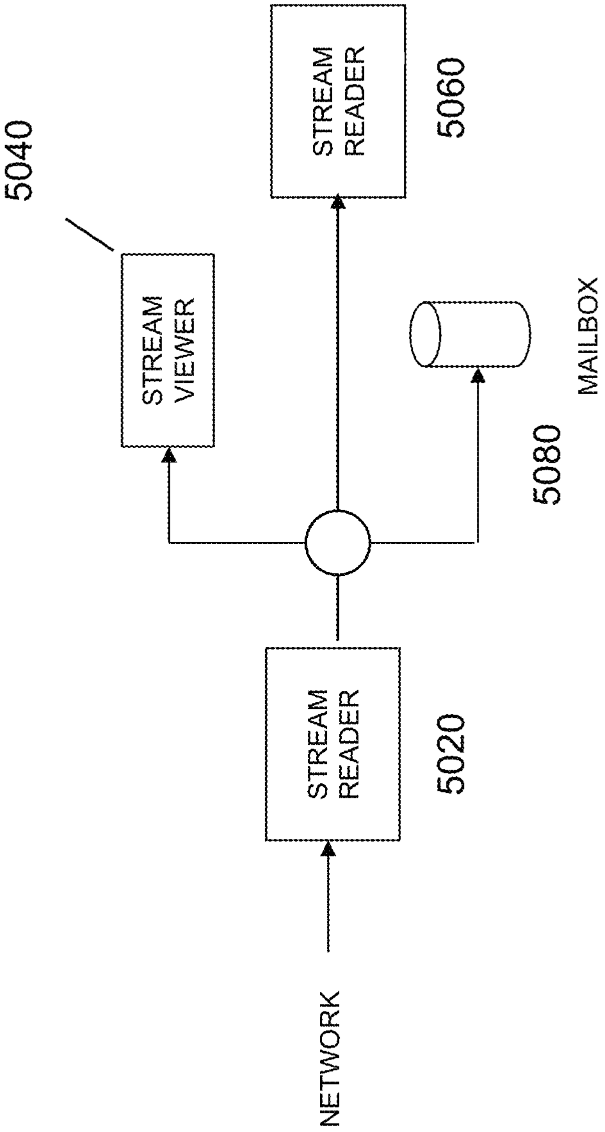

NETWORK
STREAM READER
5020
STREAM VIEWER
5040
STREAM READER
5060
MAILBOX
5080

Figure 50

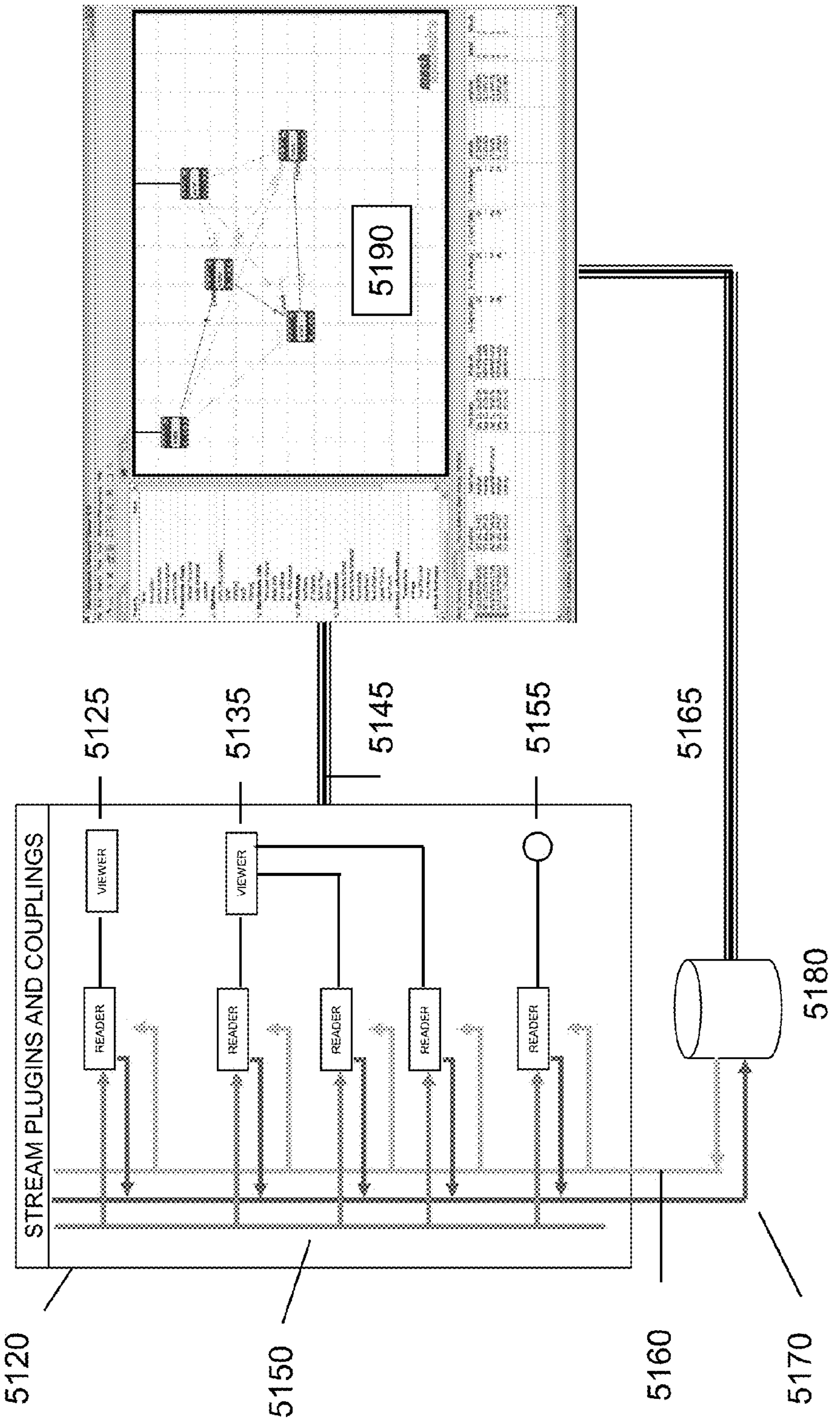

STREAM PLUGINS AND COUPLINGS
VIEWER
VIEWER
READER
READER
READER
READER
READER
5120
5125
5135
5145
5150
5155
5160
5165
5170
5180
5190

Figure 51

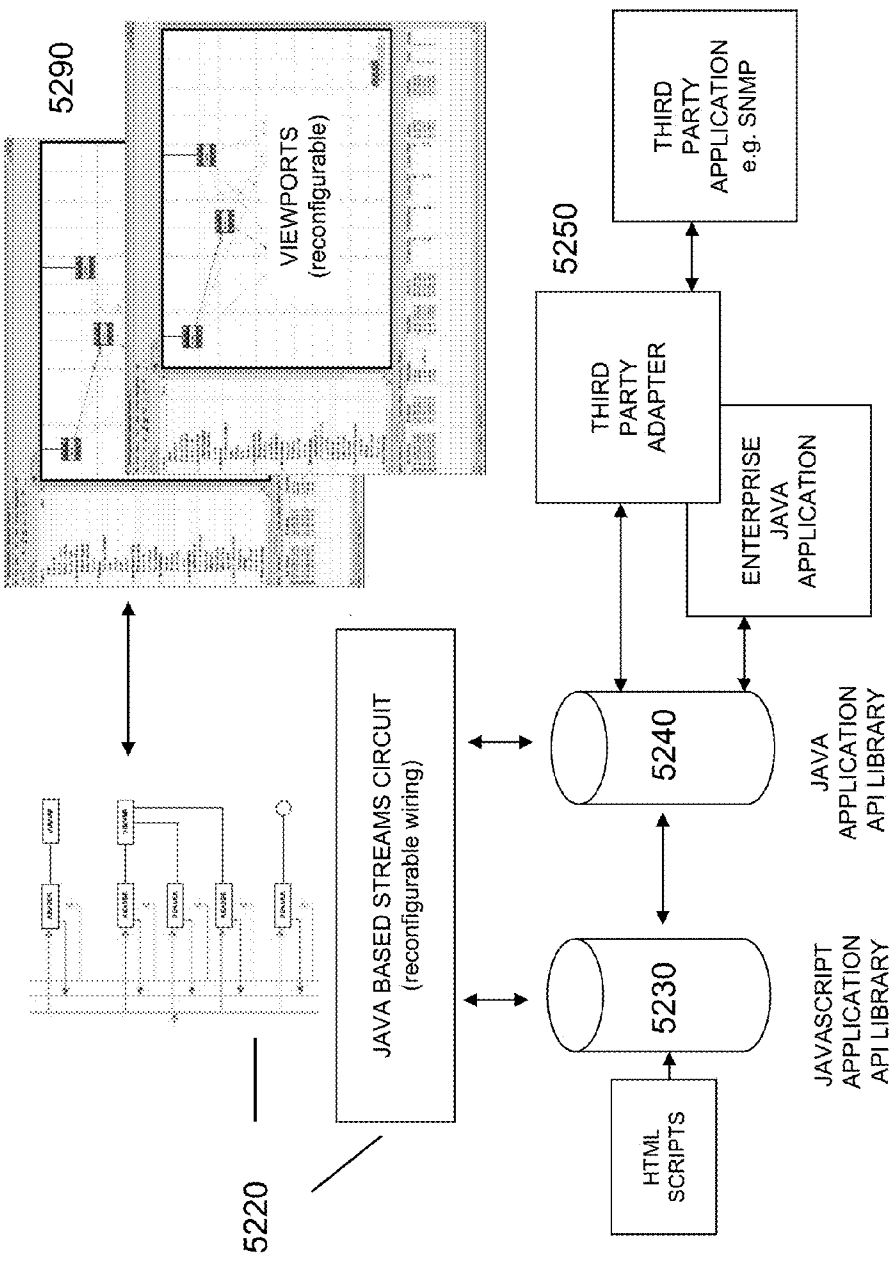
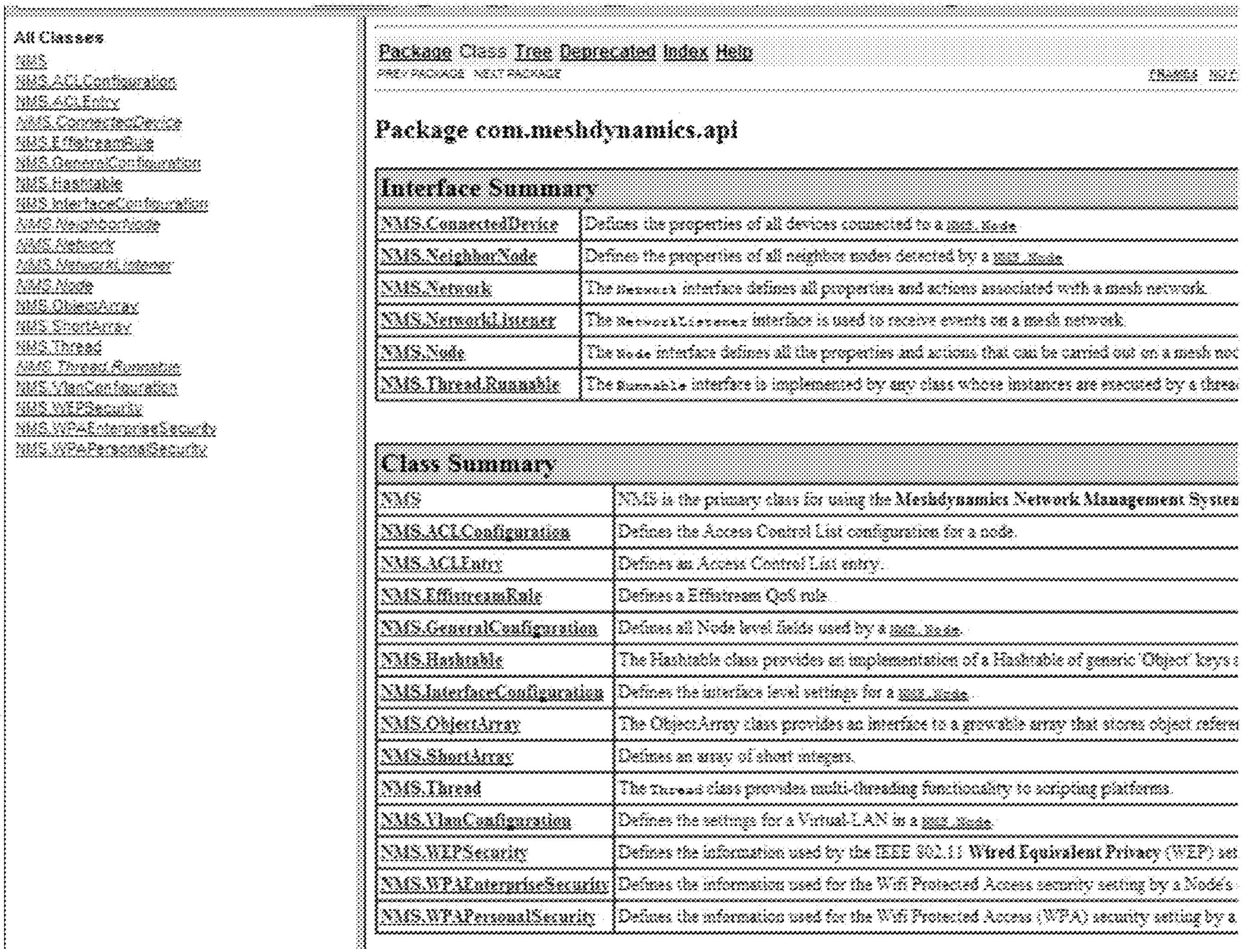

5290
VIEWPORTS
(reconfigurable)
5250
THIRD PARTY APPLICATION e.g. SNMP
THIRD PARTY ADAPTER
ENTERPRISE JAVA APPLICATION
5220
JAVA BASED STREAMS CIRCUIT
(reconfigurable wiring)
5240
JAVA APPLICATION API LIBRARY
5230
JAVASCRIPT APPLICATION API LIBRARY
HTML SCRIPTS

Figure 52



All Classes

NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguration
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecurity
NMS.WPAPersonalSecurity

Package Class Tree Deprecated Index Help
PREV PACKAGE NEXT PACKAGE
FRAMES NO F

Package com.meshdynamics.api

Interface Summary	
NMS.ConnectedDevice	Defines the properties of all devices connected to a NMS.Node
NMS.NeighborNode	Defines the properties of all neighbor nodes detected by a NMS.Node
NMS.Network	The Network interface defines all properties and actions associated with a mesh network.
NMS.NetworkListener	The NetworkListener interface is used to receive events on a mesh network.
NMS.Node	The Node interface defines all the properties and actions that can be carried out on a mesh noc
NMS.Thread.Runnable	The Runnable interface is implemented by any class whose instances are executed by a threa

Class Summary	
NMS	NMS is the primary class for using the **Meshdynamics Network Management Systen**
NMS.ACLConfiguration	Defines the Access Control List configuration for a node.
NMS.ACLEntry	Defines an Access Control List entry.
NMS.EffistreamRule	Defines a Effistream QoS rule.
NMS.GeneralConfiguration	Defines all Node level fields used by a NMS.Node.
NMS.Hashtable	The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys a
NMS.InterfaceConfiguration	Defines the interface level settings for a NMS.Node.
NMS.ObjectArray	The ObjectArray class provides an interface to a growable array that stores object referen
NMS.ShortArray	Defines an array of short integers.
NMS.Thread	The Thread class provides multi-threading functionality to scripting platforms.
NMS.VlanConfiguration	Defines the settings for a Virtual-LAN in a NMS.Node.
NMS.WEPSecurity	Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) set
NMS.WPAEnterpriseSecurity	Defines the information used for the Wifi Protected Access security setting by a Node's
NMS.WPAPersonalSecurity	Defines the information used for the Wifi Protected Access (WPA) security setting by a

Figure 53

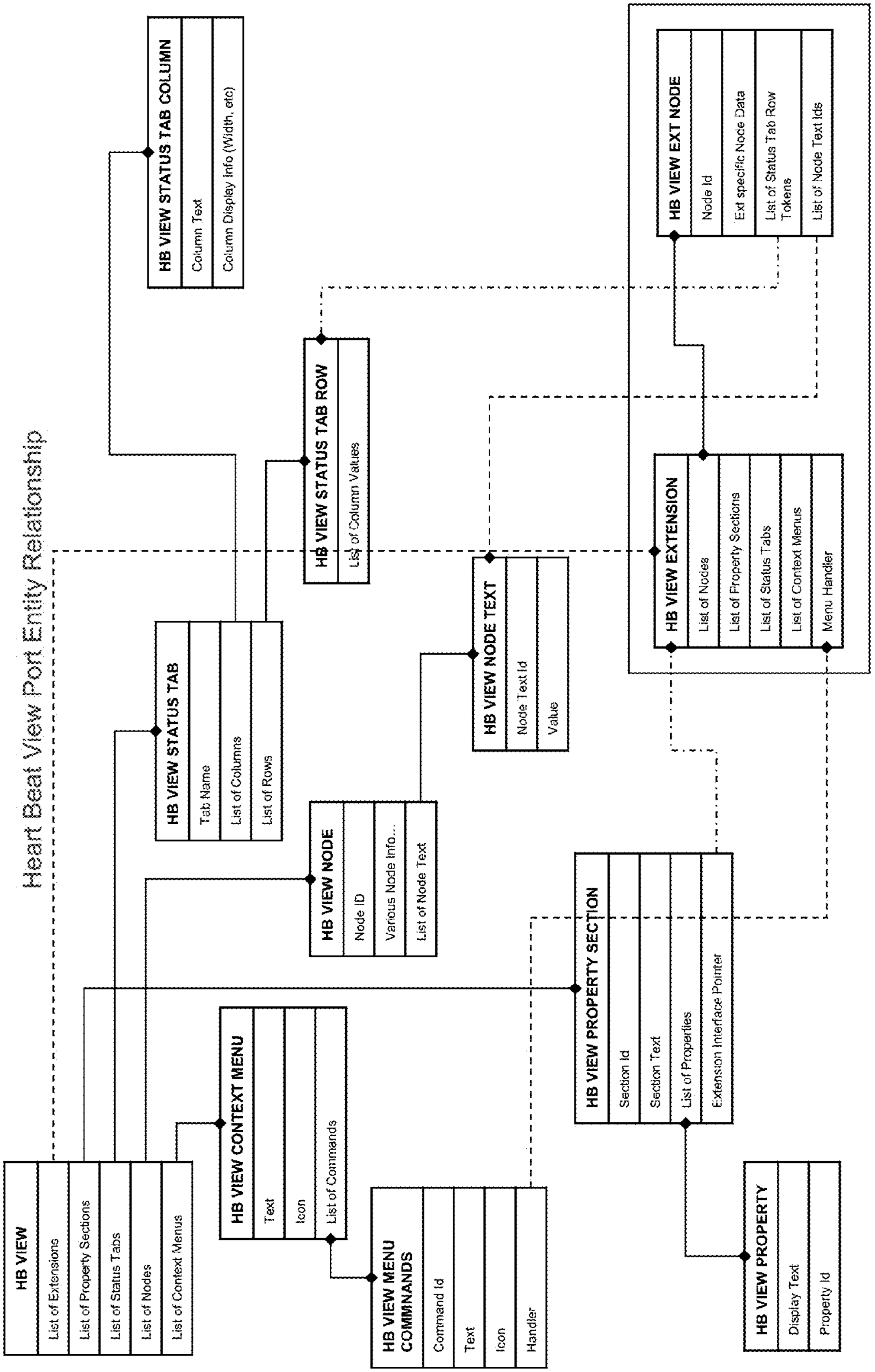

Heart Beat View Port Entity Relationship
HB VIEW
List of Extensions
List of Property Sections
List of Status Tabs
List of Nodes
List of Context Menus
HB VIEW CONTEXT MENU
Text
Icon
List of Commands
HB VIEW MENU COMMNANDS
Command Id
Text
Icon
Handler
HB VIEW NODE
Node ID
Various Node Info...
List of Node Text
HB VIEW STATUS TAB
Tab Name
List of Columns
List of Rows
HB VIEW STATUS TAB COLUMN
Column Text
Column Display Info (Width, etc)
HB VIEW STATUS TAB ROW
List of Column Values
HB VIEW NODE TEXT
Node Text Id
Value
HB VIEW PROPERTY SECTION
Section Id
Section Text
List of Properties
Extension Interface Pointer
HB VIEW PROPERTY
Display Text
Property Id
HB VIEW EXTENSION
List of Nodes
List of Property Sections
List of Status Tabs
List of Context Menus
Menu Handler
HB VIEW EXT NODE
Node Id
Ext specific Node Data
List of Status Tab Row Tokens
List of Node Text Ids

Figure 54

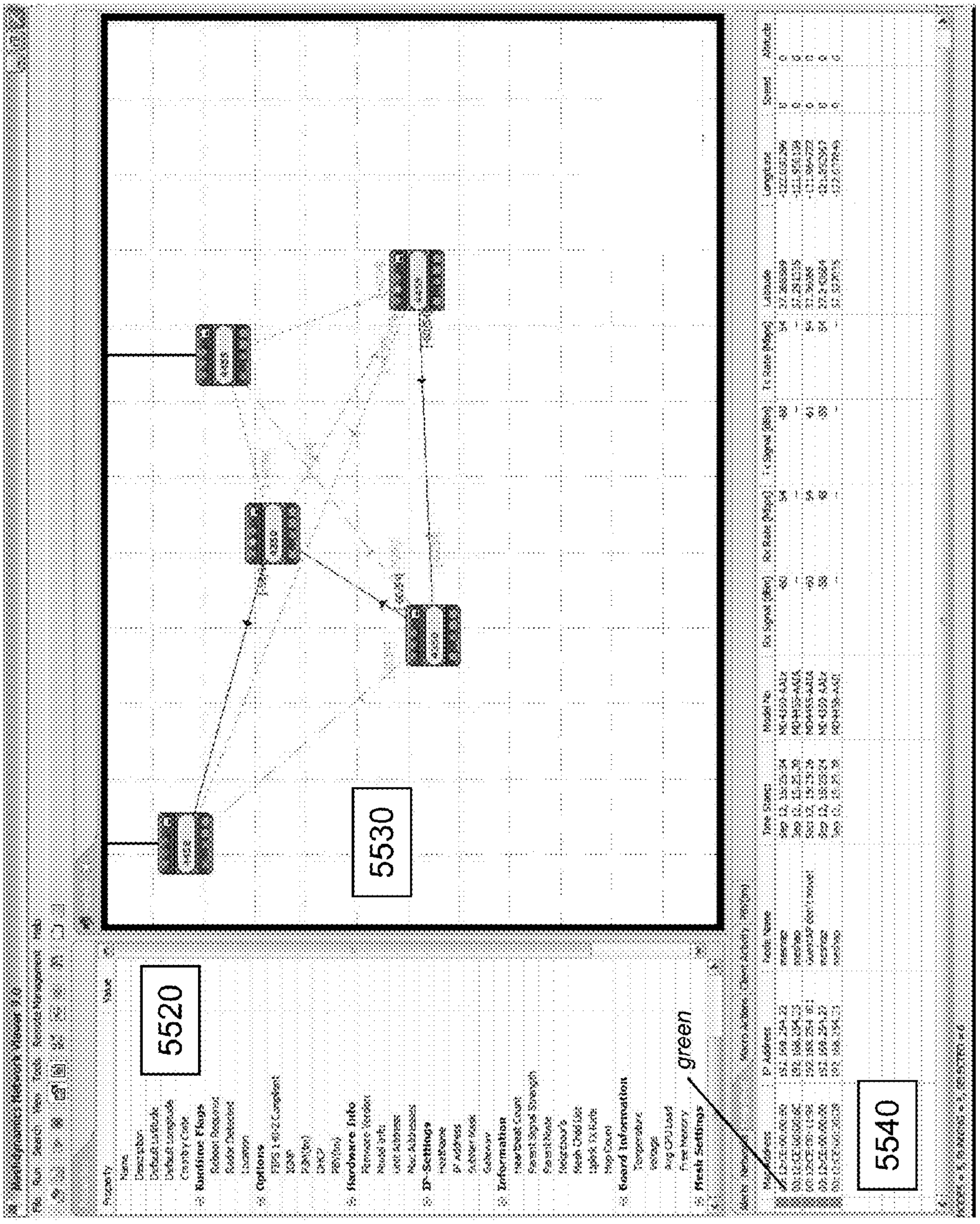

Property
Value
Name
Description
Default Latitude
Default Longitude
Country Code
Runtime Flags
Reboot Required
Radar Detected
Location
Options
FIPS 140-2 Compliant
DHCP
Hardware Info
Firmware Version
Model Info
Unit Address
Mac Addresses
IP-Settings
Hostname
IP Address
Subnet Mask
Gateway
Information
Heartbeat Count
Parent Signal Strength
Parent Node
Neighbours
Mesh Child List
Uplink TX Rate
Hop Count
Board Information
Temperature
Voltage
Avg CPU Load
Free Memory
Mesh Settings
5520
5530
5540
green

Figure 55

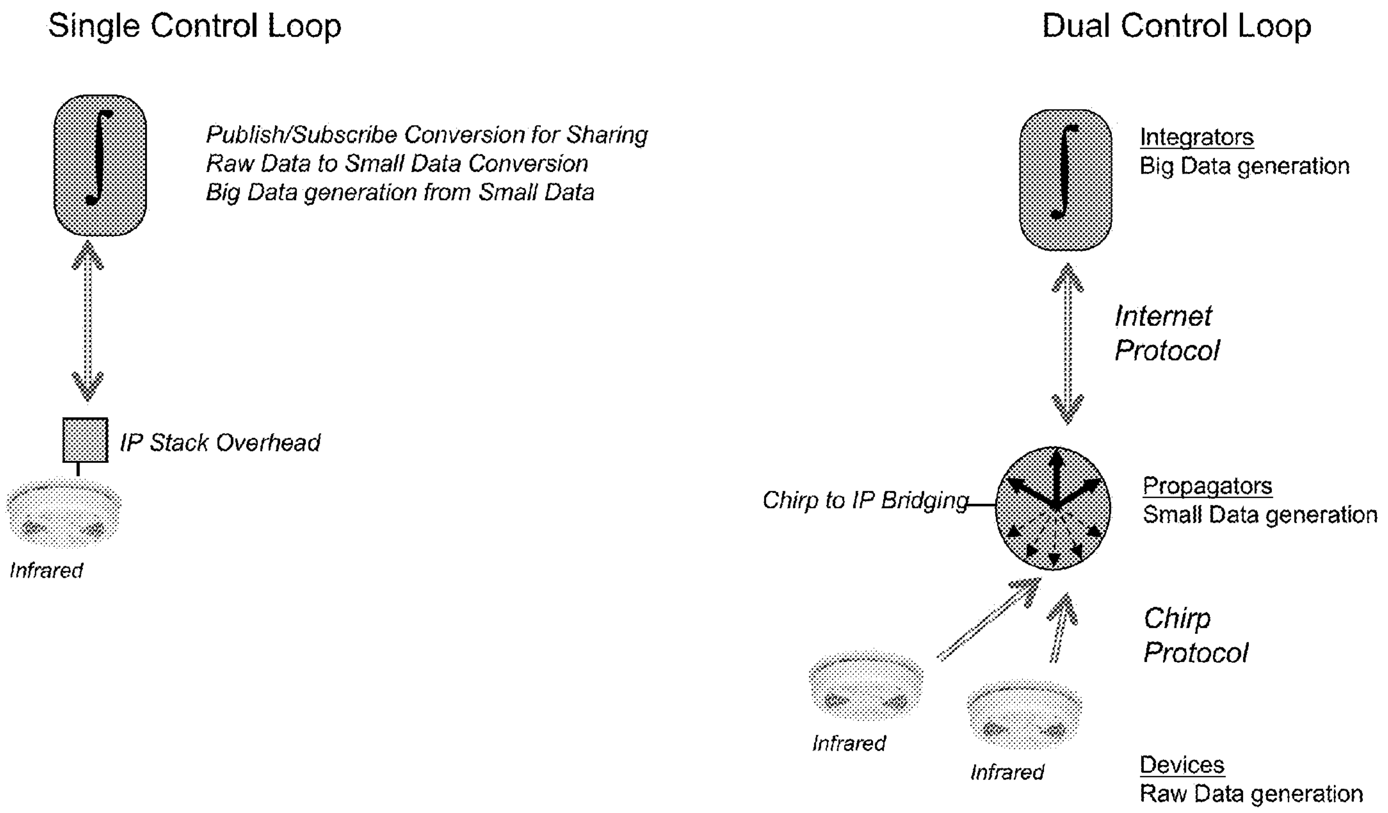

Single Control Loop
Publish/Subscribe Conversion for Sharing
Raw Data to Small Data Conversion
Big Data generation from Small Data
IP Stack Overhead
Infrared
Dual Control Loop
Integrators
Big Data generation
Internet
Protocol
Chirp to IP Bridging
Propagators
Small Data generation
Chirp
Protocol
Infrared
Infrared
Devices
Raw Data generation

Figure 56

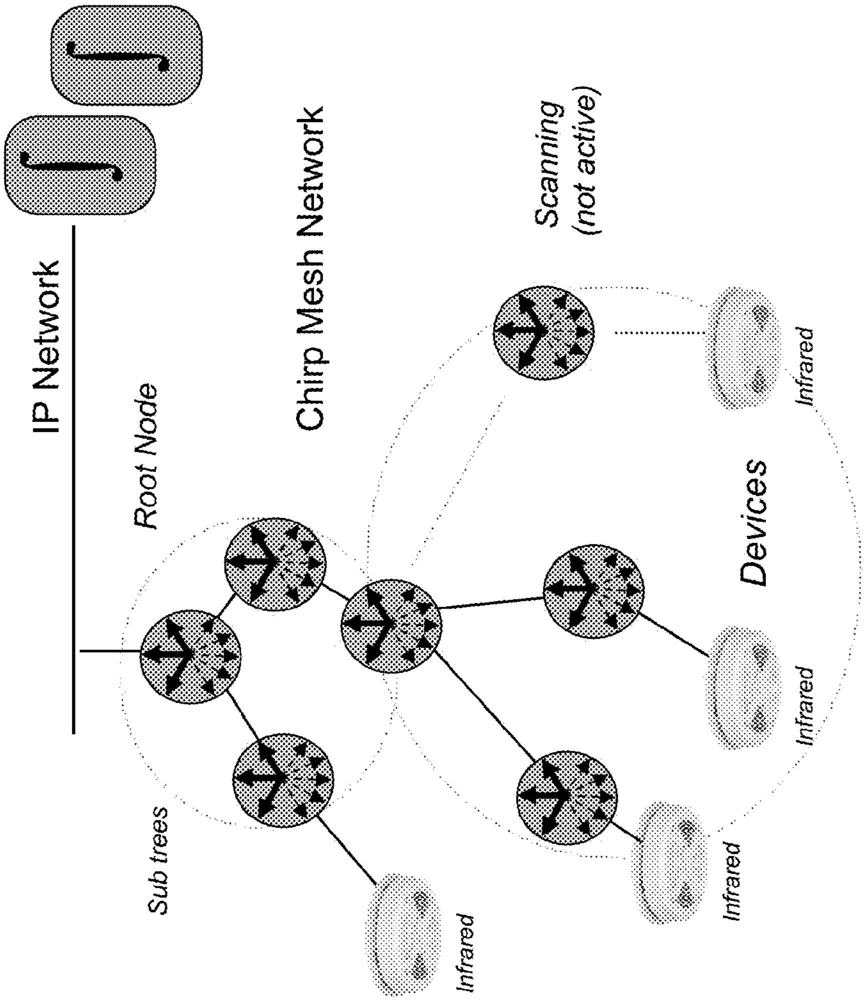

IP Network
Root Node
Chirp Mesh Network
Scanning
(not active)
Sub trees
Devices
Infrared
Infrared
Infrared
Infrared

Figure 57

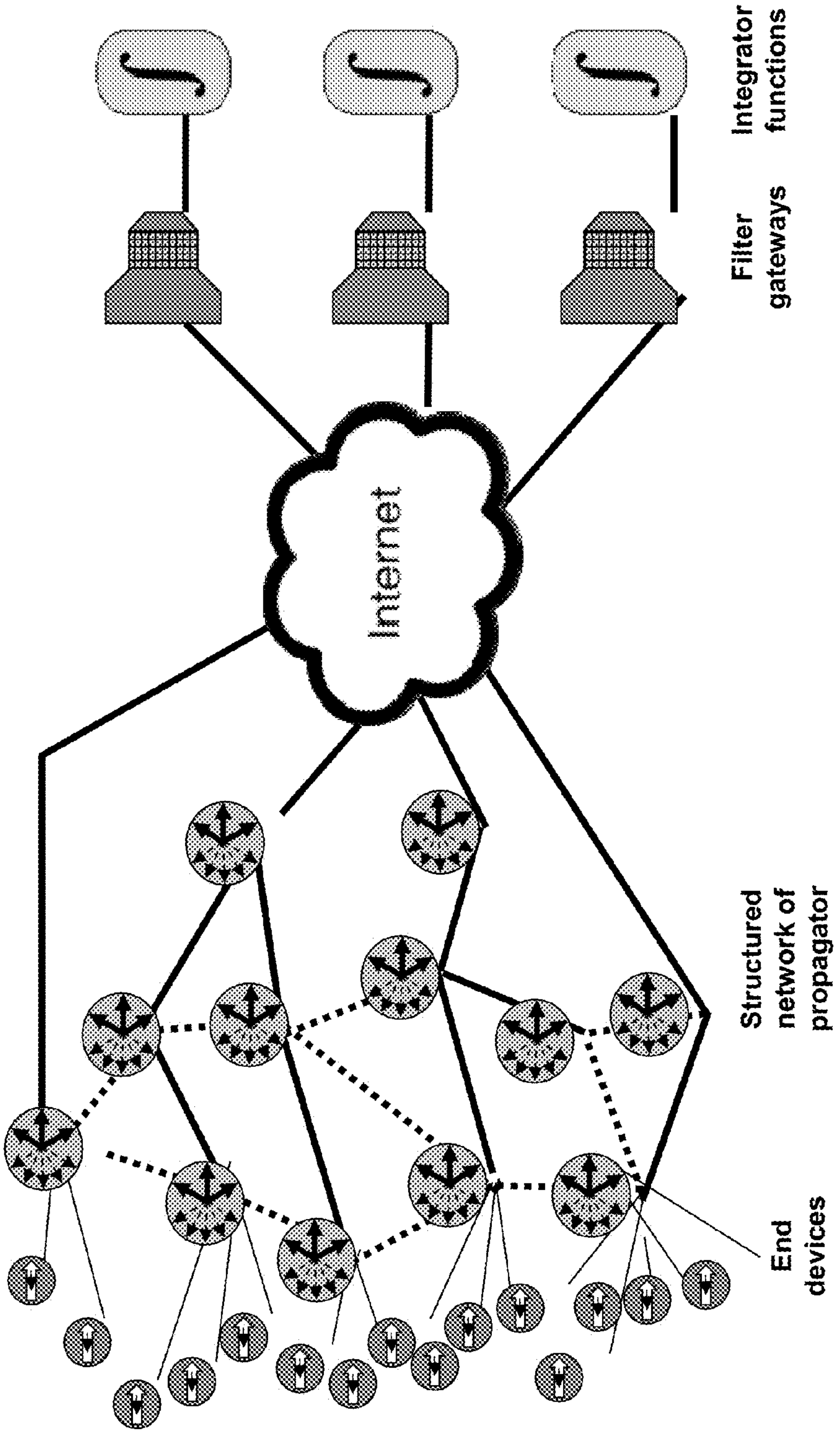

Integrator functions
Filter gateways
Internet
Structured network of propagator nodes
End devices

Figure 58

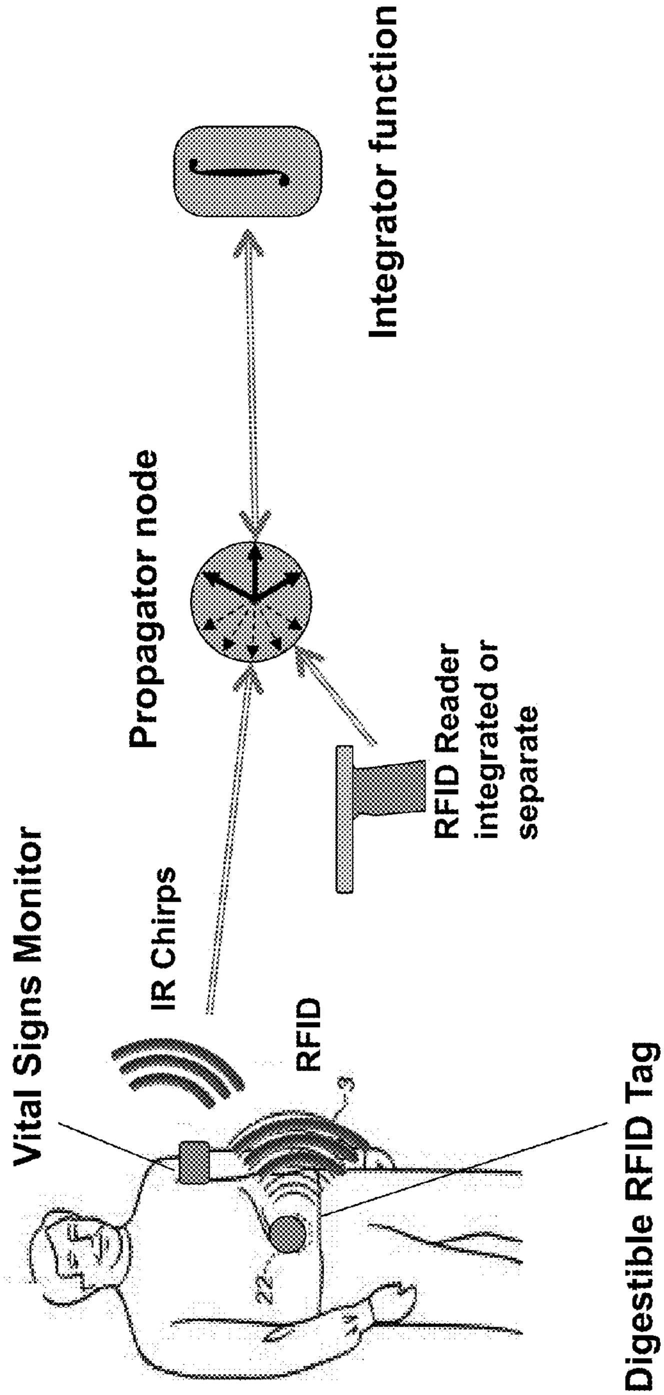
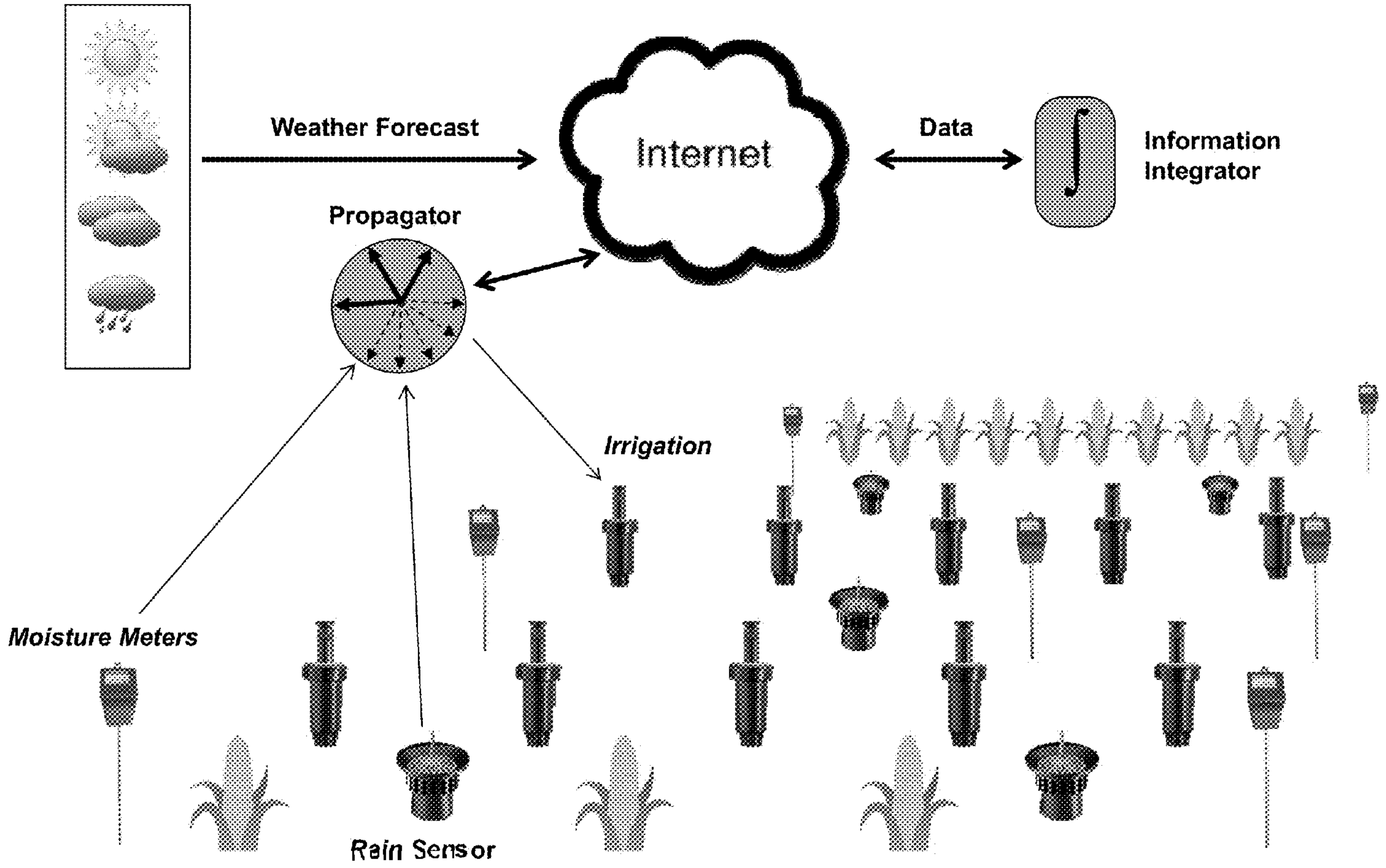
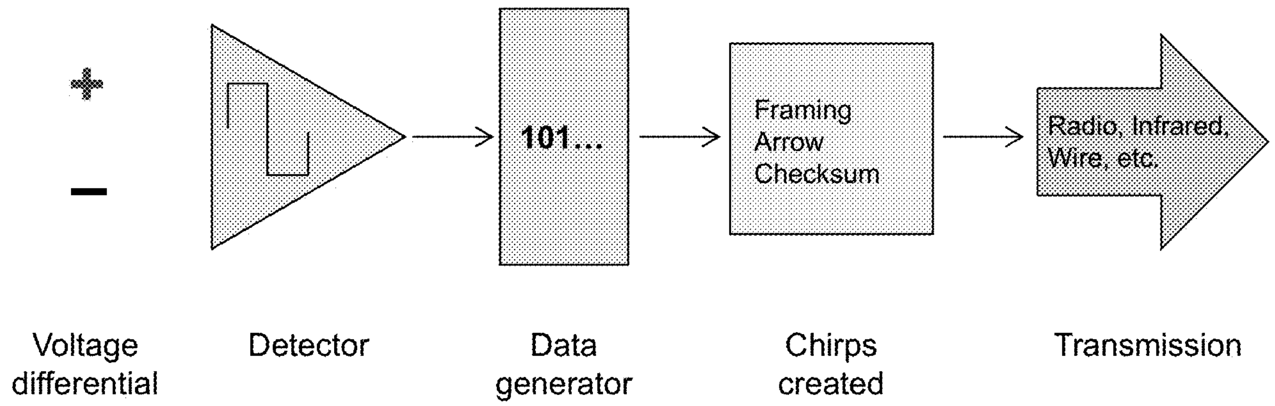

Vital Signs Monitor
IR Chirps
RFID
3
22
Digestible RFID Tag
Propagator node
RFID Reader integrated or separate
∫
Integrator function

Figure 59

Figure 60

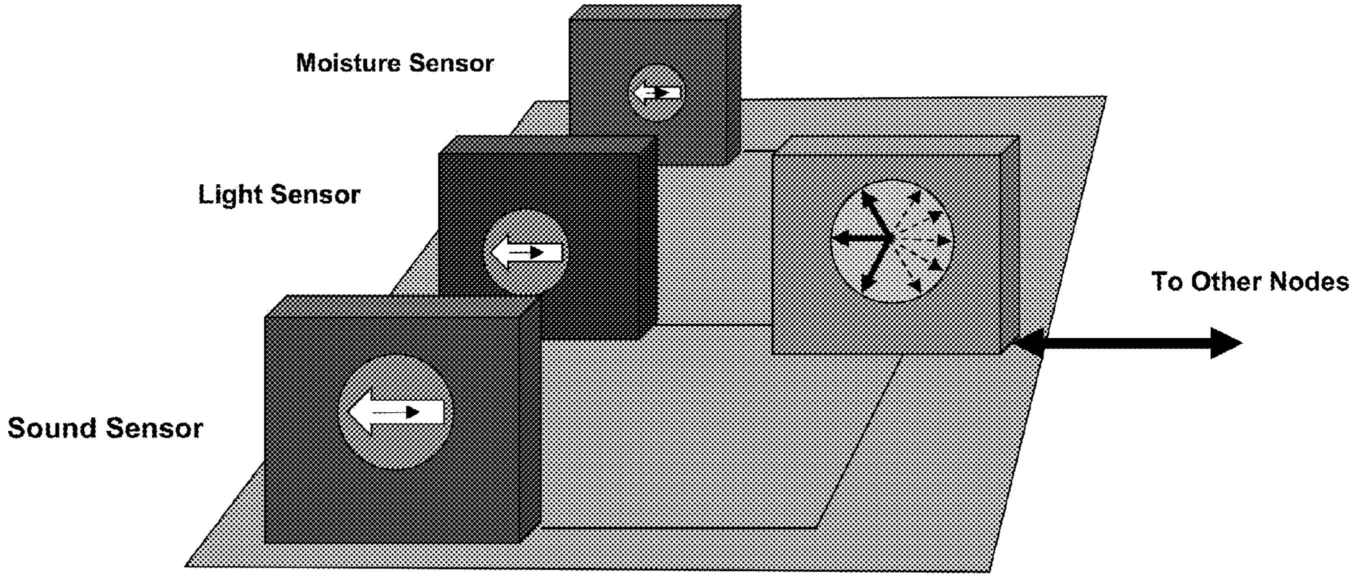
An Integrator function retrieves data from End Devices such as moisture sensors as well as external feeds and uses the information to control irrigation valves.





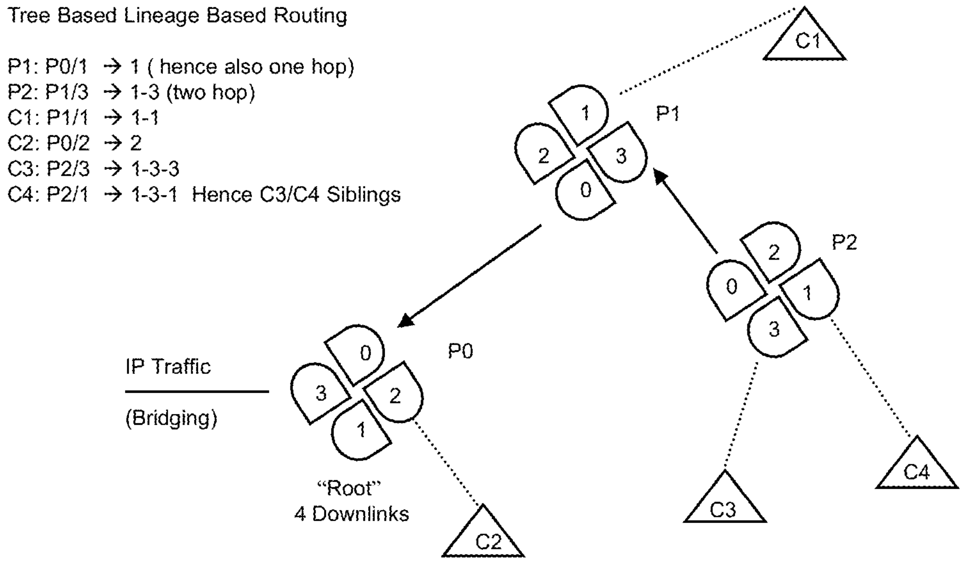
In millions of simple End Devices, basic physical states will be converted to Chirp payloads. An address, "arrow" of transmission, and checksum are added to this payload to form the complete Chirp packet.

Figure 61



Some devices will combine multiple IoT functions in a single package. Here an End Device includes a Propagator node that may provide networking services for additional nearby End Devices.

Figure 62



Tree Based Lineage Based Routing

P1: P0/1 → 1 (hence also one hop)
P2: P1/3 → 1-3 (two hop)
C1: P1/1 → 1-1
C2: P0/2 → 2
C3: P2/3 → 1-3-3
C4: P2/1 → 1-3-1 Hence C3/C4 Siblings

Figure 63

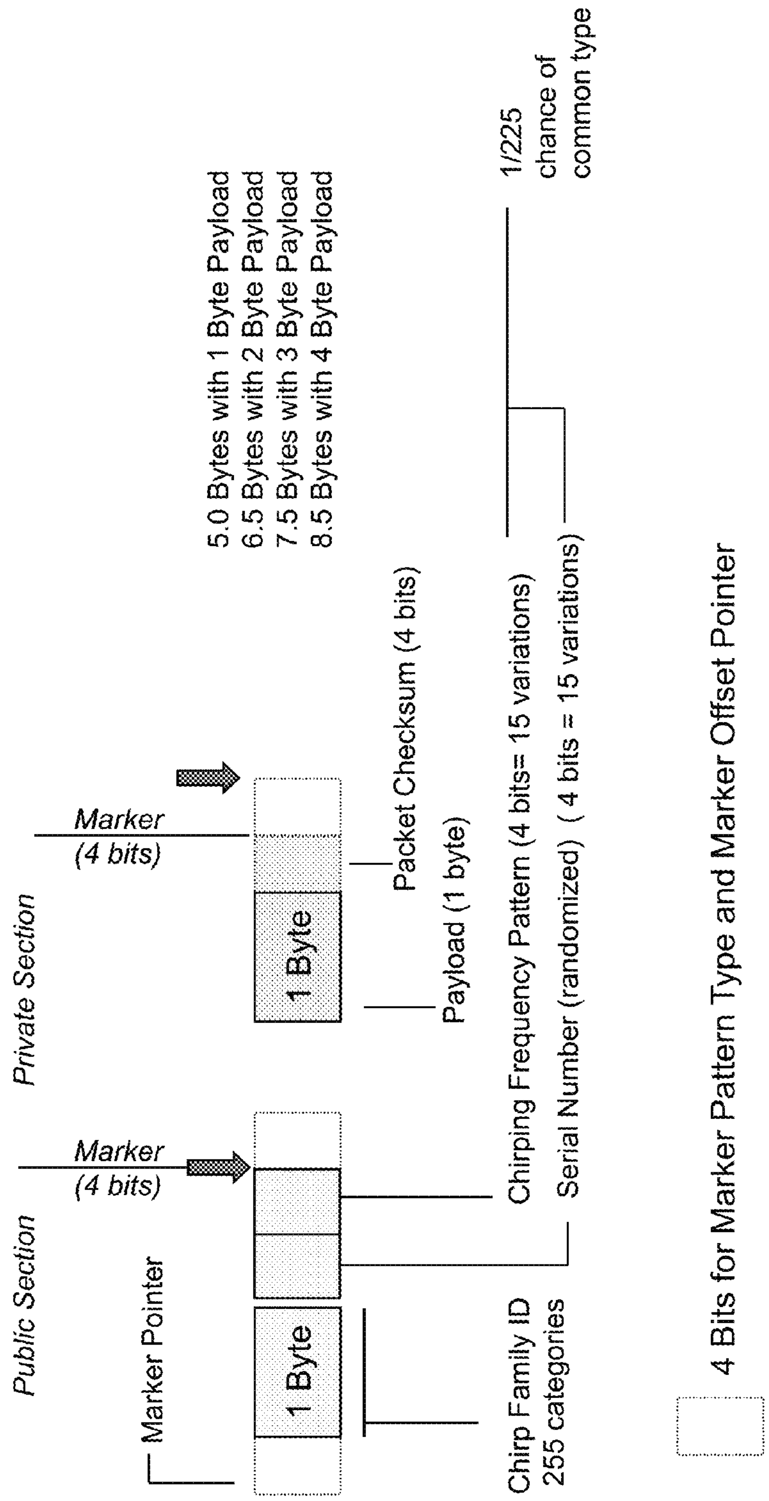

Public Section
Private Section
Marker Pointer
Marker
(4 bits)
Marker
(4 bits)
1 Byte
1 Byte
Chirp Family ID
255 categories
Chirping Frequency Pattern (4 bits= 15 variations)
Serial Number (randomized) (4 bits = 15 variations)
Payload (1 byte)
Packet Checksum (4 bits)
5.0 Bytes with 1 Byte Payload
6.5 Bytes with 2 Byte Payload
7.5 Bytes with 3 Byte Payload
8.5 Bytes with 4 Byte Payload
1/225
chance of
common type
4 Bits for Marker Pattern Type and Marker Offset Pointer

Figure 64

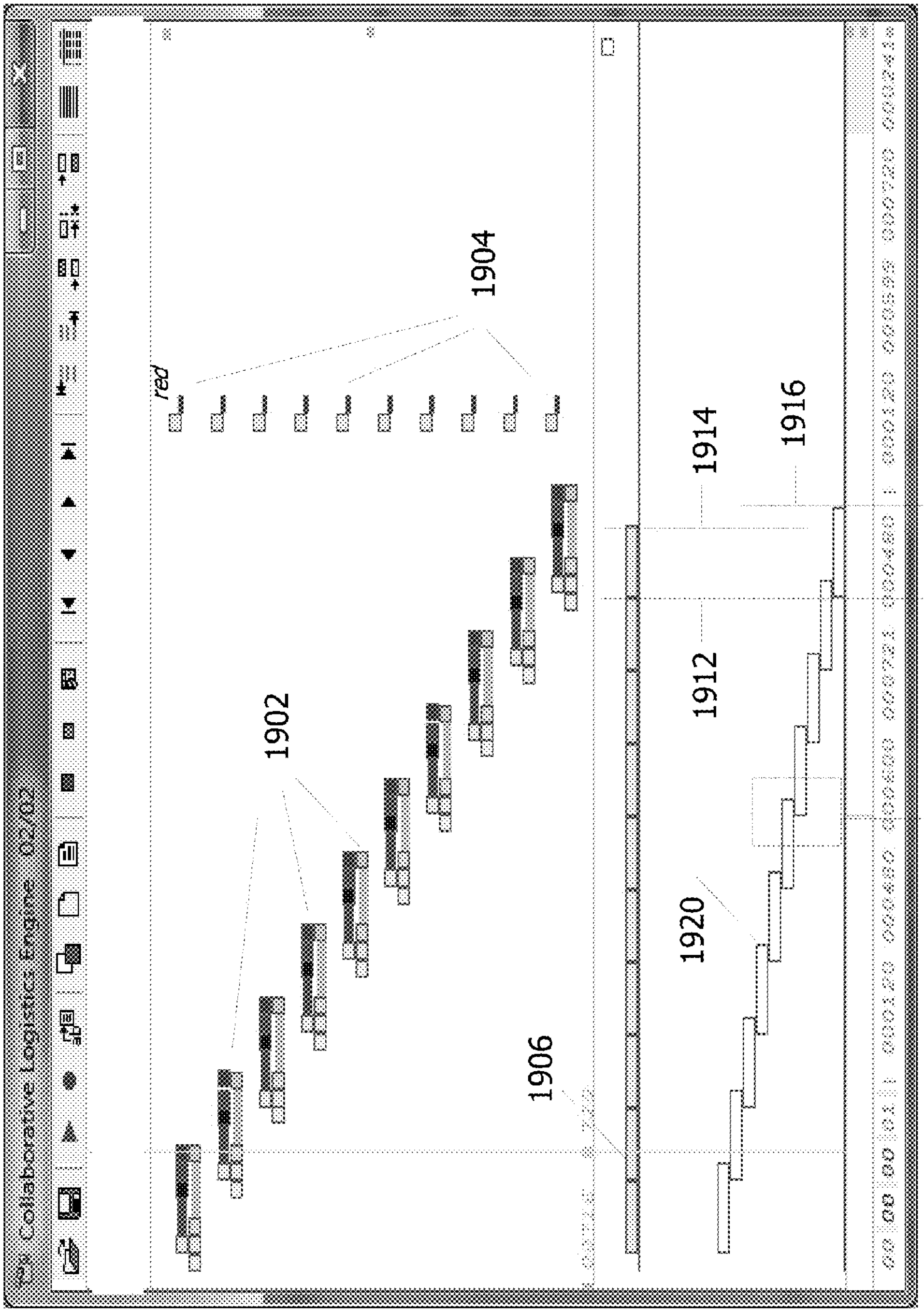

Collaborative Logistics Engine 02/02
red
1902
1904
1906
1912
1914
1916
1920
1922

Figure 65

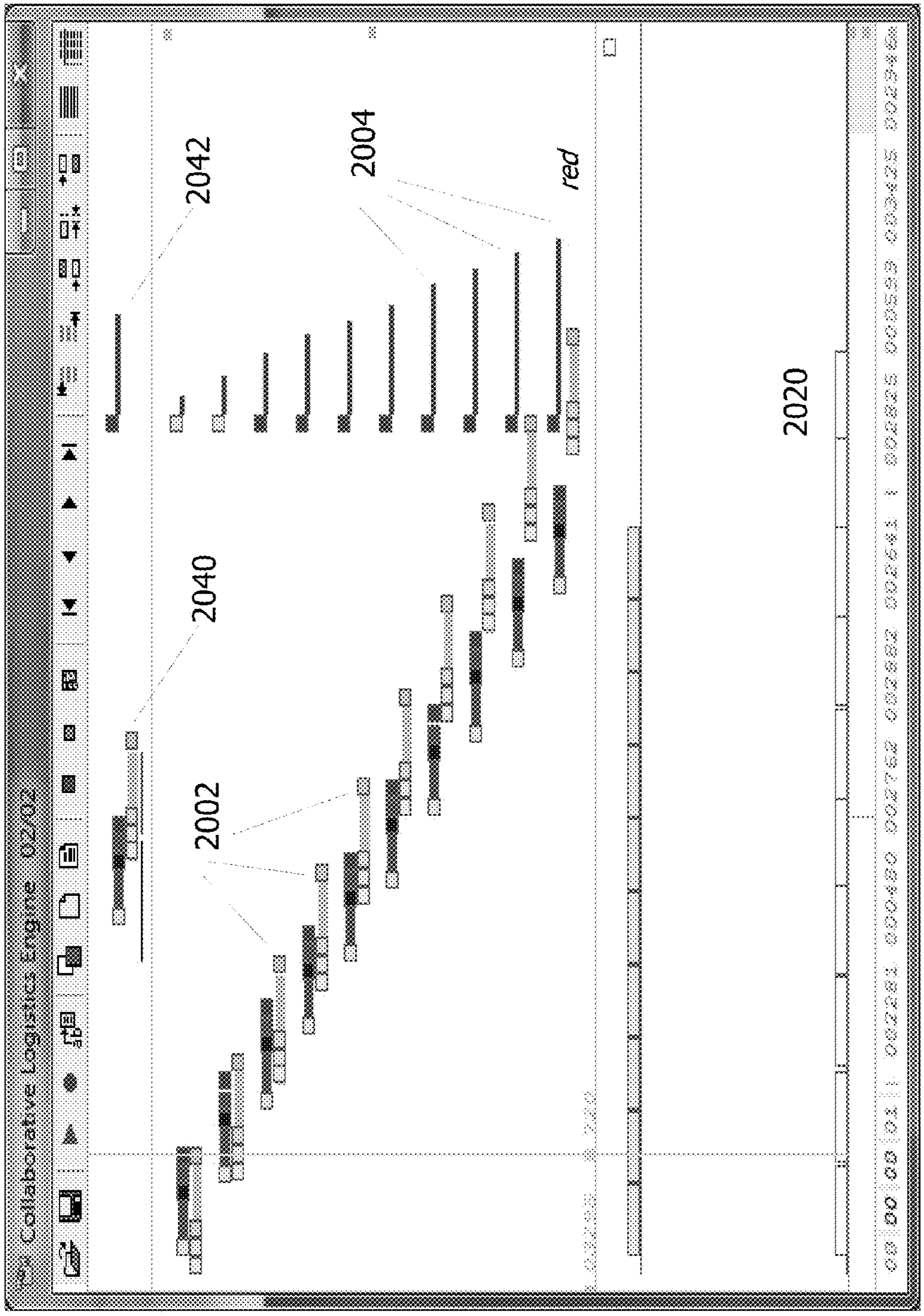
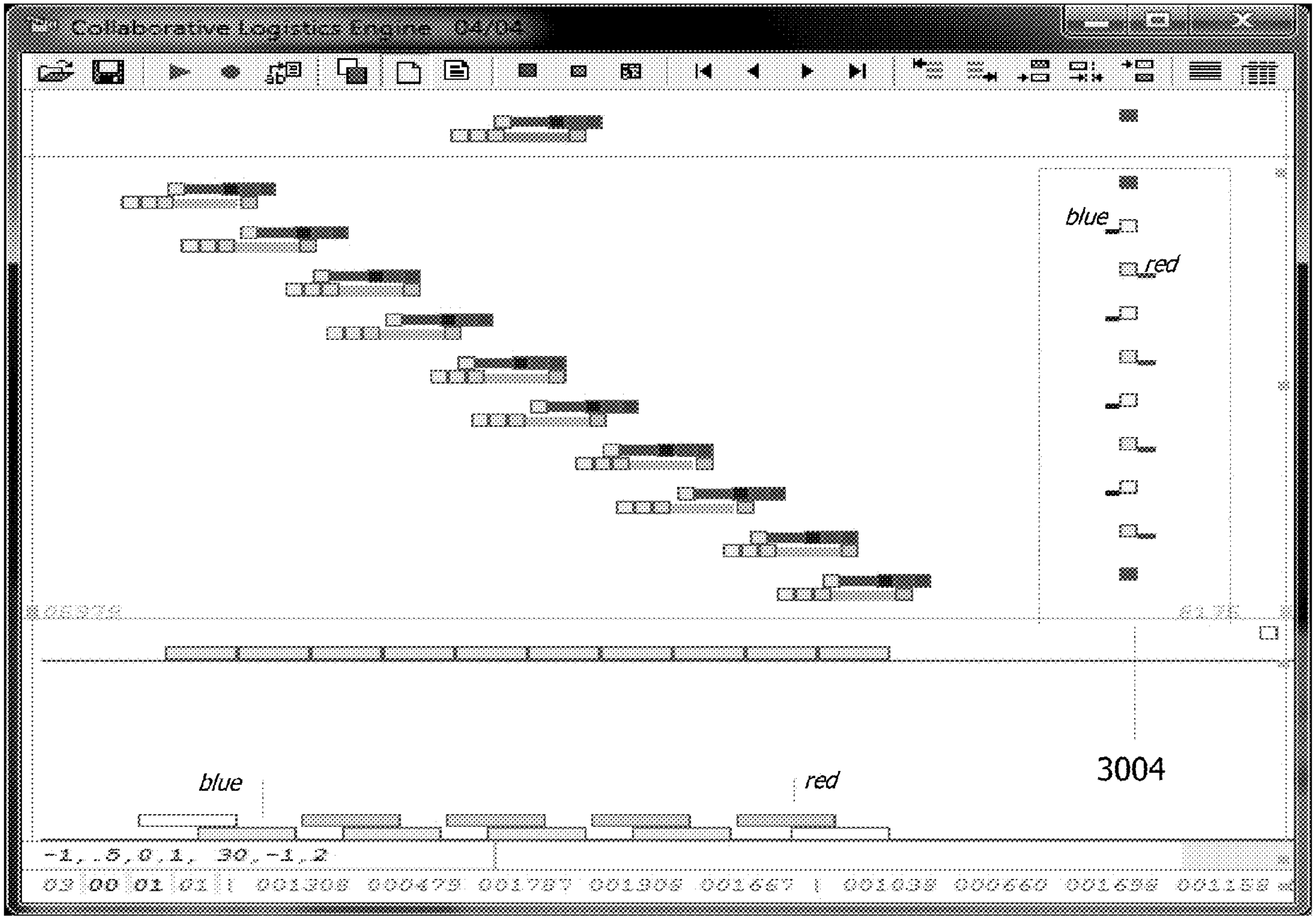

Collaborative Logistics Engine 02/02
2042
2004
2040
2002
red
2020

Figure 66



```
-1, .5,0,1, 30,-1,2
03 00 01 01 | 001308 000479 001787 001308 001667 | 001038 000660 001698 001158
```

Figure 67

CHIRP NETWORKS

CROSS REFERENCES

The instant application claims priority as a continuation in part of U.S. Utility application Ser. No. 13/764,008, filed on Feb. 11, 2013, which in turn claimed priority as a continuation in part of U.S. Utility application Ser. No. 13/627,883, filed on Sep. 26, 2012, patented as U.S. Pat. No. 8,923,186 on Dec. 30, 2014.

U.S. Utility application Ser. No. 13/764,008 also claimed priority as a Non-Provisional of Provisional U.S. Patent Application Ser. No. 61/615,802, filed on Mar. 26, 2012, and Provisional U.S. Patent Application Ser. No. 61/621,926, filed on Apr. 9, 2012, the contents of which are hereby incorporated by reference.

The instant application claims priority as a continuation in part of U.S. patent application Ser. No. 13/571,294, filed on Aug. 9, 2012, the contents of which are hereby incorporated by reference.

The instant application claims priority as a continuation in part of U.S. application Ser. No. 13/541,446 filed on Jul. 3, 2012, which in turn claimed priority as a non-provisional of U.S. Provisional Application No. 61/555,400 filed on Nov. 3, 2011, the contents of which are hereby incorporated by reference.

The instant application claims priority as a continuation in part of U.S. patent application Ser. No. 12/696,947, filed on Jan. 29, 2010, patented as U.S. Pat. No. 8,520,691 on Aug. 27, 2013, which in turn claimed priority as a non-provisional of U.S. application Ser. No. 61/148,803 filed on Jan. 30, 2009, and also as a continuation in part of the U.S. Utility application Ser. No. 11/084,330 filed Mar. 17, 2005, currently abandoned which in turn is a continuation-in-part of U.S. Utility application Ser. No. 10/434,948, filed on May 8, 2003, patented as U.S. Pat. No. 7,420,952 on Sep. 2, 2008, the contents of which are hereby incorporated by reference, including Appendix A.

The instant application further claims priority as a continuation in part of U.S. Utility application Ser. No. 12/352,457, filed on Jan. 12, 2009, patented as U.S. Pat. No. 8,477,762 on Jul. 2, 2013, which in turn claimed priority to U.S. application Ser. No. 11/266,884, filed on Nov. 4, 2005, and issued as U.S. Pat. No. 7,583,648 on Sep. 1, 2009, the contents of which are hereby incorporated by reference.

The instant application further claims priority as a continuation in part of U.S. Utility application Ser. No. 12/625,365, filed on Nov. 24, 2009, patented as U.S. Pat. No. 8,514,852 on Aug. 20, 2013, which in turn claimed priority to U.S. application 61/117,502, filed on Nov. 24, 2008, which are hereby incorporated by reference.

FIELD OF THE INVENTION

This invention relates to the field of computer networks and machine communications and, more particularly to a network system and method for facilitating large scale messaging emanating from the edge of the network e.g. sensors and actuators that interface with the dynamics of the physical world and each other, through terse messaging (chirps). Small" data publishers near the edge, condition the data based on subscriber interests

BACKGROUND OF THE INVENTION

Over the next decade, most devices connected to the Internet or other global network will not be used by people in the familiar way that personal computers, tablets and smart phones are. Billions of interconnected devices will be monitoring the environment, structures, transportation systems, factories, farms, forests, utilities, soil and weather conditions, oceans and resources. Many of these sensors and actuators will be networked into autonomous sets, with much of the information being exchanged machine-to-machine directly and without human involvement

Machine-to-machine communications are typically terse. Most sensors and actuators will report or act upon small pieces of information—"chirps." Burdening these devices with current network protocol stacks is inefficient, unnecessary and unduly increases their cost of ownership.

The architecture of the Internet of Things necessarily entails a widely distributed topology incorporating simpler chirp protocols towards at the edges of the network. Intermediate network elements perform information propagation, manage broadcasts, and provide protocol translation. Another class of devices house integrator functions providing higher-level analysis, for both near-edge analytics and broader-scope analysis. Small chirp data will feed "big data" systems.

The propagation of pollen and the interaction of social insects are relevant to the emerging architecture of the Internet of Things described in the instant application. Pollen is lightweight by nature, to improve its reach. It is inherently secure, only the receiver can decode its message. Nature's design is very different from today's traditional large packet and sender-oriented network traffic.

This application describes reasons why we must rethink current approaches to the Internet of Things. Appropriate architectures are described that will coexist with existing incumbent networking protocols. An architecture comprised of integrator functions, propagator nodes, and end devices, along with their interactions, is explored. Example applications are used to illustrate concepts and draw on lessons learned from Nature.

SUMMARY OF THE INVENTION

This invention addresses multiple embodiments of the logical 2 radio approach depicted in FIG. **4** provides solutions that address a variety of networking applications. These embodiments are shown in FIGS. **5**, **6**, **7**, **8**, **9**. Extensions to support voice and video in mission critical public safety networks are described in FIGS. **10**, **11**, **12**. This invention also addresses how combining the logical radio embodiments may be dynamically reconfigured—in the field and on the fly—to support both infrastructure style mesh networks and conventional ad hoc mesh networks. These Hybrid mesh networks for military and public safety applications are discussed and shown in FIGS. **13**, **14**, **15**. Implementation specifics and Architecture abstractions are addressed in FIGS. **16** through **21**. FIGS. **22** through **23** extend the logical 2 radio approach to include Chirp devices, bridging over to the IP side using propagators, Propogators are depicted in FIGS. **56** through **63**. Organically grown, (not central standardization committee based) chirp protocols is introduced, see FIG. **64**. Its relevance to localized zones at the network edge is explored. These edge chirp mesh network extensively use methods taught in referenced applications notably:

a) Aggregation, FIGS. **12** and **24** through **26**,

b) Distributed DHCP address generation, FIGS. **27** through **29**,

c) Generation of compiled rule based directives specifically for embedded systems, see FIGS. **30** through **31**

d) Timing the transmission times based on Collaborative Scheduling, see FIGS. **65** through **67**

e) Methods of coexisting within the same wireless medium as IP devices using RTS/CTS/NAV

BRIEF DESCRIPTION OF DRAWINGS

In order to more fully describe embodiments of the present invention, reference is made to the accompanying drawings. These drawings are not to be considered limitations in the scope of the invention, but are merely illustrative.

FIG. **1** illustrates how the network topology is changed by selecting a different backhaul in a two-radio system, with one link to the backhaul AP and the other link servicing the child AP. It depicts four network topologies. Each of the four network topologies provides a different set of performance in terms of latency and throughput. The mesh control software adjusts the latency and throughput parameters to meet voice/video or data performance requirements in terms of latency and bandwidth.

FIG. **2** contrasts the conventional "Dual Radio" mesh with the Logical Two-Radio Mesh. On the LHS of FIG. **2**, 2 radios labeled **010** and **020** provide client connectivity and ad hoc mesh backhaul functionality, respectively. All the mesh backhaul radios (**020**) are on the same channel depicted by the clouds of coverage of the same color (**030**). There are all part of the same sub-network. In contrast, on the RHS of FIG. **2**, the same radio (**025**) provides service to clients and also backhaul functionality but operates in different sub domains depicted by different color clouds of coverage (**035**). The LHS resembles a "hub", the RHS a "switch". Hubs are not scalable.

FIG. **3** compares the two step process of a single radio relay to a two-radio relay. On the left side, (**010**) a single radio relay is shown. Every packet received has to be re-transmitted on the same radio. Thus the bandwidth with every hop in a single radio mesh network is reduced by approximately 50%. After three hops, the Bandwidth would be ⅛ of what is available at the Ethernet backhaul. On the RHS (**020**) a two-radio backhaul is shown, where packets received on one radio are re-transmitted on another radio operating on a non-interfering channel. Now there is no bandwidth reduction with every hop and bandwidth is preserved with every hop. Two radio mesh backhauls are thus scalable while single radio backhauls are not.

FIG. **4** shows how the structure of two-radio multiple hop mesh network where each two radio unit services a Basic Service Set (BSS) by configuring one of the two radios to serve as an AP to its clients. Clients may include the second radio of another two radio system, with this radio configured to run in station mode, providing the backhaul path back to the Ethernet link. In the insert, the uplink radio (labeled **010**) connects to the parent mesh node while the downlink radio (labeled **020**) acts like an Access Point to client radios, including other mesh nodes that connect to it through their uplink radio. Note that all service radios (**020**) operate on different non-interfering channels, depicted by different color ovals.

FIG. **5** shows the similarities between FIG. **4** and a wired switch stack with the same chain of connectivity **040-050-060**. Both have the same tree-like structure and up link and down link connections. In both cases the units (**040,050,060**) operate on a distinct sub domain.

FIG. **6** illustrates one embodiment of the two logical-radio approach with three physical radios. Two radios constitute logically one radio of the two logical radio concept, while the third physical radio serves clients as the second radio of the two logical radio concept. By eliminating the sharing of physical radios for both backhaul and client services, the backhaul bandwidth has improved and also reduced the dependency to use the same type of radios for the backhaul and the client. In the insert, the uplink radio (labeled **010**) connects to the parent mesh node while the downlink radio (labeled **020**) connects to the uplink radio of child mesh nodes. The service radio (labeled **030**) act as Access Points to client radios shown as triangles. One such is labeled **040**. Note that all service radios (**030**) operate on different non-interfering channels, depicted by different color ovals.

FIG. **7** illustrates another embodiment of the two logical radio approach but with five physical radios. The uplink and downlink radios (shown as one radio FIG. **6**) are split into two radios, in this embodiment, with each responsible for one direction of traffic. Bandwidth is doubled and latency halved, since traffic in the opposing direction now has its own channel or logical "lane". Thus, the radio labeled **010** in FIG. **6** is now radios **012** and **010**. Similarly, the radio marked **020** in FIG. **6** is now split into radios labeled **022** and **020**. The radio pairs **012-010** and **022-020** provide the same functionality as the radios labeled **010** and **020** in FIG. **6** but with twice the bandwidth and approximately half the latency.

FIG. **8** is an extension of the five-radio embodiment shown in FIG. **7**. In FIG. **7**, there is one service radio to service both voice and data customers. However voice and data traffic has different performance requirements. By having different Access Point radios service the voice and data clients, the contention between voice and data packets attempting to gain access to the same medium is reduced. Also, with different radios servicing the data and voice clients, the voice and data packets can now be treated differently. The Access Point radios servicing the voice clients could therefore be operating in TDMA (time division multiple access) mode while the AP radio servicing the data clients operates in CSMA (Collision Sense Multiple Access) mode. The two radios (**032**) and (**034**) thus provide different functionality. VoIP devices such as phones connect to the former, data devices such as laptops to the latter.

FIG. **9** is a five-radio extension of the three-radio configuration shown in FIG. **6** but with more dedicated service radios operating on different frequencies for different types of client radios.

FIG. **10** shows the maximum VOIP bandwidth available per client, using 802.11 radios, as the number of clients increase. This is the size of the packet that each client can send every 20 ms. As the number of clients increase the size of the packet—and the associated voice quality—drops dramatically. In one embodiment, to achieve 64 Kbps voice quality, a 802.11b radio can support around 25 clients.

FIG. **11** shows the maximum VOIP bandwidth available per client, using 802.11a radios, as the number of clients increase. This is the size of the packet that each client can send every 20 ms. As the number of clients increase the size of the packet—and the associated voice quality—drops dramatically. In one embodiment, to maintain 64 Kbps voice quality, an 802.11a radio can support around 55 clients.

FIG. **12** shows extensions developed and implemented in the mesh network stack to provide an efficient backhaul for voice. The small voice packets are concatenated into larger packets and sent (as one packet) at regular intervals to the backhaul radios. At each mesh node voice packets intended for that destination are removed and the rest sent back (as one large packet).

Salient portions include the Packet classifier (labeled **010**) that recognizes voice packets based on size and regularity of transmissions, the VOIP concatenation engine (labeled **020**) that "containerizes" small voice packets into a larger "container" packet for more efficient transportation, Real time extensions (labeled **030**) to the Linux kernel enable the system to provide near real time performance regarding sending and receiving the latency sensitive VOIP container packets through the network—regardless of what the Operating System is doing at the time.

FIG. **13** shows the concept of a “Hybrid Mesh network” where 2 radio systems provide two types of service. In one case, they are part of an infrastructure mesh as shown by the 2-radio mesh node labeled **010**. In another embodiment, the same node may be dynamically reconfigured to support ad hoc peer-to-peer connectivity. The node labeled **020** (marked as E**8**) has two radios. One is intended for client radio connection to infrastructure mesh nodes—see the radio labeled **030** on the unit marked E**9**. The other provides a peer-to-peer mesh capability, as shown by radio labeled **040**. Depending on the needs of the network, the 2-radio units are dynamically re-configured to support either need, infrastructure mesh (**010**) or backhaul support to ad hoc mesh (**020**). Labels **050** and **060** designate connected and broken ad hoc mesh links.

FIG. **14** is an application of the Hybrid Mesh concept to a Public Safety embodiment. The node labeled **010** is a Stationary node on top of a light pole, in this embodiment. A mobile embodiment shown as labeled **020** is entering the building (arrow) such as when carried by firefighters. These mobile units are also called “breadcrumb” routers. The Mobile Mesh nodes provide connectivity to two-radio portable units worn by the firefighters in this picture. All firefighters are thus connected to themselves through a peer-to-peer mesh network shown as a thin line. They are also connected to the Infrastructure mesh backhaul through one or more connect points. This ensures redundancy in network connectivity. The broken link (labeled **060**, FIG. **13**) is avoided.

FIG. **15** is an application of the Hybrid Mesh concept related to a Battle Force Protection embodiment.

FIG. **16** depicts an embodiment using mesh nodes which feature four radio slots used in the modular mesh framework of FIG. **17**. There are two slots for radio cards on the front and back. Up to four radios **010** are thus supported on a single embedded systems board. The radio card antenna connections **030** are included for four radios. Two Ethernet ports **020** provide wired access to provision wired uplink and wired service access.

FIG. **17** indicates the modular mesh framework, whereby a four slot board, as shown in FIG. **16**, may be configured to provide different functionalities: Two radio Backhaul (BH) **010**; three radios BH+AP **020**; four Radio with BH AP and Scanner **030**; four radio with Full Duplex (FD) using a coupled two radio BH **040**. Further, since the modular mesh framework always forms a tree, these nodes are part of a switch hierarchy, as shown in FIG. **5**.

FIG. **18** depicts how the installation software is tagged to both the radios and board characteristics. It shows a serial line connected to load the boot loader program, after which the Ethernet port is used to complete the software installation and branding process. Compiling the install program on the board it is intended to run on performs this function, thus creating a unique software image.

FIG. **19** is a screen dump of the Flash Deployment software developed and implemented to ensure that software generated for the install of this board cannot be used by another mesh node. When the software installation process begins, the software is compiled on the board it is intended for and the compilation process is unique since it is based on a number of unique factors. The software is generated on the board that it is intended to run on—to ensure that the software image cannot be used to run on another board, thus preventing both software privacy and dissuading theft of the mesh nodes.

FIG. **20** shows that the Mesh Control Software sits above the Media Access Control (MAC) of the radio. As such it is radio and protocol agnostic, in one embodiment.

FIG. **21** shows how channel interference is dynamically managed in the logical two-radio system.

FIGS. **22** and **23** introduce an embodiment bridging across diverse wireless medium using the example of an N-Logical wireless medium bridge, referred to as the “nightlight” In one embodiment, the nightlight serves as both range extender and intermediary between device “chirps” and more conventional, IP based, communication devices and protocols.

FIG. **24** shows the synchronization of multiple voice devices accessing the same wireless medium with a focus on the time for bulk receipt of packets that are shared among the separate devices.

FIG. **25** shows a voice device talking to a dedicated voice radio and data devices taking to a data radio, with one phone **2501**, capable of taking to both **2502** and **2503** in one embodiment. The night light embodiment **2504** manages both voice and data transceivers, in the depicted embodiment.

FIG. **26**. depicts the dynamic collaboration tree for an exemplary supply chain application.

FIG. **27** shows an isolated mobile network cluster and communication within it using VOIP phones.

FIG. **28** through **29** describes an embodiment wherein isolated network clusters may converge with distributed DHCP services and inherent conflict resolution using randomized sub net address ranges.

FIG. **30** through **31** depict the process of generating an OS less image for secure small footprint devices and an exemplary graphical programming environment for simple sensory devices, in one embodiment.

FIGS. **32** and **33** depict device repeaters and range extenders. Conceptually these devices are similar to clients, such as the soldiers shown in FIG. **15**. They provide blind repeating and therefore range extension for remote devices. They also serve as a redundant path, similar to FIG. **13** but employing a single physical radio. Thus, in chirp language, they are birds that repeat and relay a bird song in string of pearl configurations.

FIG. **34** through **36** depict representative IP based “light” or low payload packets that may be used to transport chirp data over a IP based network. 802.11 packets are used as examples. Chirp data is encapsulated in such packets for onward transmissions, in unicast, multicast or broadcast modes, in search of flower/agents/tunes/subscribers interested in the chirp/pollen. In one embodiment, chirp devices use such innocuous frames to transport payload—only chirp aware routers know how to recognize them as chirp packets and process their (secure) routing to appropriate agents accordingly.

FIG. **37** through **38** map the equivalent slots/ports of wired and wireless switch equivalents as shown in FIGS. **4** and **5**.

FIG. **39** shows how logical radio modes, Uplink (U), Downlink (D), Scanner (S), Access Point for Service (A) map to physical transceivers in single radio and multi radio mesh node embodiments. The joining of tree branches **3950** to tree trunks **3960** is aided by common routers **3952**.

FIG. **40** is a simulation of a representative prior art mesh routing algorithm and its comparison to tree based routing of FIG. **4**. The thicker blue lines in FIG. **40**, **4040** denote the minimal spanning tree. Note the dashed lines have to be additionally recomputed for each node in prior art mesh routing. Performance deteriorates exponentially as O (n^2) where n is the number of mesh members.

FIGS. **41** and **42** depict a switch equivalent of logical radios operating in both wired and wireless mediums/channels, using Logical Radios Uplink (U), Downlink (D), Scanner (S) and O(n) routing. The logical radio switching module (insert) is introduced.

FIG. **43** is a schematic of how the abstraction layers for logical radios may themselves be combined to create more complex abstractions. **4320** refers to two abstractions AP (also in FIGS. **12**, **17**). The "bridge" is a combined logical radio abstraction, similar to the Uplink and Downlink (U+D) backhaul, FIG. **17**, but bridging over disparate frequencies and protocols.

FIG. **44** shows the bridging function (as described in FIG. **43**). Mobile node **4455** switches from "blue" 5.8 G backhaul to a "pink" 2.4 G backhaul. The sub tree beginning with mobile node **4457** is thus operating on a non-interfering channel/frequency/protocol. The static counterpart is **4460**.

FIG. **45** depicts a "string of pearls" configuration of static mesh nodes. A mobile mesh node, traveling at 60 mph makes temporary connections with each node in the string. Switching from node to another is seamless and unbroken, as noted by the video output below. Note that this is raw video and did not include the efficiency enhancements described in prior application Ser. No. 12/625,365. The process is repeated with single radios embodiments, using logical radios. Bandwidth degraded along the string of pearls, as expected, but video output was still jitter free and unbroken, due to proactive Scan Control, FIG. **12**, Logical Radio abstractions and the benefits of O(n), tree based routing.

FIG. **46** is reprinted from Ser. No. 11/434,948 (FIG. 10). It depicts the dependency of latency sensitive traffic to the network tree topology, specifically, the number of siblings in sub trees along the route to the destination node/parent/root.

FIG. **47** depicts the use of a reserved time slot for transmitting bulked, latency sensitive data, in accordance with the protocol explained in Ser. No. 11/266,884, whereby clients remain silent during transmission in this time slot. The time slot allocations may be fixed or variable. In one embodiment it is dynamically managed by Collaborative Scheduling, 61/555,400

FIG. **48** depicts broadcasts/streams restricted to a region. The region may be defined by geography, membership and mesh topology e.g. restrict the number of hops or sub trees. Further, the region may include directions: up/down or a set of turn by turn directions. An example of regional streams may include a section of the home, where only siblings of a sub tree need Note that backhaul bandwidth is not affected outside stipulated regions. Restricted broadcasting improves overall network health.

FIG. **49** is effectively the reverse of FIG. **48** and is global: e.g. not restricted. Tree based topologies favor global broadcasts. Streams from the root are always downwards. While streams from nodes may be either, they are typically upwards. The majority of devices populate the edge of the network and their pollen is typically upward bound, necessitating bulking, exception handling and deterministic time mail delivery along the route.

FIG. **50** depicts the Stream Reader, an agent authorized to peer into network router transmission and receiving packet queues, prior to their onward transmission through the network. Like Post office sorters, they identify and sort packets for scheduled deliveries, prune dead letters, duplicate messages etc. They also provide decoded outputs for Stream viewers, a custom GUI for the data. Stream readers may also forward output to other readers, mail boxes or messaging systems.

FIG. **51** depicts a circuit diagram of Stream Readers and their associated Stream Viewers, wired together to provide a capability, in this case "feeding" a section of the composite view ports **5190**. The composite view port is back drivable since its connection may be to real or historical data.

FIG. **52** depicts the adapters and API interface components that provide an extensible, open library of stream reader and viewers. This enables the rapid prototyping of custom circuits to provide specialized competencies. The view port additions enable human participation in managing the network health. This includes, through adaptor view ports, all assets of the network and their health.

FIGS. **53** and **54** depict the published interfaces for the Network Manager Streams API and the Heart Beat Entity relationships, respectively. Together, they enable speedy viewport development.

FIG. **55** depicts an embodiment of methods outlined in FIGS. **47** through **54**.

FIG. **56** compares contemporary thin client, single control loop architecture to a dual control loop, with a membrane separating the two control loops but, through the chirp to IP bridge/membrane, there is bidirectional, pruned and selective traffic, based on collaborative scheduling of bus schedules in both directions. Note that the two control loops are working on their own frequencies but neither is waiting on the other, see also FIG. **46**. This predicates the need for a "hub", e.g. Propagator. They also serve as bridges between the upper control loop, IP based and lower, tighter (low latency/isochronous) control loops preferably in more efficient chirp protocols. Note that the overhead of converting raw, machine specific raw data to a more palatable device abstracted format (e.g. small data) is performed within the cloud in the single control loop model in one case, and delegated to propagators in the other. Further, the lower control loop, between propagators and devices can be low latency/isochronous while the upper control loop can focus on more infrequent high level tasks: performance tracking, exception handling system updates (event based, low latency), routine hourly reports (periodic) etc.

FIGS. **59**, **60** depict a publish/subscribe exchange/market supporting multiple devices and integrators each operating in their own private communities/but also part of the same logical exchange

FIG. **57** through **58** depicts a burgeoning market place "exchange", where the confluence of multiple sources of terse but potentially rich content streams, often in organic protocols occurs at propagator trees. The root node serves as the Chirp to IP bridge/membrane. Small data is progressively refined and pruned, in proactive manner, as data moves upstream, like salmon upstream. Chirp packets no longer needed are discarded along the way, thus managing content relevance. Since chirps are category based, the protocol handlers (on both sides of the membrane) are logically part of the same publish subscribe bus system, with buses operating at different schedules. The distributed system manages the bus schedules to ensure control loops at all levels are operating without disruption. An Analogy would be nation wide bus services involving both greyhound and county run bus services collaborating on bus schedules to minimize overall delay, based on current traffic supply/demand.

FIG. **59** is a example of a small "exchange" of multiple data streams, operating on different, non interfering wire-less media. The "vital signs" Integrator is fed exception and periodic, non urgent pruned data from the propagator. The exchange between device and integrator is managed by having two segmented control loops, maintained by the propagator.

FIG. **60** is an example of a proactive control system, operating on the confluence of both local and external data publishers. The propagator, with appropriate transceivers, picks up multiple sensor streams, from a grid of diverse sensor types. Local Integrators/Agents residing in the propagator, can quick discern patterns and overall state of a large area—since small data is being generated and shared across a local mesh network, see FIGS. **57** through **58**. Local data streams are consolidated to provide a composite view of the region of interest. This feeds a second control systems where big data publishers provide a more global perspective. Thus, weather forecasts predicting rain, can cause the cloud server to direct the propagator network to direct which section of the corn field need additional local irrigation.

FIG. **61** depicts the simple circuitry needed to mass manufacture very low cost, low footprint, light "pollen" generators.

FIG. **62** depicts a first layer, rudimentary small data generator, where sensor fusion/consensus generates early warning signals with fewer false positives. The propagator and first layer integrator may be bundled as one device, servicing both single and multi-sensor subscribers in the local meshed network and beyond.

FIG. **63** depicts a four leaf clover like propagator, with 4 independent transceivers, 90 degrees rotated from each other. The 4 transceivers may be logically assigned uplink, downlink and scanner functionality, based on where the clients and relay node parents are located and the current network routing priorities. The transceivers may be dynamically and logically reassigned as the network topology changes or as clients migrate into and out of the network. Note that, as other 4 port devices depicted in this application, there is only one uplink per the 4 leaf clover design—it is scalable O(n) tree architecture. One layer is shown—overlays of such devices cover different wireless transmission media (e.g. Infrared LEDS for Chirp and Bluetooth for IP networking) A stack of such four leaf transceivers than thus service both Chirp and IP clients, including other propagators. Since routing is lineage based, siblings are easily recognized. Ability to see siblings provides fail over redundancy with minimal change to routing tables. This engenders stable, healthy networks.

FIG. **64** depicts a category classification based approach to publishing M2M data. The Marker Pointer tells the router/propagator/agent where to look for the type of Pattern being used. Thus Marker patterns located at Byte 6, would be part of the 6.X . . . family. A 4 bit Marker pattern value of 15 would imply that data being published is part of the 6.4.15 family. This coarse granularity may be sufficient to route the published data in the general direction of interested subscribers. Further levels of finer granularity are available to agents aware of what Marker Pattern **15** signifies e.g. how the category data in the 6 bytes of (open but cryptic) finer classification data are expected to be read.

FIGS. **65** through **67** relate to the collaborative scheduling engine described in previous applications. FIG. **65** is the starting condition where each task is taking just a little longer than expected. The usual first in first out queue, see FIG. **66**, results on excessive delays further out for entities at the tail of the queue. FIG. **67** employs collaborative scheduling to balance out and maintain overall "customer satisfaction". In the example shown, two tasks are "stacked", because they are contending for the same resource a portion of the time. Stacking implies more resources are needed. When the resources are limited must be shared, this is not an option. As an example, consider the case of chirp devices and IP devices sharing the same wireless channel. Collaborative scheduling shifts the two tasks in opposite directions in time, to remove the stacking. The shifts are based on shifting those task start/end times within "Customer Satisfaction" constraints defined by the blue/red ranges for the task. An application to timing transmissions between chirp devices and their routers/bridges is discussed.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

The description above and below and the drawings of the present document focus on one or more currently embodiments of the present invention and also describe some exemplary optional features and/or alternative embodiments. The description and drawings are for the purpose of illustration and not limitation. Those of ordinary skill in the art would recognize variations, modifications, and alternatives. Such variations, modifications, and alternatives are also within the scope of the present invention. Section titles are terse and are for convenience only.

Chirp Device Extensions

Raison D'être for Chirp/Pollen Networks

This section is an elaboration of methods taught in earlier applications, including applications discussing the two-logical radio concept, VoIP data containerization, and software management. Specifically, the following concepts are applied in the embodiments discussed herein:

1. N Logical, Physical device abstracted wireless transceivers, FIGS. **7-9**, **13-17**, **22**
2. The wire-less Radio and Protocol Agnostic Mesh Control Layer, FIG. **20**.
3. Bridging across Wireless communications media, FIGS. **9**, **22**.
4. The scalability of tree based logical network and mesh architectures, FIGS. **4-8**, **24-26**.
5. Efficient use of wireless bandwidth by sending bulk or containerized packets, FIGS. **10-12**. **24-25**
6. Community mail boxes for intermittent connectivity (e.g. Emergency Response, Ser. No. 11/084,330)
7. Dynamic Mesh Topologies based on Aggregate Latency/Throughput tradeoffs (FIG. **1**)

This section applies those methods to transport "chirp" broadcasts akin to VoIP packets from "chirp" device networks to co-existent and incumbent IP based network devices and protocols.

Why Chirp Protocols are Needed.

Traditional networking protocols/techniques, specifically IP-based protocols, originated from sender-oriented communications. They emulated legacy frameworks of human-human networks to support human-to-human correspondence. Thus the methods of routing postal mail resulted in email. Emails are sender oriented. Sender oriented communications are intended to be read by stated intended recipients only. It is not lightweight: it contains destination addresses and encryption so they are not tampered by hackers. They were historically point to point communications.

With all this overhead in place, on every packet basis, IP based Routing protocols favor economies of scale in moving large packets, preferably with best effort only. When QoS is required (latency/jitter/assured delivery) IP protocols extract a premium. Small packets are also charged a premium because the "standard" minimum packet size was not designed with them in mind. Traveling inside the minimum size, they have to pay the same price in transmission time and bandwidth requirements as the "standard" minimum.

This adversely affects the delivery schedule: of lightweight packets (e.g. VOIP) that now travel no faster than large packets, through no fault of theirs, other being smaller than the standard packet size. Hence container based transport for VOIP, FIGS. **10-12** are discussed in referenced applications.

Filling the packet container reduces the "price" per VOIP packet and also ensures reliable (assured latency/jitter) delivery. Without containerization, client capacity rapidly deteriorates, FIGS. **10-11**. The IP highway is not suited for short chirp like devices, whether they are VOIP packets or other types of small-payload transmissions.

In addition to favoring larger packet sizes, retries (as in birds chirping repetitively) are frowned upon. Resending large packets increases traffic and TCP/IP overhead costs are based on a low number of retries. Further, retries are discouraged because they may flood the network e.g. broadcast storms. Retries are the exception, not the de facto modus operandi. Hence various forms of "virus" checks on email, file transports etc are used. Since small packets are treated as one large packet, any device that begins to "chirp" like birds who chirp blindly and repetitively are costly especially when these chirp-like packets are travelling solo, FIGS. **10-11** and hence the value of concatenation and scheduled transmissions of larger packets see **020** in FIG. **12**.

In contrast with legacy networks, communications that uses retries and over provisioning/broadcast storms are common in Nature. For instance, pollen distribution by plants is not sender oriented. Instead, as many messages are sent as possible, in all directions. Note that these storms (e.g. pollen, monsoons) are seasonal, their time to live functions are encoded in their design. They do not have to be explicitly stated in each packet header. Pollen has its time-to-live function genetically encoded in it. Beyond that time it is ineffective. No network flooding can occur, despite over provisioning. Broadcasts are managed at a distributed systems level through the mesh control layer, FIG. **20**, not on a per packet basis. Now, flooding can be contained, as in nature.

Further, Nature's "packets" are receiver oriented. Pollen is promiscuously propagated, witness the temporary broadcast storms called allergy season. It is the receiver that has the (genetically encoded) secret handshake to unlock the pollen packet data. Sender data need not be encrypted.

In Nature, receiver oriented security enables pollen to be lightweight (terse) and carried far by even small low power winds. Further, pollen is reasonably patient or latency indifferent. As long as winds appear within pollen season, spring will occur. Light weight chirps/pollen are thus secure and lower Total Cost of Ownership (TCO) based on their lightness and "patience."

In the world of Internet of Things (IoT), there is a need to be able to support lightweight chirp-like data without unduly taxing either the incumbent IP based networking protocols or the "chirp" device. This network protocol would support economical and effective transport of small, terse, repetitive "chirps". In many cases the chirps are latency indifferent. Further, they would allow varieties of subscriber driven (receiver oriented) temporary broadcast storms without adverse effects,

Collaborative Coexistence is key since some chirp packets have subscribers reachable only through the incumbent IP network, e.g. VOIP chirp-like packets for an overseas subscriber. These networks and protocols must also conform to the existing frameworks and mediation layers e.g. FIGS. **12**, **20**. IChirp/pollen devices are simple—they do not know where the flower is. Hence IP destination addresses cannot be provided. Chirp aware routing supports these new receiver oriented packets.

Further, if chirp devices intend to operate in the same frequency spectrum as IP devices, then both dance partners need to share the dance floor without constantly stubbing each others toes (e.g. collision→causing interference→some causing retries→possibly network flooding). Coexistence requires chirp devices not habitually create collisions or accidents on the IP based highways, like bad drivers. Methods taught with legacy chirp like devices (VOIP) Ser. Nos. 11/266,884 and 11/084,330, see FIGS. **10-12**, **24-25** ensure both types of transmitters dance well together in the same RF space.

In general, IP based routing services require customers to choose their "value" between extremes:

1. Guarantee of Delivery vs. Best effort (lower cost)
2. Maximum acceptable Latency (pay more for low latency)
3. Maximum acceptable Jitter
4. Cost tradeoffs between Throughput and Latency (see Fig and Appendix A)
5. Level of usage: unlimited versus bandwidth throttled (adversely affects latency)

On IP networks, the lowest cost is bulk mailing, latency indifferent traffic. Since pollen is reasonably latency indifferent, it makes sense to use scheduled bulk transport. FIGS. **10-12** teach methods related to which packets need to leave on which bus and how often those buses need to visit the bus station (polling frequency). FIG. **24-25** focus on what would be the best times for pollen to "arrive" at the bus station or other "wind" transport for onward routing.

Collaborative Coexistence with IP Devices

As the name suggests, "Chirps" are short duration and terse commands/status messages, primarily for machine to machine ("M2M") communications in home, factory or outdoor enterprise networks. M2M communications are purpose built and terse. Chirps from a TV remote to a TV, for example, has a limited but adequate word vocabulary to be able to control the complicated communications sources used by the television to change channels.

Using diverse wireless media (e.g. sonar, audio, Wi-Fi) reduces "channel" interference between concurrently active multiple radios, FIGS. **6-9**, **13-15**, **22-23**. However core issues of dynamic channel interference remain: there are still limits to the number of available "channels" despite extending the usable channel list. The challenges addressed by previous work, still remain real and relevant.

There is a need to devise means for coexistent, preferably non-interfering, independent and alternate communication devices/protocols that operate in the same frequency spectrum but with minimal interference with existing, incumbent and/or "standard" protocols.

IP based devices operate on CSMA/CA protocols. Using random back off, radios "sense" collisions and, in decentralized manner, avoid collisions. As an analogy, there can still be two or more people talking at the same time but it is minimized because most people are being polite and waiting their turn. In the event a packet did not go through, it is retried but this is the exception not the rule. The protocol is robust, scalable and ubiquitous for IP based wired and wire-less traffic. Thus an "agile" and "polite" competent networking protocol already exists.

Listening and avoiding interference in proactive manner is also referred to as "agile" and/or "polite" systems. Agile is analogous to one dance partner compensating for the (random) clumsiness of the other. Polite refers to listening and then avoiding interrupting others (collisions) in conversations.

Chirp devices and protocols leverage this politeness/agility of IP stack based devices to coexist. If Chirp durations are very small, many that occur during IP protocol enforced silent periods have no adverse effect. Further, chirps transmitted at the same time as active radios, may not significantly affect IP based traffic if they are so short that radios can adaptively manage the chirps as noise.

Low cost chirp devices can be very simple, in one embodiment. Some simple chirp device, like birds, chirp "blindly," with no consideration or verification of whether the chirps are heard or not. Multiple such blindly chirping devices can be chirping at the same time, resulting in retries and interference. These (repetitive) deadly embraces are avoided by randomly scheduling "blind" chirps by the device transceivers. This is a simple fix, used in one embodiment.

In general, the combination of short duration and random transmission may suffice in this un-orchestrated and decentralized framework, as in a park, with multiple birds, all chirping blindly.

Note this is "blind" randomness—there is no sensing component in the device. Hence it is not entirely fool proof. But, neither is CSMA/CA—e.g. "hidden node" problems. Further, not all chirp devices need be blind. Some capable of "listening" may adaptively time their broadcast per methods taught in application Ser. No. 11/266,884 and FIGS. 24-25.

In one embodiment, it is thus possible for IP protocol based wireless transceivers operate in the same frequency domains as chirp devices. For Wi-Fi radios, for example, these chirps are treated as random and transient noise. It is adaptively filtered out using Automatic Gain Control, Error Correction, Noise cancellation, and other methods, while the Wi-Fi radios are sending messages other than chirp communications.

As more chirp devices join the network, their random chirps are like white noise for the IP based wire-less or wired transceivers. These adaptive IP transceivers adjust. It is their modus operandi. At the worst case, one or two IP based packets may be retried, but IP throughput is not appreciably affected. It is certainly no less affected than having a nearby Access Point (AP) producing channel interference. This is because chirp transmissions are short, by design. Thus, short random chirps, are inherently capable of coexisting with legacy incumbent wire-less (or wired) transmission protocols.

In another embodiment, Chirp packets contain information of their intended transmission pattern. The Access point, can thus, anticipating a chirp, preempt contention by sending out a CTS "incoming" notification to its IP client stations, thus clearing the air waves. Further, sophisticated chirp devices, with listen/see capability can be directed by chirp router embodiments to modify transmission times and channels, see embodiments FIGS. **22-23**. Advanced chirp-aware routers, with chirp equivalent of scanning radios in one of the slots, see **030**, FIG. **17** may also scan the RF environment. The AP can then direct some chirp devices to schedule their chirps to "avoid" or "cluster" with other concurrent transmitters. Clustering changes the transmission times of listen capable chirp devices to engender sequential chirps—and the AP can schedule a bulk concatenated CTS "incoming" warning to its IP clients.

Further, Ser. No. 11/266,884 teaches methods so custom VOIP phones and advanced chirp inside (chirp aware) device equivalents, which schedule their chirps to avoid collisions with each other and downstream bulk broadcasts from the AP, as shown in the embodiment of FIG. **24**. Chirp aware devices are capable, therefore, of being agile and polite within the network environment. Instead of random blind chirps, these proactive devices are listening and timing transmissions to be collision free with bulk transmissions back from the bridge router, FIG. **23**, or other collaborating chirp inside devices (e.g. TV, camera etc), FIG. **22**. Collaborative ant like agents schedule small chirps to avoid each other or to cluster in dynamic alignment with resource stacks taught in Ser. No. 13/571,294.

Thus, Chirp protocols require no modifications to existing de facto standards wire-less (IP based) devices. Such agile and polite IP based wireless radio transceivers are commodity items, supported by Apple, Atheros, Broadcom, Cisco, GE, Google, Intel, Motorola, Samsung, Sony, Qualcomm, etc.

Concatenation: Chirps are small and repetitive, analogous to small VOIP packets. IP does not favor small repetitive packets, see FIG. **10-12**. Like VOIP packets, chirp packets will be repackaged for bulk transmission when bridged from Chirp networks to IP networks, FIG. **12**, **22-23**. Small Chirp packets, like small VOIP packets, will be routed/distributed efficiently using pruned broadcasting methods as described in Ser. Nos. 12/352,457 and 11/266,884. In one embodiment, Chirps transmitted to IP based networks are containerized and under router management: they cannot individually create broadcast storms.

FIGS. **14-17** depict embodiments using interference agile, scalable wireless mesh networks using Wi-Fi transceivers ("radios") and IP based protocol. FIG. **9** depicts bridging across multiple wireless networks operating on non-interfering frequency bands and/or protocols (e.g. Wi-Fi, Wimax, VOIP). FIG. **22** indicates bridging to other wireless media on different types of frequency bands (e.g. infrared, Wi-Fi). Other wireless media include any form of electromagnetic communication, such as sonar, audio, light flashes. The Mesh Control layer, shown in FIG. **20**, is common to all.

Note that the mesh control layer provides "radio" and "protocol" agnostic mediation. These words were not restricted to either Wi-Fi radios or IP protocols, as indicated above.

In one embodiment, the system provides a common mediation layer for all disparate forms of mesh networks that are bridged together to form one logical meshed and scalable network topology, FIG. **22**. Only then can a logical mesh network exist, across multiple frequency and protocol domains. The wire-less radio and protocol agnostic mesh control layer (FIG. **20**) includes mesh networks different from IP based protocols as shown in FIG. **9**. A new protocol is "chirp" based. It is specifically designed to minimize interference with existent and incumbent network protocols. The mediation layer supports it, in one embodiment.

The mediation layer and the logical radio concept are interrelated. In FIG. **7-8**, two radios are shown, for backhaul, which together form one logical radio. That logical radio connects to another logical radio, also consisting of corresponding physical radios, see FIGS. **7-8**. Note the chirp-like devices (such as VOIP) are not sharing their physical radio with other devices. Thus, in the logical radio concept, both chirp-like devices and others are supported, in radio and protocol agnostic fashion. Thus the embodiment shown in FIG. **22** corresponds to FIG. **9**, and the disclosure of application Ser. No. 11/084,330.

With the mediation layer in place, chirps may be efficiently and reliably transported up/down the (bridged) logical network tree, comprising of both IP and chirp networks, see FIG. **22**.

Scheduled Bus Service for Chirps

Ser. Nos. 12/352,457 and 11/266,884 teaches methods for VOIP, to enable chirp like broadcasts from chirp like device networks to be efficiently repackaged to travel on existent and incumbent prior art IP based Network devices and protocols.

Further efficiencies are possible by engineering the broadcast "beam" direction and spread angle.

Nature uses undirected beams—as in no preferred direction. That covers a wide region. Less pollen directed to the intended recipients would represent a more directed and smaller "market" focus.

Two embodiments featuring different options exist. On the one extreme, scatter shot (undirected) seasonal broadcast storms are prevalent in Nature and tolerated because they are temporary (seasonal), they are necessary (for the flow of nature), and they are robust. (because they are over provisioned).

A temporary broadcast storm, despite the unwanted effects of allergy season, is tolerated. It is the only time-proven method of ensuring sufficient pollen is "heard" in Nature's chaos based ecosystem. The storm is intentional and over provisioned but effective. It is undirected, so it covers all bases.

Pollen is lightweight. That works in its favor. Small gusts of wind appearing randomly move it along. It may need multiple gusts of wind, like multiple relay hops in a mesh network. Nature does not know where 'subscribers" are. Scatter shot approaches cover large unknown subscriber regions.

When you do know your subscriber base, directed winds of chirps/pollen is more efficient. But chirps are light weight and receiver oriented. They do not have destination addresses. Like school children boarding a school bus, they must be directed to get on the "yellow" bus/wind. In one embodiment, at each bus stop along the way, they need to be told which next bus to board. The edge router FIG. **12**, **23**, manages bus arrival and departure times and also getting the chirp/pollen bulk containerized and ready for each bus trip. Where containers go is driven by the "clients": it is directing pollen to interested parties only. This is a more directed beam vs. scatter shot. The yellow bus routes mark pockets of subscriber interests, available for trend analysis, if desired.

Using the combination of VOIP like containerization and collaborative scheduling, pollen/chirps are delivered to the subscribers interested in them/waiting for them, in a timely manner. The chirp routers manage pollen/chirps getting on the correct "yellow" buses and bus departure schedules etc.

Buses leave at regular intervals to help schedule when a container of chirps arrives and leaves at each bus station. They are a part of the mesh network support infrastructure for chirp travel. Their frequency of arrival is driven by the polling frequency (e.g. 20 ms for VOIP phones). It will vary depending on the chirp device nature and urgency per methods taught previously, FIGS. 12, 24-25. Scanning and polling frequency is adaptive, per methods taught for mobile nodes, Ser. No. 11/818,899.

Additionally, subscribers, in one embodiment, awaiting specific pollen delivery, may also request sooner or later delivery, thereby changing QoS dynamically. Pollen bus depots (the routers) schedule bus size and frequency of departure (polling), as well as time spent waiting for data to collect (minimum queue), and other variables in certain embodiments. Collaborative scheduling agents, in some embodiments, ensure global supply chain alignment of supply/demand using simple concepts like "avoid" and "cluster" to ensure appropriate use of bandwidth resources and prevent "stacking". Ser. No. 13/571,294 et al. teaches scheduling collaborative agents in control systems.

Thus, QoS equivalent services are dynamically managed in a receiver oriented network. This is a departure from Prior Art, when sender oriented IP packets must declare their QoS requirements a priori, and QoS requirements are blindly enforced along the network path at significant added cost.

Subscribers are stationary and mobile. Hence routing is a moving target Mobile intermittent connectivity (not "always-on") is supported, FIGS. **27-28**. In some cases, like mobile phones, the client is currently unavailable and messages are delivered to a trusted community mailbox. In other cases, the subscriber may request temporary mail holds or re-directs, akin to when people relocate or move residence. Chirp routers are made aware of updated subscriber locations, by secure authenticated means, in one embodiment. They maintain a community mail box and support addresses for otherwise anonymous but authenticated subscribers. Thus chirp routers serve as post offices, part of a meshed postal service network infrastructure, with coexisting IP and Chirp publishers and subscribers, see FIG. **22**.

In addition to routing, providing mail boxes, holding and forwarding mail, chirp routers may also be requested to manage a family of devices. For example, in one embodiment, all kitchen appliances broadcast chirps towards the kitchen router, which is a night light in one embodiment, FIG. **23**. The kitchen router is authenticated to communicate with a smart phone using secure IP based networks in one example. When new devices are added to the kitchen community, the smart phone is informed by the router/nightlight which processes chirps incoming from the new devices. Thus, if needed, there is a human in the loop authorizing device "pairing" with the router. Further, if settings on the device need changing, then it is only night lights and their fellow router nodes of the mesh network that can provide the best non interfering channels to use, since only they know which channels they are using, to avoid conflicts with other mesh nodes operating on different channels but sharing the frequency bands. Further, devices that are roaming within the chirp or IP local network are automatically pre-authenticated. Methods taught in Ser. No. 12/352,457, for VOIP self-forming networks are used in one embodiment of this invention.

Chirp routers perform these routing based services based on chirp signatures, described herein.

These routers recognize chirp "colors"/signatures. Also, in one embodiment, they know where (authenticated) subscriber destinations are. Hence, only they have the means to connect the two pieces of data and to dynamically direct chirp/pollen containers to subscribers/agents/mail boxes. Even if they do not know where the agents are, managed multicast and broadcast techniques will be used to find the "flowers", described presently.

Subscriber Based Winds (of Change)

Multiple options exist between the two extremes styles of broadcast e.g. scattered vs. directed. From the perspective of device transceiver firmware, the product and its target market drives device vocabulary and device transceiver competence.

Dynamic Blobs of Interest/Trends.

In a receiver oriented world, pollen/chirp "publishers" are directed to "subscribers" with potential broad interests in multiple types of pollen/chirps. In the trivial case these interests maps to multiple individual active subscribers IP mail box addresses. Efficient, pruned broadcasting taught in Ser. Nos. 11/266,884 and 11/084,330, addresses those specific needs. In the less trivial case, the "destination" address is a movable target, both literally and figuratively.

In the literal sense, mobile mesh networks are needed to support mobile clients, see FIGS. **13-16**, where the soldiers and firefighters now carry chirp devices. Chirps are consumed within the Chirp network and "outside" by bridging, as per the embodiment in FIG. **22**. Publishers and subscribers may both be mobile.

In the figurative sense, communications are a movable target because of dynamic consumer needs. Demographics of the communication network, comprising of both human and machines, is in flux at all times, as shown in the embodiments of FIG. **13-16** and FIG. **26**. Thus the "winds" and yellow buses being engineered by edge routers embodiments FIG.

23, to move pollen/chirps efficiently, are also in flux. A distributed control system monitors the environment, in one embodiment.

Dynamic Load Balancing taught in Ser. No. 10/434,948 included switching data paths to less congested parent nodes. This in effect also changed mesh topology. This was needed because latency and throughput tradeoffs were being made. In the event that the traffic is (reasonably) latency indifferent, store and forward community mail boxes provide some relief to the deluge bound to a congested node in transient congestion situations. Thus, through the heart beat (Appendix A), node congestion levels may be communicated to adjacent "buddy" nodes, who then "hold the mail" until toll costs are lower. This exemplifies proactive re-scheduling, driven by dynamics of toll/hop costs, Appendix A.

Subscribers may also request redirection. For example, chirps from all Sony TVs in San Francisco, received between 8 am and 9 am today have been containerized. They are scheduled to be sent to a customer service center in Japan. The largest subscriber "blob" is in Japan. However, network connectivity is interrupted due to a natural disaster which has hit Japan. Those containers now need to be routed to the Customer service center in India.

One may argue that this is already handled—the Japanese Service center simply has to redirect all traffic to its "buddy" counterparts in India (as part of fail-over "buddy" system policies). Buddy systems may work for policemen, but not global supply chains. A simple redirect is not always possible. Enterprise global supply chains have multiple layers to their collaboration, as discussed in FIGS. 26, and Ser. No. 13/571,294. There are many "redirects" that need to be plugged in but they are dynamic. Supply chain topologies are increasingly more dynamic. There are no fixed corresponding failover sites for every element within the supply chain. There are also many level/tiers of suppliers and consumers. Likewise, devices themselves are more complex. A machine may chirp multiple messages, involving diverse service requests. In one embodiment, the present system determines who should get these requests, and does so on the basis of several variables—it is situation dependent. There are associated costs with different options. Finally, the device or machine has an owner with his preferences and overrides. One-to-one correspondence systems are too simplistic.

Rather than build complex buddy/failover systems, the receiver oriented approach presents a simple solution—chirps or pollen is simply redirected at each bus station/router along this route. The chirp routers and control system thus stays aligned with shifts in subscriber needs.

In this embodiment, dynamic publish subscribe is scalable. Publish subscribe is generally scalable. For example, radio stations have no upper limit on subscribers. Enforcing a re-evaluation of where subscribers are, at each "bus depot" along the route, ensures that it is both dynamic and scalable.

Supporting Big Data Trends:

Subscriber driven "winds of change" are engineered and coordinated: convoys of buses, carrying bulk pollen and their engineered clones, thus concurrently supporting multiple subscriber requests. They are like VOIP packets bound for multiple destination phones, like walkie-talkie phones. One publisher is simultaneously received by all listeners. Thus pollen and clones of them, expressly created to cover multiple consumers, board multiple buses, on different schedules and routes. The chirp routers create the clones, engineer these winds and collaboratively manage their flow.

Dynamic Destination Addressing:

In one embodiment there are "default" destination directions for pollen of specific signature types and which types of bus are designated to carry them to half-way houses at each bus stop. This is the equivalent of bus station transit hubs. Here, pollen transport is rescheduled and redirected, with community mail boxes acting as temporary buffers, if needed for performance or connectivity issues. Note that this is form of rerouting is on a need to know basis, in one embodiment. The sending router only knows enough to send the school children to first bus station address. It may not have knowledge of final destinations, which may be dynamic. The hierarchical architecture is similar to a post office hierarchy (county, city, country are the hub levels), but unlike the post office, the direction or "default" address does not convert the communication into addressed and sender-oriented transmissions. The mailing address is being inspected and changed, based on the current situation at each router/bus station, a departure from the prior art.

In one embodiment, using the bus stations as decision points breaks up routing decision to hops between logistics hubs. The current active subscriber demographics and demand may be reevaluated at these hub points. Re-routing may be needed. For example, in one embodiment, routing policies may be specified. One "policy" is in place that all GE refrigerators provide a daily health status short "chirp" "color" (e.g. red, orange, blue, purple) forwarded to GE appliance service centers/subscribers. There are four such appliance region centers e.g. North, South, East and West. Based on the chirp signature and device location, directives running in edge routers "know" which bus load this pollen/chirp has to be part of. There may be multiple subscribers so it also needs to track and route multiple concurrent buses to multiple subscriber locations. Further, at each bus stop, at each mesh node, the routers must "know" enough to "sort" pollen, at varied levels of granularity for most effective onward directions. All of this is simplified by using logistics hubs.

In one embodiment, packages are bulk shipped to one central logistics hub, which manages the dispatches to other logistic centers. If packets need to be cloned to support multiple subscribers, they are, at forks along the path, in this embodiment. Further, VOIP like chirp packet cloning is managed by the router on a need basis, where forks occur—e.g. the container was traveling north, now is split into eastward and westward "half" containers and combined with other fellow "half" eastward and westward bound containers to form a more efficient whole container. Some chirp packets are cloned because it has both east and west subscribers. Thus, efficient repackaging and pruning of routing paths, as described in FIGS. **12**, **24-25** and **27-28** address multiple scenarios including mobile devices, temporal mesh network clusters and VOIP like schedulable periodicity.

These embodiments are different from the prior art post office approach that manages IP packet routing. This routing circumvents the static address schemes used in traditional mail and email routing. It incorporates support for dynamic winds of change at each stop along the route.

In these embodiments, the destination of the bus is dynamic and so is its routing path. This is analogous to not just changing bus routing (e.g. an accident) but also end destination (e.g. hospital) based on the current situation and nature of containerized load. If the load is no longer needed, it is discarded mid journey. Round tripping, caused by static destination addresses, is avoided. Further Cloning at forking junctions ensures concurrent, low latency deliveries. The schedules for convoys of buses are adaptively managed by collaborative scheduling agents see Ser. No. 13/571,294 et al.

Further, like school children, some pollen may only board secure "yellow" buses. At each routing point, like bus logistic hubs, the containerized chirp packets, or passengers, like

VOIP packets, are steered in the right directions, as part of the bulking/routing process. Thus, the process automatically ensures that birds of the same feather flock together. The hubs are like bus stops, but the bus schedules are demand driven and the packing of the bus based on current subscriber demands.

The demand and supply is thus in dynamic alignment, despite the inherent "change" in the system.

Creating and Managing Broadcast Storms

Winds, in nature are seemingly undirected broadcast storms. As in nature, seasonal, undirected and sporadic broadcasts are tolerable if time-to-live functions are adaptively managed at the control system level. This is more than just preset time-to-live or maximum hop count values instilled in IP packet header data. Macro level system control is needed so broadcast storms cannot perpetuate.

Hence virus and spam filtering software for email services. The control system should monitor and modulate "decay" function and its PID control parameters and make changes to network topology accordingly, see FIG. **1**. An embodiment of the invention has implemented inherent Distributed Decay Functions that include time-to-live, heartbeat sequences, max number of relay hops, stop at "root" nodes etc. PID parameters and mesh topology change dynamically, to ensure network scalability and stability.

Buses are scheduled departures. Winds are a more amorphous concept. They are closer to broad agency announcement to address an Emergency Response involving sharing of diverse resources e.g. Joint Armed Forces embarking on a common mission. Convoys of chirp buses emanating from multiple locations are merging, dispersing and coalescing, driven by both publisher type and subscriber demands. They are generally directed to a concentrated location (e.g. New Orleans post Hurricane Katrina). Note that multiple such "Hurricanes" can be occurring concurrently. Clustering and dispersal in mobile mesh networks was taught in Ser. No. 11/818,899, FIGS. 27-28.

In one embodiment, known subscriber requests are handled by scheduled deliveries, with varying level of urgency, latency, QoS etc. Winds serve a different purpose. They provide the capability to support coordinated efforts in sharing communication. One of many lessons learnt in New Orleans was the inability to get different agency wireless devices to "talk" with one another. Winds address this.

An example is called for at this point. The wireless communication devices used by different agencies use different protocols and security. Information flow from entity A to entity B needed to go up to a common point C, be decoded, interpreted and then encoded again for B to receive it. For example, in the embodiment shown in FIG. **27**, contiguous devices **518**, **519** cannot communicate via mesh node **510**. They need to send their messages all the way to the top, the SIP server, see **514**. This is untenable. It is worse when the devices are on different networks. "Round Tripping" from Device A to its root node equivalents then to Device B via its root node equivalents is expensive and results in latency/jitter.

Multiple solutions have been proposed involving secure middleware for interpretation of data from sender/publisher A and repackaging it for consumption by receiver/subscriber B. Then Machine to Machine (M2M) translation occurs closer to the devices. But IP packet security is of concern Chirp routers have access to the link that connects Chirp/pollen publishers to their subscribers. As explained, since chirp device naming is not unique, like IP addresses, only chirp routers know how to provide the routing needed. The system described in these embodiments is inherently secure. There is but a single point to secure in these embodiments—the chirp router mapping/routing table must be protected from hacking Routing table access may be reachable via IP only and therefore leverages existing security methods. Chirp routing security is virtually unbreakable, in these embodiments.

Inasmuch as chirp routers are secure in these embodiments, then Interpreters may be safely installed in chirp routers. These are software/hardware/firmware. They can also be cards that fit in either the front or back slots of nightlight embodiments FIG. **22**, **26**. Event bound inter agency communication is thus supported by these embodiments. The interagency communication ends when the interpreter card is removed. Winds and other temporal relationships are now manageable. This is relevant to secure remote diagnostics and repair.

FIG. 27, from Ser. No. 12/352,457, depicts an embodiment using a real time communications network with VOIP phones. It has clear parallels to chirp devices, which send packets of data just as small as VOIP packets. Further hybrid routers are both chirp aware and IP protocol aware and support transceiver slots for both, FIG. **22**, in one embodiment.

Such routers may then, with appropriate interpreters installed, provide secure lingua franca capabilities between all types of devices in the network. The routers are also capable of real time communication and translation between IP based devices and chirp based devices. The chirp birdsong is now comprehensible to existing IP based system diagnostics and repair tools. No reinvention is needed. Chirp versions of machine independent programming languages like Java and Machine Esperanto will engender Human to Machine interaction as well as M2M communication.

Application specific graphical programming for chirp devices may make it easy to teach chirp devices and/or create new variants, FIG. **31**, Light weight OS-less chips, FIG. **30** may be directly loaded on a card, from a USB or slot in computers et al, FIG. **17**. The card is inserted in the chirp device to activate it. Removing it disables it. Disabling errant devices and/or isolating machine failures and zone management are thus simplified, in some embodiments.

Pollen Signatures

Like birdsong, Chirp signatures define the broad property of the pollen/chirp. They do not have to be unique, just characteristic of a category of "bird," some bird watchers (subscribers) are currently interested in. Chirp signatures provide one piece of the routing puzzle. Chirp routers are aggregate subscriber aware, on a real time basis. This provides the other piece to the puzzle. For example, the chirp router, in one embodiment, includes a data stores of how many (redundancy), how often (frequency) and suitable directions to engineer the buses needed to move current inventory of chirps. Scheduling buses is a collaborative supply and demand logistics exercise with dynamic alignment to prevent stacking, per methods taught in Ser. Nos. 11/266,884, 13/571,294 et al, FIG. 26.

Chirps may be parallel and/or sequential transmissions. They form signature patterns and payload patterns. The signature patterns are needed by the edge routers to categorize the type of pollen/chirp and route it to appropriate subscribers. Subscribers exist on both chirp and IP based networks.

For example, the diesel generators in a community network wish to schedule a mass repair visit. They need to share information. They can certainly send their data to the IP based customer service center in India. However, for each diesel generator that is a round trip from chirp network to service center and back. If instead, an embodiment of the invention is used and the generators are capable to chirping over a meshed chirp network, FIG. **22**, then based on chirp signatures, diesel generators can recognize each other and use collaborative agents internally to perform needed scheduling for repairs. This was also discussed in Ser. No. 13/571,294.

Signatures are thus key to both intra-network and internetwork chirp payload propagation and routing, in one embodiment. Further, chirps are also temporarily stored in community mailboxes, in transit in times of congestion or network non-availability. To get a sense of the "demographic" of passengers awaiting transit services, chirp routers inspect signatures and accordingly arrange appropriate transit services (e.g. small bus or convoy of buses).

Using phone audio "chirps" in one embodiment, signatures are a distinctive pattern of ringtones. Each chirp/ringtone is based on multiple variables e.g.:

1. Chirp transmitter type (e.g. infrared, audio, Wi-Fi)
2. Its selected frequency channel
3. Its selected Power level
4. Other parameters specific to transmitter type

A two chirp/ringtone signature has two chirps/ringtones in sequence. Each ring tone can include one or more transmitters. Even in the simplest case of on/off power levels, the two chirp signature has four distinct states (0-0, 0-1, 1-0, 1-1). The device is therefore capable of three distinct non trivial (not all zero, or silence) states for each transmitter. (2^2-1). In general, the number of possible signature sequences for simple on/off transmitters is (2^M-1), where M is the number of ring tones in the sequence. Hence a three ring tone signature sequence has 7 distinct non zero variations (2^3-1).

In general, for P total distinct (as seen by receivers) power levels, including the trivial case P=0, M number of chirp sequences, there are (P^M-1) non trivial signatures per transceiver. Further, if each chirp also has F frequency channels/tones/chirps to choose from, then the number of distinctive, non trivial, signatures/states per transceiver t are:

$$S_t=(F_t*P_t)-1)^M \text{ Where:}$$

S_t: number of distinct signatures/states for transceiver t

F_t: number of frequencies/channels/ringtones/light colors available to transceiver pair t

P_t: Number of distinct (as seen by receiver) power levels available for transceiver pair t

M: number of ring tone/chirps in sequence.

In an N-Logical wireless transceiver ("radio") framework, with multiple simultaneously operating transceivers (see FIGS. **7-9**), there are a total of

$$S_{total}=S_1*S_2*S_3 \ldots *S_t$$

Where:

S_{total}: Total nontrivial unique signature "tunes available with t active transceiver pairs.

Thus a smart phone ringtone sequence of three, each of which has 10 ringtones can generate $((2*^{10})-1)^3=6859$ unique non trivial signatures/words. Adding another form of radio frequency communication in the system, a smart phone with camera flash, operating simultaneously as a single on/off complement, doubles that number, in one embodiment.

Further, a restriction may be placed that the sequence contains no silent chirps. Even with just 1 power level ("on") there are $(1*10)^3-0=1000$ unique, non-zero, 3-ringtone signatures available. If each signature denotes a "name" then a smart phone is capable to addressing 1000 unique, audio ringtone aware, chirp devices in its "network."

The ringtone concept may be applied to other non-audio frequency domains. The ringtones may also include multiple simultaneous send/receive on diverse "channels" (e.g. infrared and audio), in some embodiments.

The composite ringtone is analogous to a musical chord: it's a "richer" tone. The chord carries more information for multiple uses e.g. redundancy, error correction and multilevel security, denoted by number of chords and receiver capability to receive them some or all the tones.

They may also be used to provide "hidden meanings", like secret handshakes, nods or winks. The "public" message may be simulcast with a "private" message decipherable only by devices with capable receivers. Thus, enhanced services may be provided to select customers, by providing them the appropriate chirp equivalent of a TV cable decoder that, with installed firmware, provides access to more TV channels in the "bulk" broadcast.

Different parts of the complete message may also be emanating from multiple chirp devices. For example, in one embodiment, ring tone chords from different devices provide two-level authentication system capabilities. Thus a police car will not start unless the policeman and his partner are safe in the car, determined by synchronized chirps from both their smart phones. Further the phones are secure in that only the authorized users have the correct access codes so stealing the phones renders them ineffective. Thus, the equivalent effect of multiple signatures on a two signature check is supported, before the car will cooperate. Chording thus provides multiple security layers.

Chirps are a Generic Concept.

Ringtones are one example, easily available on smart phones. Another could be color LEDs e.g. traffic lights have 4 states e.g. red, orange, green and black (off). Complex hues of a color (e.g. red+blue=purple) may be chirped across from devices to smart phones. Thus, like musical chords, chirps can be "rich" in information even if it is a short, unobtrusive burst. With the appropriate image or sound processing software, smart phones and other computers are thus multi-lingual devices, capable of translating chirp chords from Wi-Fi, Infrared, Audio (ringtone) and light patterns (camera flashes and photo image analysis).

In one embodiment the smartphone software consists of one or more apps. The apps are loaded on these computing devices also provide translation mechanisms to understand what was said in chirp languages and what to do with the payload data. The chirp signature, like a bird chirp, is essential for this translation and categorization—it tells us what type of bird is speaking Note that the apps were downloaded through secure and authenticated sources via standard IP.

If it is useful to have birds of a feather flock together, then chirp signatures may also share some chirps common to the flock. Thus, based on the signatures, higher level systems, such as the embodiment of FIG. **26**, know where the chirp devices of type "Yellow bird" are and their states. Demographic data is readily available.

Decentralized Naming with Inherent Conflict Resolution

Chirp routers employ in one embodiment the chirp equivalent of the distributed DHCP server based IP addressing scheme with inherent conflict resolution, as shown in the embodiment in FIG. **29**. Using those methods, instead of IP based device addresses, devices are given names, randomly chosen, but with inherent conflict resolution. These names can be changed frequently by the router for all but the simplest "blind" (no listening) chirp devices. Blind devices, in one embodiment, like garage door openers, have simple means for identification variation, such as DIP switch settings so their timing and/or channel are modifiable by the user, as directed by the routers governing the devices.

Lineage Based Uniqueness.

In the chirp network of one embodiment, there are no pre-assigned unique "names" or "signatures" as existent in the IP world of MAC-IDs or IP addresses. Those approaches require a central authority to manage conflicts resolution is address values. Sender-oriented communication packets need unique destination addresses. In contrast chirp device names, related to their chirp signatures, are assigned or modified from factory default, if needed, when the device or "thing" first joins the local network governed by a router. In the event two devices have the same factory default signature, for example, one will be changed, remotely or manually, as described in the garage door opener analogy. Thus devices and its siblings have unique names or "addresses", as seen by the chirp "access point" they are connected to, in one embodiment.

Sibling names must be distinct but need not be unique, in one embodiment. If a device name is distinct, amongst its siblings, then tree based logical routing is sound. As an analogy, Eric, child of Paul, child of James is distinct in a tree based topology from Eric, child of James, child of Paul. The routing table entries and routing paths for the two Eric's are distinct: James→Paul→Eric and Paul→James→Eric. Note that in this approach, the "lineage" is exploited to provide context/delineation between two devices of the same name and in the same network. This form of identification, not needing unique names/addresses and using lineage to provide distinctions, is a departure from conventional IP based networking, where each MAC-ID "name" or IP "address" is expected to be unique. Methods taught in Appendix A, however, still apply and are equally relevant for route path management. Lineage based device naming are inherently secure because only routers know which Eric is being addressed.

Small Names:

With non unique names, the number of "ringtones" in a signature sequence can be small since we are no longer striving for uniqueness. The same three ring tone sequences may be used repeatedly in different, non adjacent sub trees of the network, with no adverse effect, as long as lineages are distinct. When they are not, in one embodiment, chirp routers will either first attempt to change the name (for embodiments where the chirp device name is programmable) and otherwise will notify the user to change device signatures, (e.g. dip switches for garage doors) or move them to another chirp access point/router.

Name and Signature Swapping

Many not-blind chirp devices take direction, in one embodiment. That direction comes from routers and tune directives from trusted "mother" agencies. When a chirp signature or chirp language/protocol has been compromised, the device may be taught different birdsong. "Mother" chirp directives may rename a device and switch from one language to another. The languages may be closely equivalent e.g. dialects of a purpose built machine Esperanto. Changing names and word "look up tables" thus provides additional enterprise security.

Chirp Data Transport is Inherently Secure

Receiver Oriented:

Only routers can provide the link that connects Chirp/pollen publishers to their subscribers. Further chirp device naming is not unique, like IP addresses. Therefore names can be changed, for many listen-capable devices. This receiver oriented system is inherently secure, in one embodiment.

Further, in the case of a bulk broadcast, app. Ser. No. 11/266,884 teaches methods for VOIP aware phones capable of deciphering the message for them and discarding the rest. As such, the bulk container is like mixed and jumbled bag of pollen. Each flower/subscriber takes what it needs and ignores the rest. Thus, multiple messages for multiple devices may be sent but only the device it is intended for can decipher it. Chirp transmissions are inherently closer to pollen in that they may also be widely broadcast without sender side security layers. This reduces encryption overhead significantly since the "secret" is known only to the intended flower/recipient. For example, in one embodiment, a chirp of an error code is not encrypted inasmuch as it is meaningless without access to the secure routing table.

In another example, humans have multiple forms of secret data in handshakes, nods and winks intended for specific audiences. Certain groups have a secret handshake that provides intended recipients with additional information. Neither party shaking hands may have prior knowledge of the other. They are shaking hands in broadcast mode, visible to everyone. In fact the person initiating the handshake may not even be aware that he is communicating special information—it is simply the way he always shakes hands. Like pollen, the security is "genetically encoded." The onus shifts to the recipient to decode the message, based on secret signatures they were taught to look for. Thus security infrastructure requiring encryption at the source is no longer essential, since the message signature is already encrypted and only intended recipients can decode it. In other words, the publish/subscribe broadcast may be "open" and hence lightweight.

Hidden Meanings:

In one embodiment, how communications are understood may depend on multiple levels of security. Adding more transceivers, operating in diverse frequency bands, is an effective way of sending partial messages, and is used in one embodiment of the system.

Consider three sibling devices in a network. The first has infrared capability. The second has a microphone. The third has both. A three element ringtone sequence is simultaneously sent from a fourth dual transceiver device, which has both infrared and sound transceivers. It is interpreted differently by all the three receiver devices. The first two will get only partial messages. Hence, with multiple transceivers, messages with multiple layers of meaning may be transmitted concisely. Further, messages being sent on independent channels may be syncopated in time, making it harder to decipher. Lastly, both signature and payload are "tunes" and flow into the other in continuous transmission. Only the receiver knows where one ends and the other starts. The secret handshake is recognizable only by intended recipients.

Further, assume the intent is to deliberately obfuscate the "signature" being transmitted. Hence decoy signals are be sent on one or both "channels," in one embodiment, that are being watched. Only devices that know the secret handshake, can piece the "real" message together, removing the decoy components of the transmission.

This is different from frequency hopping techniques. In frequency hopping, sequences are on different frequencies and the receiver knows when to change frequencies based on a mutually known sequence order. Here, each "tone" in the ringtone sequence consists of involving multiple transmitters in simultaneously providing layers of signature security. Thus different devices receive different messages and meaning. These chirp equivalents of musical chords are hard to decipher fully unless you know the entire "tune" being sent on different transceivers. Further, syncopations in time—so multiple tunes can be sent, then silence, then another set of tunes, adds complexity. Decoders put it all together, taking note of silence. Silence durations also are cues, like nods/winks/inflexions.

Some Decoders can access the entire information, others have limited access. Thus, despite their simplicity, even smart phone ring tone chirps are difficult to decode, especially when there are multiple concurrent independent transmissions. See embodiments shown in FIGS. **7-8**, **23**, which are the equivalent of musical chords.

Existing temporal key management schemes used by some embodiments further improve signature and payload security. Multiple existent means for encryption and security exist and are applicable. However, recall that some chirp devices are intentionally simple and have relegated storage and computation elements to chirp aware routers (e.g. night lights, smart phones, etc). Hence their ability to decode is maintained to be non-computationally intensive in one embodiment, the chirp equivalent of a short ringtone pattern which can be deciphered or transmitted easily. For example, a simple three tone ringtone and a three bit Boolean bit mask suffice in one embodiment.

Changing the bit mask, using "dip switches", for example, can change the bit mask and therefore signature "tunes" that the chirp device will listen for. In the example of changing dip switches for the garage door opener, the owner changed the signature "tune" for his chirp device.

There are no limits to the complexity of chirp signatures. In mission critical or enterprise level security, three ringtones may not suffice. Sophisticated decoding software solutions and integrated circuits are supported in this framework for some embodiments. The night light (or other devices) have a removable insert that contain security chip decoders, in one embodiment. See FIG. **22**. Ser. No. 10/434,948 teaches methods to build secure, OS free chips, FIG. 30-31.

Chirp broadcasts emanating from chirp routers contain multiple signatures and payloads. Chirp routers may send them as single tunes, intended for one device, or a container, intended for multiple clients of the chirp network, akin to the VOIP container described in Ser. No. 11/266,884, 11,088, 330, FIG. 12.

Consider security in these typical scenarios supported by embodiments of the instant invention.

1. Single tune for single chirp device: In one embodiment there are three chirp devices on a local network (wireless or wired). They both have three ring tone signatures and five tones for payload data for a total of eight ring tones. It is unclear to snoopers, which machine is being addressed when a single eight ring tone tune is sent.

2. Bulk container for multiple chirp devices. Note that signature ring tones and payload ring tones are indistinguishable without some knowledge of the devices communication needs. Further, the order in which the tunes are concatenated to produce a "bulk container" is immaterial to the devices. They take what they need and throw away the rest (like pollen). But it is obfuscating to snoopers. Hence randomizing the order in which tunes are assembled into the bulk container is a simple yet effective security measure. Further, in one embodiment the entire (IP based) container is be encrypted by temporal keys using well known IP based encryption methods.

3. Obfuscated Mode: In another embodiment, a third chirp device is added, a five ringtone, fifteen ringtone device (a total of 20 ringtones). It is clearly distinguishable from the 8 ringtone devices. However, like pollen, devices take what they need and throw away the rest. Hence the chirp router, as a decoy mechanism may send out 24 ringtones, with irrelevant packing at the end of the 20 ring tone data. Now, it is unclear whether it is a bulk mode transfer for three eight ringtone devices (3*8=24) or one 20 ringtone device with padding. Even if the snooper knows the types of chirp clients in the network, the secret handshake in the "pollen" remains secret, decipherable only to the initiated.

More complex forms of encryption are possible by using the ASIE Information Elements in the IP based wire-less radio beacons, in other embodiments. There are embodiments using both an IP layer encryption and Chirp layer encryption. In different embodiments each of the layers has temporal keys which are periodically reissued. The chirp layer is not burdened by a preexisting, public protocol. Transmission encryption and other security are arbitrarily simple, complex or nonexistent, in different embodiments. FIGS. **19-20**, **30-31** relate to methods previously taught to remotely manage secure devices.

Where Chirp devices reside inside a local network is not public knowledge. Their routing is a function of both the IP address of their parent chirp node/router and chirp device signature.

Chirp Signature, protocols, internal routing information is accessible only within the local area network and restricted access to authenticated members only, typically router embodiments that manage the devices e.g. night light, smart phones et al.

Temporal Names

In one embodiment, routers may employ a distributed DHCP IP addressing scheme with inherent conflict resolution, FIG. **29**. They are capable of autonomously changing their current IP addresses and privately communicating this way within their local IP networks. They are also internally capable of changing IP addresses for their IP clients and those IP to chirp tunnels that connect chirp devices to IP addresses. Finally, they may also request listen-capable chirp devices to change their "names" and even teach them a new equivalent vocabulary, in one embodiment. Thus a traffic light swaps "green" for "red"—only the router or its agents knows that red now means go.

If an IP based tunnel/socket needs to be established for real time interaction with a chirp device, it is the router, using DCHP, that makes a temporary IP connection available, per methods taught for temporary and mobile mesh networks, see Ser. No. 12/352,457. Thus all "contact" is router managed and the router is secure in several embodiments of the invention.

Pollen is moved based on its chirp signature or its explicit naming of an agent/flower. Where it goes is managed by the router. Only the router network knows both pieces of the puzzle (sender "color" and subscribers for those "colors"). Further, in a receiver oriented world, only the destination address (e.g. Agent-ID) need be known to the router. It cannot decipher the contents. Only subscribers/agents at the final destination know how to unlock the pollen's message.

For enterprise level chirp security there are multiple alternatives. Temporal ring tone sequences, generated by the routers, may be added to chirp signatures. At each hop they may be replaced by another (thus only adjacent routers know each others keys). Or network level keys are distributed, by IP, from the Enterprise and their agents. They are known only to those in the trusted network and are periodically changed. Without this additional key, all chirp communication is unintelligible. A simple 2 or 3 light flash or ring tone sequence, sent by the routers, thus suffices in one embodiment. If the routers are using beacons to inform chirp devices of their presence, then this data could be in the beacon chirp. Since it is a temporal key, hacking it has limited value. Further, the chirps do not need to be encrypted on a packet basis. Container level encryption may be sufficient, for those embodiments, thus reducing security key distribution complexity. This reduces the complexities of large scale security key distribution and its management.

Secure interpreter cards offer a portable form of device security for one embodiment. Machines chirp terse status and error codes. These are received by a meshed network of night light routers or their agents. Removable Interpreter cards may be installed in the slots, FIG. **17**, **22**, **26**. Chirps can be interpreted and translated at multiple locations of the logical mesh network tree, including at the router itself. Further, the cards and their agent may have limited access privileges to the chirp data. Hence multiple agents/cards may be needed to get the full message decoded.

Unauthorized entry into the router also changes very little, unless you can permanently change where it gets its subscriber base map. But this is periodically supplied by the Enterprise, heavily encrypted from the IP side of the chirp router/bridge. "Dongles" inserted in "back" slots of routers, FIG. **17**, **22** are embodiments that are highly sensitive to security.

In one embodiment, secure software upgrades to chirp devices and the router follow the same method taught in Ser. No. 11/088,330, see FIG. 18, 19 and in Ser. No. 10/434,948, FIG. 30,31. Dongles may be used in the router slots or USB hubs to provide access only when inserted.

An interpreter card, in one embodiment, is inserted in one of the many slots (front and back) of the night light or router, FIG. **16**, **17**, **23** **26**. When leaving the home, the end user removes the cards. Note that in one embodiment the "card" could also be a proximity chirp from an authorized device e.g. smart phone or dongle on the end user's key chain. When that chirp is heard, chirp routers are active.

In one embodiment, the same security card is used at a second location, such as at the end user's office, to perform similar interpreter functions. Thus secure communication is portable across home and work environments. And, in some cases, the smart phone suffices as the "card.", engendering secure voice and data communications, authenticated by smart phones at both end. Since the mesh nodes/routers also manage regular IP traffic, both types of devices are accessible to the card, for secure inter-device communication.

In effect the interpreter cards are used as a common home/office entry key in these embodiments. Further, the same card may have a separate section that filters out chirps from unfamiliar or unauthorized devices. It is then also acting as a firewall for both chirp and IP devices. Multiple cards, working together, can support complex collaborative efforts, involving both humans and machines, FIG. **26** and described in the associated application.

Extensible Chirp Vocabulary

Humans generate generic concepts, based on words strung together to form sentences. Concepts are complex and to communicate them human conversations can be tediously verbose—hence the adage: a picture is worth a thousand words. Machines, in contrast, are purpose built. Their "vocabulary" is limited. What they wish to convey is purposeful and terse. For example, automated traffic lights can effectively communicate with just four states (red, orange, green and black for inactive/malfunction). Chirp device transmissions are also designed to be terse. They are intended for purpose-driven communications e.g. Machine-to-Machine (M2M) communications.

The stringing together of complex concepts is a sequential process. Hitherto, prior art in networking has focused on Human-Machine communications. They were designed to support verbose (large packet) communication, but at the price of inefficiencies for small packet transfer (e.g. VOIP). At an abstract level, all IP based communication is essentially faster Morse code.

Machines, in contrast, may also have a lot of data to report—like a core dump—but this is the exception, not the rule. Most of the time simple red, green or orange "chirps" suffice. Further, since machine states are limited, small vocabularies are adequate. For example a color based language can concisely communicate many shades/hues/meanings tersely: complex shades of red sent from a single LED can communicate a lot tersely/swiftly, without lots of words.

Thus Chirp device communication may be short (for coexistence) but rich in meaning. In some embodiments, the devices are also capable of faster parallel communication. A data byte is a sequential ordering of 8 bits of data. A serial port transmits 8 pulses sequentially. An 8 LED parallel port, on the other hand, "chirps" once. Advanced LEDs or audio devices therefore transmit, in parallel, large packets of data in one chirp. The chirp bandwidth is dependent on the resolution of transceiver to send and decode the "parallel" port data.

On major drawback of quick parallelized data flow is it can be easily missed. TCP/IP like protocols may be implemented to resend data but it defeats the objective of coexistence with verbose (sequential data) IP dance partners also operating in the same spectrum. Instead, like birds, repetitive chirps ensure that at least a few chirps are heard. Also, like pollen, the embodiments of this system err of the side of sending more repetitions in the hope that at least that a few bear seed. This implies, like Allergy season, a timed broadcast "storm." Further, like pollen, the data is receiver oriented, so, like bird chirps, everyone can hear it but only intended receivers can decode it. Pollen can travel light.

Chirp vocabulary is driven by what the devices wish to convey and since it is receiver oriented, it can be as simple or complex as needed. However, if chirps are using the same frequency spectrum, these chirping "radios" must broadcast short bursts, repetitively and randomly. "Chirp aware" devices, unlike their advanced Wi-Fi and Bluetooth cousins, emulate limited agile/polite behavior.

"Small" Data Feeds Big Data

Chirp sequences (in parallel or serial flow) form "tunes". Tunes are used as signature patterns and data payload. Or a concatenated and encrypted version of both, where encryption includes delayed transmission as in syncopation. In one embodiment, two tunes are really a jumbled version of one secret handshake, where even the silence may have meaning, known only to intended receivers.

While humans can hear birdsong, the chirp sequence "meaning" is known only to the birds. We can draw conjectures but since signatures and payload are both tunes, it is unclear where one sort of tune melds into another. Hence humans can hear all the myriad bird conversation in the park and yet understand none. We have not been provided the secret handshakes.

Bird chirps respond to changes in the environment. For example, a cat walks through the park. Our eyes follow it. Our ears notice how the chirps follow the cat's motion as it moves from one tree to another. Chirp tunes will change both in the sequence of tones and their intensity. We therefore, as snoopers, may be able to discern activities common to the same consensual domain because we are matching patterns in two different sensor domains (eyes and ear). We are putting two and two together. Multiple sensor fusion drives our inference engine.

Over a month, the cat may visit different parts of the neighborhood. There are trend indicators there but the sampling duration may need to be months to accurately pin point "affected" regions. The quantity of data to be analyzed is considerable. Some may need to be stored and reviewed later by the big data analysis engines that are predicting trends based on past history.

Over time, it is noted that these "small" data pattern repeats itself around dusk most nights. "Big" data engines may then infer that a nocturnal animal (e.g. cat) is causing "disturbance" in the "reference signal Thus "small" data, un-intelligible to us, is processed into more coherent form, which in turn is used to draw conclusions about the environment not transmitted per se in each "small" data transmission.

Putting small events together to infer a complex event or trend is difficult. It may require a control system component, Bayesian reasoning, to filter out the noise from reference signal disruption. This is taught in collaborative framework, see FIG. 26, Ser. No. 13/571,294 et al. Further, 61/615,802 teaches techniques for detecting deviations from reference signal patterns and reporting them for further processing to higher layer functions. "Small" data events, based on observation, are thus recorded and sent "up" for "big" data analysis and action. In embodiments of this system, small events feed complex event analysis.

The night light routers of one embodiment of this invention are like node branches in a park of other trees and birds. The router hears chirps of birds in its network of branches. At one level the router simply has to know which subscribers care about which chirp signatures and arrange for buses or "winds" to carry the chirps to them. However, it is also a hub for its bird subscribers. Further, it provides disembodied intelligence and storage for these simple devices—like holes in its trunk for nests. Each tree and its branches form a root and relay mesh network. Each node in the chirp network mesh tree is a logical first level filter for interpreting small data birdsong.

In one embodiment, Chirps include signature (e.g. like device names) and payload. Payload and signatures are melded to form tunes, in one embodiment. Tune payload may include short directives (scripts) sent as series/parallel pattern of chirps e.g. encryption, security signatures, data payloads, commands, data requests, software upgrades or real time diagnostic conversations. Like VOIP phones, chirp devices and remote customer support centers establish a real time tunnel to talk in secure chirp languages, in on embodiment. In other embodiments, they even switch languages and continue the conversation easily—recall that machine states are terse and purpose built—their vocabularies are concise and therefore can be remapped to other "colors" or ringtones easily.

Chirp directives are called tunes because in a receiver oriented world, only specific devices can "dance" to the "tune"—those that know the secret handshake. Tunes are (secure) agents that know bird speak. Further, end devices are purpose built and OS-less, FIG. **30**-**31**, in one embodiment. They dance only to specific tunes from "mother." In some embodiments, the system components may be capable of "changing their tune" (e.g., mode of operation) based on directives from the "mother" or "mothers." At any point in time, however, these tunes are ant-like collaborations: simple robust stimulus-response pairs. Like traditional get-set protocols, each tune has a specific dance partner.

Tunes typically perform simple tasks. They are ant like in abilities—a tune is not complex construct like a song. However, working together, they perform complex tasks, or complete songs, like super organism e.g. ant hills.

These tunes/agents, intelligible only to authenticated chirp routers, may also contain application specific inference engines as described in Ser. No. 13/571,294. In one embodiment, they are interpreters for encrypted "short hand."

For example, in one embodiment, the kitchen night light router is directed to forward only exceptions: no news is good news. An installed tune states that, as long as at least one chirp is heard and it is not a red flag, the router does not forward it. Another installed tune/ant, for another subscriber, states that, at midnight, 24 stored chirps should be sent, so it may "plot" the hourly pattern. This "tune" uses local storage and time stamping, heart beat sequence numbers etc., see Appendix A.

Other ant-like tunes, in other embodiments, may include post office like services: hold some mail for a period of days but forward others, discard others etc. A variety of conditional transport mechanisms are thus supported. QoS equivalent tunes may drive bus schedules and polling frequency, based on pollen types (chirp signatures) and their subscriber size, urgency and interests. Note that all the heavy lifting is at the router—the end devices are still thin client.

At a more macro level, in other embodiments using distributed collaborations, FIG. **26**, may include collaboration from ant-like tunes operating on other routers. For example, the a router having a battery backup, the kitchen night light notes no chirps were received from kitchen appliances in the last 24 hours. In this embodiment, the router is aware that it is running on battery backup and that chirps were expected at regular intervals. Before filing multiple repair request reports, a collaboration tune request that the kitchen night light confer with the living room night light embodiment, a Samsung TV via the power line network. The Samsung TV confirms that its devices are fine. Bayesian Inference and causal reasoning engines, correctly infer that the kitchen fuse has blown. If so, further causal reasoning indicates that perishables in the GE refrigerator are suspect. The home owner is informed via his smart phone. No repair reports are made, other than to notify of a potential overload condition. Thus, application specific intelligence at the chirp router level reduces customer support overhead and false alarms. Simple ant-like collaborative agents, working together engender complex reasoning from simple messages or even the absence of simple messages.

Multiple tunes thus support dynamic and diverse needs of multiple subscribers, without unduly burdening the network, since only requested exception packets and/or clones of packets are sent.

Further, packet cloning along a route is managed by each router on a need to know basis, in one embodiment. Where a fork is needed is event based, since the publisher-subscriber relationship is dynamic and transient. Subscribers may direct a Mother (root) node to remove some of its publishers from the "follow" list. Some, en route, will be removed at the next bus stop. Thus a dynamic bus schedule based supply and demand alignment is constantly taking place—e.g. containers traveling north as one convoy, now split into eastward and westward containers and combined with other fellow eastward and westward bound containers to form a more efficient bus convoy. Thus, efficient repackaging and pruning of routing paths, as described in FIGS. **12**, **24**-**25** and **27**-**28** address multiple scenarios including periodic chirps, mobile device communication and temporal mesh network clusters.

In a subscriber driven supply chain, QoS etc is dynamically defined by the subscriber and their agents (e.g. Tunes) Recall there is no static QoS value inherent in chirps, vs. for IP packets. Chirp pollen are simply directed to the appropriate "yellow" buses, from one bus station to another along their route to subscribers, Only Chirp routers within a local network know bus schedules and/or if a chirp packet was placed on a bus. Further, this type of data may be accessible only via secure IP based protocols, Malicious or malfunctioning devices, either chirp or IP, cannot manipulate device routings because these are event driven and dynamic. Rule driven event based systems are hard to "crack"—explicitly stated bus schedules don't exist, so cannot be "stolen" or manipulated.

Rich Chirp Streams:

Routers are intermediaries between "small" data publishers and their "big" data consumers (machines and humans). In addition to routing, authenticated routers serve as distributed "big" data agents. Interpreter "tunes" correlate patterns and map different types of "events". Big Data tunes, resident on the routers, monitor chirps and correlate "cat events" to corresponding birdsongs/tunes in their tune pattern library.

In one embodiment, the same small event feeds multiple complex event analysis and prediction engines. No one single small event is significant (and can be missed), but a swarm of them is a noticeable trend. The interpreters at the router nodes are data sieves/filters and in effect are also virtual "rich" chirp publishers to subscribers above it in the information food chain—feeds on data to generate actionable intelligence e.g. trending.

The multiple small event chirps may feed into an interpreter that chirps a simple yes or no, terse but rich" in content. This data "richness" is possible with interpreters residing at edge routers (both root and relay nodes). Inference Intelligence is distributed with collaboration between the layers, all the way from edge routers up to core routers—big data analysis systems. A hierarchical, distributed tree based collaborative control system emerges, as the embodiment shown in FIG. **26**.

Communication can thus be both terse and rich throughput the collaborative ecosystem. It can be sporadic, intermittent or periodic. All of this drives control signals in an adaptive distributed communications and control network, FIG. **1**, **26**.

Some chirp devices, in one embodiment, use rich chirp shorthand, periodically, like VOIP packets, to communicate periodic updates or participate in a remote diagnostics session. Sometimes the time delay between transmissions matters (e.g. why a Mars rover needs onboard intelligence). Moving number crunching and interpretation closer to the source also ensures that context is not lost in translation in a rich chirp to its interpretations along the tree. There is less traffic on the IP highway and also more "relevance" and context to a chirp closer to its source. Given that in embodiments of this system, intelligence is clustered in routers—end devices are a thin client or even an OS-less one. This is win-win at multiple levels (e.g. lower costs, collaboration ease).

Consider video surveillance. Today, IP based cameras forward raw feeds to a central location. This is thin client but the data is not "rich." In an embodiment of the invented system a video surveillance systems include smarter cameras connected through a chirp mesh network. Each camera has been taught what an intruder pattern looks like, from its own chirps and those of its adjacent mesh nodes. When any camera detects an intruder, it sends out a broadcast chirp. All cameras in the community network are alerted and "follow" the intruder, based on taught entry and exit paths in the building and their locations. Finally, a night light embodiment receives the salient footage from each camera, from their mail boxes, in one example. The router contains the substantial software and hardware to assemble these simple messages and present the information to the alarm system or to the human operator. The end result is a pieced together video that takes up where the last camera left off. This is forwarded to subscribers (e.g. Police).

Thus rich surveillance chirp stream overhead is significantly less than central monitoring of multiple "dumb" cameras requiring the capacity to support raw IP video feeds sent centrally.

Further, only exception handling is being sent over IP, all internal communication is chirp based, local and contained, in the present embodiment. If a broadcast storm does occur, it stops at the mesh tree nodes (routers) which containerize the chirps.

Thus turning off errant zone of "things" is managed at the nightlight parent node for client chirp device relays and device sub trees in the logical mesh network tree, FIGS. **9**, **22**. Note that the nightlight is multi-lingual and messages to turn off/on chirp devices are not decipherable by those "rebel" chirp devices—IP based encryption is foreign to them, they have no IP stack. They also have no access to IP transport except through the night light.

In one embodiment, these chirp streams are stored in community mail boxes on routers. There, they await the next bus for their journey to subscribers. The chirp is terse e.g. yes, the cat has entered my region and this is the cats current location. Big Data systems take that "rich" chirp stream, of small "events" and interpret it to make sense of complex events. This is made easier when Big Data agents have defined the "tunes" directives that end devices are dancing to. They know exactly what is being said.

Thus rich chirp streams, periodic, or sporadic, are sent to multiple subscribers, each of which has different and possibly very diverse interests and "tunes" Further, demographics are complex—GE or Samsung refrigerator chirps are international. Over the IP network, chirps are available for consumption and complex analysis, anywhere.

In a collaborative ecosystem of one embodiment, FIG. **26**, multiple "rich" chirp events are concurrently feeding multiple big data analysis engines providing layers of intelligent analysis. Further, in a trusted network of one embodiment they are also privy to each others findings and interpreters to arrive to larger trends that affect them both. Symbiosis is mutually beneficial. Thus, it may be discovered that GE refrigerators are being bought by people who own Samsung TVs—perhaps a common customer support center is mutually advantageous. Further GE and Samsung may collaborate so their chirp signatures and vocabulary/tunes do not overlap—thus reducing the need to change frequencies (e.g. the garage door opener dip switch setting changes). This would improve Customer Satisfaction, a shared objective.

Returning to the video surveillance embodiment, with distributed intelligence, weak points in the perimeter security are identified for future repairs at the same time that the police department is notified. Thus rich chirps, in this embodiment, drive complex events and result in multiple responses. Supply chain logistics efficiencies are extended by interpreting rich chirp streams running closer to the sources. The Nightlight embodiments are logical hubs.

Centralizing intelligence at hubs reduces the cost of purpose built chirp like devices yet together, device and nightlight provide the same functionality as more expensive general purpose devices. Thus an infra-red camera, purpose built to cover a perimeter, is lower cost than a generic purpose raw video camera. The purpose-built camera is ant like, but with collaborative agents, equally competent.

Returning to the birds in the park analogy, consider one bird atop each tree, responsible for reporting intrusions in assigned regions. The infra-red birds chirp periodically, a heart beat, stating they are alive. It is low power, sufficient only to be heard by a "buddy" bird, charged with reporting an exception error in case its buddy dies. Thus co-channel interference is curtailed. Adaptive Power Control methods are described in Ser. No. 10/434,948. Chirps are local and "contained."

Continuing the example, an intruder is detected. Like birds in the park, chirp volume and chirp effective range increase. It is propagated by repeater relay nodes and reaches router/hub, FIG. **22**, Tunes convert chirps to English and inform humans. Thus low cost, purpose built ant like rich chirp devices can provide zone based security at a fraction of the hardware cost of more generic IP based devices. Further, like buddy systems, one device can watch over another's domain and provide sleep and wake-up tunes. Thus changing of the guard is also supported in a not "always-on" world. Purpose built chirp devices, like ants, can provide ant hill like complexity, at costs lower than their generic, IP based cousins.

Managing Blind Chirp Contention

Using methods taught in Ser. No. 13/571,294 et al, in one embodiment, the routers are like school teachers, managing the unruly chatter of school children. Some are "chirp aware" and self-directed to be more polite and agile. They need less "retries." Other simpler devices are chirping blindly. Being random ensures that deadly embraces are avoided by two devices chirping at the same time. However, blind chirpers, like unruly school children, may cause temporary contention with others, sharing the same classroom.

Randomness reduces the chances of the blind chirpers creating habitual noise during periodic bulk transmissions, as shown in the embodiment of FIG. **24**. Further Improvements to reduce blind chirp interference include the following approaches in some embodiments:

1. Move blind devices to another "band" e.g. from Wi-Fi to Infra-Red, FIG. **22**. This reduces blind interference but requires that chirp based mesh nodes and routers also support multiple "bands" This improves versatility regarding device support. It also increases capacity and bandwidth. Duplex communications are also now possible on the two separate channels see FIG. **7**,**8**,**9**.

2. Upgrade some chirp devices to "listen/see" capabilities so they are less clumsy on the transmissions dance floor. Application Ser. No 11/266,884, for example, teaches methods related to listening for beacon prior to transmission. These beacons contain Application Specific Information Elements (ASIE). ASIE provide cues for when best to chirp, for example in one embodiment. Further, chirp routers and access point equivalents can tell their clients when they are available to listen to them and/or request silence them in the interim. Like police car sirens, they send a "silence" command, requesting that devices be silent till a mission critical transport has completed. Chirp routers may also request clients to "sleep" and then provide a persistent wake up clarion call when they are ready to receive their data. Variations on these themes are taught in Ser. Nos. 11/266,884, 11/088,330, 11/818,899.

3. Provide simple accessible settings. Garage door openers in cookie cutter neighborhoods occasionally can open other homes unintentionally. The owner changes the dip switch settings on their controller, thereby reducing this unintentional interference. In one embodiment, Chirp routers direct human to modify factory default settings on chirp devices, if needed. This was addressed in Ser. No. 13/571,294 and its references.

Thus, devices, like school children, need not all be polite/agile "chirp aware." If they are capable of receiving simple "stop" or "go" instructions, the edge router school teacher is capable to managing the interference environment proactively. Scheduling the interactions may be as simple as round robin scanning techniques described in Ser. Nos. 11/818,899 and 11/088,330, see FIG. 13-17. Further, chirp aware devices, like good students, take charge of unruly ones, in one embodiment. Adaptive Power control is used in one embodiment so blind chirpers "whisper" their chirp/states to more sophisticated neighbors, who, like good neighbors, propagate their chirps in timely fashion, potentially bundling it with their own. Thus buddy systems between polite and unruly chirp devices help maintain decorum in the classroom.

Chirp Aware Devices

Very simple chirping devices, like Infrared LED based TV remotes, don't need to chirp constantly-their human is closing the loop for them. One short burst of chirps is sufficient for a button press.

More complex chirp devices may pack more data into each short chirp. Consider an 8 LED parallel chirp, conveying 8 bits each chirp. Chirps may thus be short and yet "rich" (e.g. parallel 8 LED chirp has 8× more data in a single flash than a single LED flash).

These "rich" chirps eventually need to be converted to digital sequential bit based data packets, to travel on the IP based section of the logical network tree, There are tradeoffs between using one 8 LED chirp flash (fast and less obtrusive or collision prone) or a laborious sequence of 8 one LED flashes (slower, more obtrusive or collision prone but uses simpler firmware). If Serial Chirp devices chirp often, they should be polite: more collision or chirp aware, choosing chirp times wisely.

As an analogy, sparrows chirp frequently and blindly. Wiser birds, like owls, listen and wait till "winds" are in their favor and there is less "noise" or collisions from other birds. This improves their signal to noise ratio, reduces transmit power, increases effective range. Fewer repeats are needed. A few owl hoots, in the collision free silence of the night, propagate effectively because the "Chirp Aware" device times its chirps intelligently to occur in collision free time zones.

Further, in embodiments where chirp devices can listen and be directed, the parent router may direct them when to chirp, thus supporting reservation time slots. Lastly, devices chirp may be assigned sequence numbers so broadcasting is managed, using the sequence numbers to avoid flooding. Appendix A teaches relevant methods to improve chirp propagation using heart beats sequences. Thus as long as chirp devices, like school children, can follow direction, they do not need to be intelligent or aware of the transmission status of the entire system.

Some embodiments use pollen chirps which have a finite relevant "life." Working backwards from how urgent timely delivery is to the aggregate subscriber, the logical mesh tree topology is aligned, FIG. **1** as is the bus size and frequency of departure, FIG. **12**,**24**. In this logistics supply chain between suppliers and demanders. Aggregate demand is calling the shots at each bus stop. The system is attempting to stay in dynamic alignment at each router along the path. Aggregate Supply/Demand Alignment and the arbitration and auctioning mechanism, employing collaborative scheduling, is taught in FIG. **26** and Ser. No. 61/615,802.

Thus, since the pollen is being containerized and repacked at each bus stop, changing subscriber demands are incorporated in real time. The bus journey may seem more meandering, like searcher ants, than a more directed and predictable path of an IP packet with static and preset destination. But this approach, with stops along the way, at each bus stop, ensures that both routing and subscriber interests are in alignment. Routing issues related to congestion along the IP highway (e.g. accident) is addressed by adaptive mesh network topology, dynamic load balancing et al, taught in Ser. No. 10/434,948.

Changing destinations of entire bus loads (e.g. hospital, now, not the school), is an extension to IP based networking. Next generation IP protocols may allow senders to include both a specific IP address (default setting) and a more "fuzzy" suitable wind direction to aid pollen to:

a. travel more efficiently (fewer hops, low latency) and/or

b. travel more cost effectively (bulk, latency-indifferent delivery) and/or

c. travel more reliably (e.g. TCP/IP like services ensure packet delivery)

Wind carrying pollen change directions based on demand. Publish-subscribe "demographics" is reviewed by "big" data inference and analytics subscribers. They predict trends that may feed back into the alignment driven control system. Collaboration topologies, FIG. **26**, change accordingly. The logical meshed network trees FIG. **1**, **9**, **22** changes connectivity in response. The radio and protocol agnostic mediation layers, FIG. **20** at each network node, enable scalable distributed control.

Chirp Aware Routers

Chirp Routers include chirp-to-chirp routing and also bridging across chirp and IP based devices.

The bridging includes wire-less (e.g. Infrared, Sonar, Wi-Fi, Bluetooth, Audio et. al) and wired connectivity devices (e.g. power line, Ethernet, serial/parallel cables et. al).

Logical Radios:

FIG. **22** shows an embodiment using a bridge provided by an edge router that is both Chirp and IP protocol aware. The bridge **2216** transports chirp devices operating with low cost infrared LED transceivers, transporting data through the nightlight embodiment, via Wi-Fi into a wired network **2212**.

Some devices in the bottom "infrared" layer of FIG. **22** have Wi-Fi radios (e.g. laptops, TV). They use the same physical Wi-Fi radios for logical radio chirp and IP transmissions, in one embodiment. One physical radio supports diverse media access protocols (MAC) concurrently. Logical radios FIG. **7-9** and protocol agnostic control layers, FIG. **20** enable this. Hence, Wi-Fi radios, "upgraded" to support both Wi-Fi and chirp protocols, can now communicate over two different protocols, for different packet sizes, latency, retry and reliability requirements. How the two coexist is a matter of collaboration. For example, the dance partners may agree to not stub each other's toes: this means that VOIP like chirps are bulk shipped and/or Chirps are directed to be sent during silent periods e.g. IFS spacing times. Alternately, chirps, being short, may be included in the Beacon Information Element section. Lastly, more Wi-Fi physical radios may be added, like more lanes on the highway. The mesh bridge device, in one embodiment, has slots for dedicated chirp only road/channel see VOIP radios FIG. **7-8**. Thus multiple physical and logical radio combinations are supported in modular embodiments FIGS. **9**, **17**, **23**, **26**.

When the slots are limited, one Wi-Fi radio may be used for both chirp and IP packet flow. Chirps speak "foreign" tongues, unintelligible to all except their agents. Further their transmissions are being managed at the router level. Thus, next generation Wi-Fi devices, as embodiments of this system include chirp-aware features—chirps sent in the quiet times of IFS spacing, known to Wi-Fi radios and managed by the router. In another embodiment the chirp is digitally encoded in the ASIE section of router beacons. Further, the smarter chirp devices can be directed to time their chirps like smart VOIP phones, FIG. **24**, in yet another embodiment. As described in a later section, the Access Point may also send out an anticipatory "incoming" siren like warning, using CTS packets, to silence Wi-Fi client, if it knows/surmises chirp periodicity/patterns. Many options thus exist for concurrent chirp stream and IP transmissions using one transceiver pair. Logical radios and protocol agnostic control layers support using the same physical radio but different Media Access Control (MAC).

These chirp aware routers also manage efficient bulk packaging and scheduling of buses of VOIP like chirps per methods taught in Ser. No. 11/266,884 FIGS. 24-25 and Ser. No. 11/088,330, FIG. 9. The mediation layer, FIG. **20**, supports both stationary and mobile nodes, FIG. **14-17**. Methods employing Chirp equivalents of SIP registries is taught in Ser. No. 12/352,457, FIG. 27. Thus, chirps may more efficiently be directed to their agent/subscribers without unnecessary brute force means like broadcasting.

Mobile Agents may not be currently resident on the system. Methods supporting persistent and temporal mesh networks are taught in Ser. No. 12/696,947, see FIGS. 28-29. These methods are relevant to both chirp and IP based chirp devices (e.g. VOIP). Mobile and intermittent connectivity devices are linked through mail boxes and persistent mesh networks, see 61/148,809, FIGS. 27-29. Routers can provide a logical and convenient post office and local support center hub. These edge routers are also fellow collaborators in the flow of small data stream fish upstream to bigger information pools.

Beyond bridging and routing, chirp aware mesh nodes provide disembodied machine intelligence for low cost and/or blind devices. The cost of making blind devices agile/polite is avoided. Like a school teacher, the nightlight embodiments provide order in the classroom. Chirp listen-capable devices are told to stop and go so they do not interfere with important transmissions, see FI. **24-25**. With heavy lifting relegated to routers, chirp based end devices may be lower cost and "thin client". Computing and storage needs are entrusted to others in the "social network" community FIGS. **22**, **26**. Thus, the Apple TV box can also, with additional memory or "tune" slots, support chirp based collaborations. Note that through USB serial ports, routers service low cost chirp devices with no IP stack, in one embodiment.

Ant-like Scheduling agents e.g. "Avoid" and "Cluster" also monitor the environment and help minimize blind chirp contentions, in one embodiment. Tree based collaboration frameworks, FIG. **26**, schedule bus route and departure times to "avoid" and "cluster" and manage "Stacking" Ant-like Tunes, operating through the night light embodiments are thus, collaboratively managing the activities of the ants (devices) in the ant colony (collaborative distributed network). Like ant hills, complex organisms can be built on top of collaborative simple ant like "tunes." Exceptions are addressed at multiple levels of the super organism, through distributed responsibilities but central thought process, in this case managed by the secure nightlight embodiments and/or mesh nodes.

Thus, through nightlight embodiments, both chirp and IP collisions are collaboratively managed.

Further, some chirp aware devices, like VOIP phone counterparts, are capable of timing their transmissions without supervision from the nightlight, see FIG. **25**. For others, the nightlight provides stop and go signals. Note also multiple transceivers are supported FIG. **22**, **2214**. Such devices support wire-less **2216**, and/or wired connectivity **2212** (e.g. Ethernet, Power line). Serial USB connections connect external specialized secure transceivers. Thus stop and go directives are received on different frequencies, all part of the logical radio and logical mesh tree.

Multiple Existing Devices are Night Light Possibilities.

Mobile smart phones, tablets or laptops have the transceivers needed to communicate "tunes" to and from light based, sound based and Wi-Fi and Bluetooth based purpose built chirp devices. Additionally, software "apps" provide interpretation and translation functions between chirp, IP and Hybrid Chirp and IP devices, in one embodiment.

Smart phones from Apple, Google, Samsung, H T C, et al support cellular service and/or VOIP and VOIP service (e.g. Skype, Google, Vonage). Laptops and tablet computers support IP and VOIP. Thus, they may also, serve as routers, using wireless and wired IP networking to provide real time communications between chirp devices and support agencies.

Many consumer devices already have multiple transceivers. For example, most electronics makers provide smart phones, computers, music players, Internet TV appliances. The Apple TV and Google TV have Wi-Fi, LED, Ethernet and potentially power line reception capability. They can direct traffic on both Chirp and IP roads.

They can thus manage situations where a chirp device needs an IP tunnel for diagnostics repair, Like traffic policemen, they hold IP traffic, enabling chirp ambulances collision free lanes. Collaborative Scheduling agents monitor the "Stacking". If Ambulances become more frequent, the control system topology changes to accommodate it, if needed. Thus rich real time chirp streams are viable.

Further, many chirp capable devices have Infrared and Wi-Fi. Like custom VOIP phones, they are taught as discussed in the embodiments in Ser. No. 11/266,884, FIG. 24-25 to be more agile. Apps, loaded on night light (AP) and/or VOIP like devices, may coordinate multi-band duplex communications, FIG. **7-8** for remote diagnostics.

Many devices have touch displays to facilitate human interaction. Situation displays, Network management system or "dashboard" views, of home/factory network and machine states etc. thus supported by these devices. Some, like Apple, Google TV set top boxes access large TV displays. There are good night light candidates, as are TVs. Large displays hide "antennas" for different frequency spectrum (e.g. 900 MHz, Wi-Fi radios, Infrared chirps for added security etc).

Further these devices also use their internal and/or external, removable SD card storage for community mail boxes as needed by one embodiment of this invention. They thus provide a base for mobile, intermittent connectivity devices.

Many such devices card slots, FIG. **17**, provide portable secure storage for chirp apps, tunes, mail boxes and other payload data. The cards serve as on/off key, in one embodiment, —if removed, associated devices disconnect and/or shut down. Further, devices may insist on being "tucked in" at night. If their mother "app" does not chirp "Good Night" at a specific pre-set time, frequency or GPS location the device will sleep until awakened by a mother chirp signature.

Nightlights also provide disembodied and distributed intelligence for communities of devices. It is the logical place for community mailboxes and intra-network collaborations. These include the more sophisticated "Chirp Aware" devices that, like polite/agile students, can take care of their own collisions with each other. They look towards the night light to manage the unruly ones.

Distributed collaborations, FIG. **26**, includes collaboration from ant-like tunes operating on other routers. For example, the (battery backed) Kitchen night light notes no chirps were received from kitchen appliances in the last 24 hours. Before filing multiple repair request reports, a collaboration tune request that the kitchen night light confer with the living room night light embodiment, a Samsung TV via the power line network. The Samsung TV confirms that its devices are fine. Bayesian Inference and causal reasoning engines, correctly infer that the kitchen fuse has blown. If so, further causal reasoning indicates that perishables in the GE refrigerator are suspect. The home owner is informed via his smart phone. No repair reports are made. Thus, application specific intelligence at the chirp router level reduces customer support overhead and false alarms. Simple ant like collaborative agents worked together to provide complex reasoning.

Collective Consensus

In one embodiment, Single chirps are less reliable than consensus from multiple sibling devices. Further, hypotheses are supported by corroborating evidence from independent observers e.g. the living room router information assisting in the conclusion that the fuse is blown. The combination, in a trusted network increases local inference capability.

There are multiple applications where intelligence/autonomy is needed closer to the end devices. Beyond the issues of the need to avoid "round-tripping" up to a root node and back, proactive response is engendered by providing local, application specific intelligence and decision making capabilities near the devices. Solving the "problems" at lower levels reduces what percolates to the top. Distributed collaboration methods are discussed in Ser. No. 13/571,294 et al.

Concurrent Conversations

If a chirp device needs a real time IP tunnel (e.g. remote software install) then nightlights provide DHCP based IP addresses with inherent conflict resolution for the bridging, FIG. **27-29**. Further, chirp command channel and IP data channel may be separate, FIG. **7-8** but, through an interpreter layer in the mesh node, provide a human or IP packet based interface to the chirp device. Many Chirp to IP tunnels may be created with temporary DHCP address as the stub.

Chirps are being continuously broadcast, in one embodiment. The Network Management System and/or Diagnostics data is thus simulcast where machine participants such as Collaborative agents, may offer advice on scheduling etc. Thus chirp routers support their less advanced and less versatile chirp "country cousins." The relationship is symbiotic since disembodied intelligence of low cost chirp devices reduces the total cost of ownership of the entire community of devices, both chirp and IP based. Centralizing the intelligence at the routers is good, secure strategy for all.

FIG. 27, from Ser. No. 12/352,457, depicts a real time communications network with VOIP phones. In embodiments where routers are chirp aware the methods taught therein are applicable. Routers slots that are modular support chirp and IP protocol aware and support transceiver slots, FIG. **22**. Hybrid routers may then, with appropriate interpreters provide cross communication between all types of devices in the network.

Night lights thus provide connectivity and interpretation layers (described as propagator and integrator layers in application Ser. No. 13/571,294). Further, chirp devices with store and forward capacity may act as relays for devices not directly reachable by routers (root nodes).

FIGS. **32-33** show a variant on FIG. **13**. The devices are serving as repeaters and range extenders but use the same physical layer to service both uplink and downlink in the two-logical radio concept. Thus, in applying the two-logical radio concept, the routers listen and then repeat after a delay, like birds repeating each others songs. FIG. **32-33** thus serve as single radio repeaters in FIG. **14-15**. Note that the Mesh Control Layer, FIG. **20**, in active in all units shown in FIG. **32-33**. Thus, Chirp aware devices serving as propagators/range extenders are also be mesh aware, in some embodiments.

FIG. **13** Label **030** depicts a two radio bridge, with one radio as an uplink and another as a single radio mesh AP, also referred to as single radio ad hoc mesh networking Radio **040** routing is a graph, not a tree. Tree structures have no loops and complications of looping are thus eliminated. Recall that the routing table is O(n) in size. Routing updates are much faster, even with many nodes in the network. In contrast Radio **040** is a hub like structure which can involve looping. Routing table are O(n^2)—they grow exponentially. Updates are slower since graphs have loops and minimal spanning trees have to be recomputed periodically. Thus, this form of range extension is to be avoided in general. However, it is a viable and much used form of mesh networking.

FIG. **13** units **010**, **020** provide bridging. If the two radios **030**, **040** are also "chirp aware", then standard IP Wi-Fi radios may be used for both chirp and IP transport concurrently (e.g. chirps in the beacon or silent periods) or switch range extenders periodically, for both Chirp and IP devices. This is an example of dual mode operation: night light embodiment and device range extenders.

Thus existing products and devices, with chirp like transceivers, are capable of supporting multiple and dynamic levels of secure human-machine interaction and secure machine-machine interaction. High Security is inherent in this pollen based world. Further, through interpreters cards in the back slots of FIG. **22**, terse machine-machine communication now more easily incorporate human-in-the-loop components, in a real time manner, if needed. The birdsong is now comprehensible.

Human interface is simpler with local in-circuit interpreters, close to devices and the context. System level diagnostics of large and often distributed control systems is challenging when parts of the system cannot be studied in isolation or when local events are not easily repeatable. Duplex communication and multi-lingual interactions make real time interpretation and analysis of machine chirps easier to follow. In embodiments of the system it is easier and faster to catch local "bugs" and "fix" them.

Software "Apps" or agents also provide users with status reports on the health of their chirp devices on tablets or phone routers. These apps include collaboration agents such as "avoid" and "cluster" that collaboratively schedule repair visits, per methods are taught in 61/615,802, 61/555,400.

Multiple tunes thus support the varied and dynamic needs of multiple subscribers, without unduly burdening the network, since now only "rich" chirp exception and/or clones/relays are sent "up."

Agent/Tune capabilities span the gamut from simple relay propagation through real time interpretation all the way to completing transactions in a global supply chain scenario, in different embodiments of the system. Here devices self-order replacement parts or automatically upgrade automatically, because their human parents have opened an "account" for them. These devices are now capable of self-healing.

For example, a newly purchased chirp device is registered and software "apps" are downloaded to the owner's smart phone. The "apps" contain tunes for remote diagnostics, over the air firmware upgrades, et. al. These apps are authorized, by the user, to charge "ringtone" services to her phone service account or bank/credit card account, with usual overrides, counter signatures beyond a certain limit etc. Thus the human in the loop can step in and challenge a device replacement part request. For instance, She would rather use those funds to buy another type of device. Planned Obsolesces replacement patterns are thus supported. Notifications of death are also supported. Low cost products can chirp before they "die", reminding their owner to replace them, similar to smoke alarms chirping before their battery dies.

Devices chirp their state to nightlight "Mothers". Resident Tunes can serve as aggregation agents for Enterprise customer support and sales. When a product is purchased and registered, the chirp router may "open" a subscriber account for these chirp signatures based pollen. Thus, the enterprise has a secure IP based address through the router that patches through a chirp equivalent of a VOIP phone line for it. This simplifies remote diagnostics and repair. It also enables a centralized Enterprise Server to monitor devices and provide after sales services, thus enabling Womb to Tomb remote customer support. Intermittently connecting devices are serviced through community mail boxes at the "home" post office branch (e.g. home base router).

Womb to Tomb support may require communication over the IP bridge with varying degrees of urgency, based on the type of customer support purchased, in one embodiment. This is akin to QoS but is subscriber driven and both authorized and secure.

Multiple levels of QoS are supported.

Consider restaurant food delivery services as a metaphor. Their menu, like pollen, is freely broadcast over multiple subscribers "channels" e.g. Radio, TV, Web. Consumers visit (and "open accounts") at multiple restaurants on chirp signatures and their individual desires at this point in time. The pollen broadcast runs the gamut from lean and simple data (e.g. the temperature is "green") to rich and complex actionable knowledge (e.g. the cat is in the kitchen).

Some restaurants have their own delivery "buses" that they have leased or scheduled with the bus travel service provider (e.g. Chirp routers). The cost of the bus "ticket" is added to the cost of the meal delivery paid for by subscribers. Other restaurants may suggest the customer contact a trusted delivery service (e.g. Waiter-on-Wheels). They deliver the "package" per customer QoS, latency and proof of service requirements. For example, couriers assure timely delivery (e.g. Fresh-baked Pizza).

The nightlight embodiments, with chirp-to-IP and "app" based tunes support multiple QoS levels e.g. courier service, scheduled buses and bus convoys, shipping/freight containers. Recall nightlights provide disembodied intelligence for chirp devices and their IP counterparts. That intelligence includes collaborating with others to pack in as much into a container, whether the container is a special courier, bus or train/ship container. Ant-like Collaborative Scheduling agents, FIG. **26** using auctioning mechanisms, dynamically align pollen supply with subscriber demand,

Tunes/Agents, acting as advocates and agents for their owners, can also initiate a "call home" VOIP call between a "lost" or ailing device and its support center, through trusted router intermediaries.

For example, the user may be directed to hold the smart phone close to a "lost" device. The tune then automatically calls an anonymous automated answering service. Ringtones are exchanged over an inexpensive audio transceiver on end devices. Other trusted friend tunes are part of the conference call. One is interpreting the conversation in English, using voice synthesis on the smart phone. If a decision point is reached requiring human interaction, the human is asked. Thus human in the loop and human override is optionally available through voice or touchpad etc.

Device pairing, managed by secure chirps, ensures that devices in transit cannot be tampered with. Thus, mobile chirp devices do not talk to strangers who don't sound like mother or mothers agents. Further, "lost" devices are recoverable as long they can recall their mothers chirp. Further, Publisher—subscriber relationships are multiple and varied. Chirp Devices may visit an "aunt", along the journey home and in response to her chirps, relay a message from Mother. Chirp based secure courier services are thus viable, a more modern, versatile and secure version of carrier pigeons.

Smart phone and other mobile computers thus serve dual roles as routers and couriers. They are servicing an extensible iterant chirp and IP device community network with chirp-to-IP bridging available for immediate, emergency use. And QoS is supported through IP tunneling and through trusted chirp to chirp couriers via chirp routers or other mobile chirp aware devices.

Device pairing renders stolen devices as unusable. Its chirps are not understood by unauthorized nightlight. Further, if mothers/aunts do not "tuck in" their charges at night, they may be programmed to "sleep". Nightlights may also report un-paired chirping devices to appropriate authorities.

The features stated above are embodied in the nightlight, see FIG. **22** and described in Ser. No. 13/571,294. Note however that while mobile generic purpose devices (Smart phones, tablets and laptops) can provide intermittent connectivity, static and purpose built night lights embodiments ensure "always on" connectivity for mission critical applications. Home security and surveillance is an example.

Chirp Routing Protocols

Chirps, like pollen, are often simple and lightweight, for reasons explained above. In one embodiment, they use low cost and low overhead IR based transceivers, see FIG. **22**. The chirp may be relayed through propagation agents via multiple hops. Eventually it reaches the port of entry in the chirp aware router—using transceiver slots operating in the same medium as the chirp or its relay agents, see slots in FIGS. **7**, **8**, **12**, **16**, **17**, **23**.

FIGS. **7**, **8** show transport up to the root node using dedicated backhauls for VOIP chirp-like packets and data. In another embodiment, the chirp data may be converted by agents to travel in another packet format/protocol, see **22** in FIG. **9**. Then, after the first hop, IR chirp data would be converted to Wi-Fi based chirp data formats, see FIG. **22**. Or, in another embodiment, chirp devices may share the same Wi-Fi spectrum, as described below.

In an embodiment where the chirp devices and Wi-Fi devices share the same Wi-Fi spectrum, the chirps are "simple", they operate clumsily and the agile IP based Wi-Fi devices must proactively avoid contention. This agility may be provided by Access Points, in one embodiment. One slot of nightlight embodiments, FIG. **17**, **020**, indicates a four slot configuration with an Access Point. Another slot, could house an IR transceiver. Once received, the chirp must eventually be converted into some Internet Protocol compliant packet, to travel upstream/downstream in search of interested flower/agent/tunes.

Internet Protocol uses a "From" and "To" addressing scheme. This information is generally public, for IP based routing (wired and wireless) to work. FIG. **34** shows a representative Wi-Fi Request-To-Send (RTS) packet format. RTS announces the intent to transmit and specifies the "to" (receiver) and "from" (transmitter) addresses, see **3424**, **3426**. The Frame Control data, **3420** at the beginning of the packet contains pertinent information such as power management features intended for access points to know when the device will awake and thus buffer its packets. This is also relevant to chirp devices and their interactions with night lights. Note that Frame Control Data may be used an existing protocol to communicate a distinctly different form of addressing, related to pollen and flower/agents Chirp communications are not limited on strict requirement of providing a fixed "to" addresses, as is described herein.

The Duration Field, **3422** indicates time of transmission requested, which Wi-Fi stations use to set their Network Allocation Vector (NAV) and avoid contention. In one embodiment, to prevent network congestion and to prevent transmittal of outdated chirps, even blind chirp devices—chirping randomly—can provide this information. Co-located devices can then be agile and avoid stepping on their clumsy fellow dancers in the same RF space. Note also that, an RTS is typically followed by a CTS or Clear-To-Send from the Access point managing 802.11 stations associated with it. Therefore, if chirp devices specify their periodicity, or the transmission pattern they are following and their current pattern sequence index, then the APs can preempt contention at the expected chirp transmission time by sending a CTS ahead—like a police car siren, it warns both IP aware and Chirp aware (hear and send) devices of other, unexpected traffic.

Vendor specific chirp information exchange may be supported in the 802.11 standard through Action Frames, see FIG. **35**. In one embodiment, these contain 1 byte for Category and 1 Byte for Action type, see FIG. **35**. Hence there are 255 non trivial categories of information, with 255 non trivial types of data being sent, each of which has 255 non trivial Dialog Tokens—expressing data formats. With appropriate filters one Action Frame could provide data for multiple agents/tunes in a compact transmission. Note that, akin to the RTS packet, it contains Duration information. It may also contain the chirp equivalents of Destination Address DA, **3520**, Sender Address SA, **3522** and BSSID, **3524**. DA and SA relate to Receiver and Transmitter Addresses in the RTS, FIG. **34**. BSSID may be loosely thought of as a chirp in search of a specific "flower"/tune/agent.

Chirps may thus be encapsulated in exemplary Wi-Fi Action Frames, for onward transmission to other chirp aware routers. The packets will travel through prior art—and not chirp aware—routers without incidence.

The exemplary Action Frames may be sent in unicast, multicast or broadcast modes—this is dependent on the Destination Address DA. Should IP multicasting be used, then, with IGMP protocols, chirps will be efficiently transported to the interested members of the multicast group. Efficient transport mechanisms have been described in Ser. Nos. 11/266,884, 11/818,899, 12/696,947 involving bulking, scheduled delivery, servicing isolated clusters, maintaining SIP like registries, etc. are all are applicable to chirp transports over IP.

If the subscribers/agents are not known or choose to be hidden, then IP group based multicasting (e.g. IGMP) is not useful. A more brute force approach is needed. This is broadcast mode, where the Chirp packets may travel both upstream and downstream of the mesh tree.

Broadcasting is how pollen reaches the "interested" flowers in nature. As long as the broadcast durations are managed, flooding and network congestion is contained. For example, mesh network nodes described in this application send out "heart beats" announcing their presence and current state regularly in broadcast mode. Heart beat counters are used to avoid resending of "old" packets. Second, the mesh topology is tree-like, hence broadcast directions are limited to up, down or local (within siblings). Third, the packets themselves may be encoded with time to live function or the maximum number of mesh tree hops. These methods have been successfully applied to contain flooding in a tree based mesh network. They are be reapplied to "chirp" heart beats, maintained by the routers, if needed, in one embodiment.

In the event chirps need to leave the mesh network and enter non chirp-aware networks, flooding control is used, with time to live functionality employed in one embodiment. Additionally, the bulk container bus delivery service is specifically designed to efficiently send packets over non chirp aware networks by forwarding them to a chirp aware router at the other end, using standard IP based routing, with applicable encryption. Thus chirps will get to where they need to go, to their agents/tunes/flowers, using either multicasting or broadcast modes.

While (pruned) broadcasting techniques will get the chirp to the interested flowers, it is over-provisioned, like allergy season. One approach to providing more routing information is to specify both a navigation agent and a data handling agent in the same chirp. The navigation agent accesses a portion of the chirp packet. Access is limited to the routing/navigation of the chirp, not its payload. Navigation directives may be either physical or logical. Physical navigation is turn-by-turn directions e.g. three nodes up, third child sibling node down, stop. This is useful when private and static networks are deployed. Logical navigation is more flexible e.g. move up within three hops in search of Agent-For-GE-Toaster, stop. If the agent is not found, then put the chirp in a lost and found area in a community mail box for forwarding to mobile agents. For example a kitchen night light provides updates, via smart phone agents, to a user, when he returns home, in one embodiment.

Inherent Security in Chirp Data Transport

Chirp data transport involves traversing the IP network, in some embodiments, and are thus susceptible to snooping/hacking But this is not your typical IP data packet since the data is based or Chirp/Pollen-ID and/or Flower/Agents-ID etc. These are not typical IP or MAC-ID type Sender/Receiver Address Frames, FIGS. **34** through **36**. However the same format is available for chirp devices use to specify, if desired:

a) The IP Destination Address (if applicable), and can include other addressing information depending on the type of transmission involved—e.g. unicast, multi-cast or broadcast, **3520**,

b) Chirp-ID (in the Sender Address frame, SA, **3522**),

c) Agent-ID being sought (in the BSSID Element, **3524**)

d) Any other use of the IP frame formats, recognizable by an agent.

All options (and their variants) exist within the exemplary Action Frame format, suitable for transmission over standard WIFI networks.

FIG. **36** shows a measurement request action frame, **3630**. These management or action frames look like the innocuous request from stations requesting information from a specific AP (with BSSID). Only chirp aware routers are aware that these are actually chirp packets and that the data in the DA, SA and BSSID is to be interpreted differently.

Further, chirp routers know just enough to decode the DA, SA and BSSID data sections, FIGS. **35**-**36**, to provide necessary routing. They cannot decipher the vendor specific information elements—only specific agents/tunes/flowers hold those keys. In other words, routers can engineer the "winds" and "buses" to move the pollen, define the schedules for the buses, based on QoS settings in IP-like packet, but do nothing else.

How does the Chirp aware router recognize a chirp packet? In one embodiment, the router knows which interface transports Chirp packets. It has a complete list of 802.11 stations associated with this AP. As part of the tree topology, it also has a list of all stations downstream—via the downlinks, FIGS. **4** through **8**. It does not have access to the routing tables up stream, as part of the tree based routing scheme. However, it may use distributed SIP registries that contain both chirp device and agent ID locations. Ser. No. 12/352,457 describes using dynamic SIP registries to provide VOIP phone connectivity within dynamic isolated clusters. The same principles may be used to define where agents/flowers are, or the reverse look up—where chirp devices are, of interest to a particular agent.

Through either SIP like registries, or the routing table of its IP based stations, each chirp aware router is aware that these chirp packets are not emanating from one of their 802.11 clients/stations. No one else in the system—both outside the mesh network and within it has this insight—access to the distributed routing tables of 802.11 clients and/or distributed SIP like registries is needed.

Even without the aid of SIP-like registries, chirp routers are still cognizant of the special nature of the data packets being transmitted. Chirp routers are keeping track of which interface was used to inject the IP packet into the mesh network. In the mesh nodes shown in FIGS. **17**, **23**, there are multiple interfaces generally provided—the uplink and down links of Backhauls (BH and FD) and the client Access Points (AP). If the chirp packet came in through on of the APs, then it is marked as a chirp, since the Chirp ID provided does not match an associated IP based client's IP address or MAC-ID in its routing table.

Note that the AP does not need to keep a list of chirp devices it services—it surmises its identity based on the exclusion principle, namely, this device ID, if an IP based device, is not in its routing tables. This implies, that chirp device locations and identity do not have to be stored, if anonymity is desired. The chirp will still be forwarded. Up and/or down through the up links and down links the mesh tree, marked as a chirp in search of the agent/tune/subscriber. The identity of both chirp and its interested agent/tune/flower may be hidden and yet the pollen will reach the flower. This extends existing prior art IP based routing security.

Thus, in one embodiment, even with both Chirp-ID and Agent-ID hidden agents who receive broadcast chirp packets, are the only ones privy to what is being said, by whom, and intended for whom, using this specific data format etc. And only these agents can route such packets to other agents in its private SIP like registries, have them or other agents inform routers to stop broadcasts other otherwise affect the routing—at each step along the bus delivery route.

Further, an agent can convert the data flow to be IP based, with regular IP addresses. In one embodiment, private networks coexist and span both chirp and conventional transmissions using "pattern" hopping techniques known only to them. Part of the data could travel as IP Data frames while others via chirp protocols, analogous to musical chords or dual signatures needed on a check. Only agents know the (chorded) "tune" This further obfuscates the chirp data flow.

In embodiments, chirp data comprises IP based transport packets whose format is regular and legitimate IP-based packet. It supports all the Frame Control feature sets, FIG. **34**, **3420**, including multiple frames, power management and Distribution System (DS) flags. Thus, competent agents, running on chirp aware routers, may also convert IP data (including VOIP packets) into chirp packets, send them anonymously to another agent and then reconvert them back to IP traffic. This obfuscates IP based data flow. Thus both Chirp and IP payloads may be used interchangeably to obfuscate data flow within both chirp and legacy networks.

Additional Approaches to Managing Chirp Contention

The typical chirp data transport data packet must necessarily be small/light, to avoid contentions with IP based "heavy" traffic. In some embodiments, the RTS packet FIG. **34** and the Action Frame, **35** are short packet types used and attractive candidates for individual chirp packet transmissions to the Access Point (AP) that first receives them.

In some embodiments, even blind chirp devices can include the DURATION information **3422**, so IP based devices and other Chirp devices with listening can avoid contention during the time that chirp packets are known to be active. Further, if the device follows a known transmission periodicity/pattern, then the AP can preemptively clear the communication medium, by sending out a Request to Send/Clear to Send frame (CTS) that effectively silences both chirp aware listen capable devices and 802.11 client stations.

If blind chirp devices use larger packets then blind chirp packets should register their chirp pattern with the nightlight/

router. Each chirp must then also contain the chirp pattern sequence number in each transmission. Then the router can generate a CTS in anticipation, since it knows both pattern and sequence number. The CTS will therefore preannounce the time reservation made for the blind chirp device. Thus provisions exist for larger blind chirp packets, contingent on coordination with the device's local router.

If listen-capable chirp devices use larger packets, then they also should register their transmission patterns when first pairing with the nightlight/router. The router can then generate the anticipatory CTS. Additionally, routers, based on the type of traffic pattern it is seeing, can direct these chirp devices to reposition their time of transmission to avail of a bulk CTS with a duration value set to cover multiple sequential device chirps.

This embodiment is, in effect, the reverse of FIG. **24**. In FIG. **24**, the bulk transmission from the AP to multiple listen capable devices is managed by sending out a bulk transmission to all and informing then of this common time. During this time the VOIP like chirp like devices are expected to silently listen. Conversely, the chirp router can also specify when the chirps from devices should occur so they are contiguous and thus covered in bulk periodically. Now, the devices are also being taught when to talk—sequentially. Thus the forward and reverse methods proactively deal with chirp contention using multiple means including those described in Ser. No. 11/266,884, 61/615,802, 61/555,400 et al. Chirp routers are acting as the collaborative scheduling agents and engendering collaboration between chirp and non-chirp clients in sharing the same media with minimum contention. Embodiments include chirp aware smart phones, Wi-Fi Access Points and other devices, where a logical radio "slot" provides the requisite software/firmware functionality.

Approaches to "Small" Data Feeds Big Data

FIG. **36** depicts the Measurement Request Action Frame, as a representative versatile Chirp packet. In one embodiment, the measurement data field **3630** is of variable length for flexibility. In some embodiments, three one byte sections are provided **3624**, marked as Category, Action and Dialog Token respectively. Thus each section supports 255 non trivial variations: Each chirp packet has 255 ways of expressing:

1) Category: What type of chirp data is being transmitted
2) Action: Which State information is being sent.
3) Dialog Token: What type of data format, parameter list is being used.

Despite the terse length of chirp transmissions, there are sufficient variations to define precisely the type and nature of data being transmitted. Each of the 255 categories for each chirp device have 255*255 different ways to express machine state as classified under Action, and Dialog Token. Thus M2M communications may be terse but specific in terms of the data provided to "small" data integrators. Further, chirp devices may follow patterns in bulking transmissions. Thus 5 different measurements, for 5 distinct variables, may be transmitted in one payload or five smaller ones. Further, since the Dialog Token defines the "key" to parsing the data, it could also represent 255 unique parameter list orderings. Thus the chirp data can be jumbled between patterns making it difficult, like secret handshakes, for snoopers to decipher content, especially since the pattern used is also changing, like temporal keys, but with significantly lower overhead.

Agents can operate on other agents. Hence one agent can forward the chirp—after massaging the data, if needed—to another agent and so on. Agents can also spawn other agents, so distributed computing and routing is engendered. Thus one agent could clone itself to generate two containers so two buses concurrently carry containers to different destinations. Note that pruned broadcasting mechanisms, see Ser. No. 11/266,884, ensure that chirp like VOIP packets are transported selectively. Further SIP-like local registries, see Ser. No. 12/352,457, may also be used as subscribers/agents interested in specific chirps.

Complex business process logic flow is thus possible through a single agent spawning multiple collaborative agents, all of which emanated from receiving a chirp. By distributing the intelligence in an agent based network needed to service the chirp, chirp data, while terse, is still very powerful. Terse data need not be restricted to simpleminded functions. Note that in nature, simple ants create complex colonies. By the same token, chirps can be terse but not "simple" in the aggregate. Multiple chirp flows through a distributed mesh network, will interact with a hierarchy of agents. Some provide propagation. Others serve as Integrator agents operating on multiple chirps, to generate meaningful "small" data and situation awareness. An agent based network is a significantly more powerful means of providing dynamic routing/propagation agents and higher level functions, like integrator agents, all within the same distributed meshed network.

Propagation/Routing agents also signal to each other, in some embodiments,—so if one agent receives the packet first, it can tell the routing agents in the chirp aware routers to stop broadcasting, thus containing broadcast traffic proactively. Agents on mobile devices can move from one router to another, to further obfuscate their location/identity.

The routing agents MESH CONTROL and SCAN depicted in FIG. **12** may expose their API to selected agents who may then change the routing tables and redirect traffic at both local and remote mesh nodes. A dynamically reconfigurable routing architecture emerges where agents drive the routing/scheduling of delivery at each logistics hub, working in consonance with the Mesh Control Layer, FIG. **20** and its features, FIGS. **21** through **28**. The Collaborative Scheduler 61/555,400 is another agent available in the support framework. Together, they ensure pollen reaches intended flowers as/when requested.

In some embodiments, agents are physically located on a mesh node or node clients (e.g. removable USB security stick on a laptop client device). Agents may also physically reside on the mesh node, FIG. **16** on a card slot or Ethernet port in the node. They can also be baked into the firmware at flash time, see FIGS. **18**, **19**. And they can also be "registered guests", through software that manages the ACL and other lists that Access Points use, for example. Operating inside the mesh node, they can redirect massaged data to a secure server, through secure socket connections. Many agents may be mobile, with intermittent connectivity, see FIG. **27**, **28**. Since the connection is both intermittent and short, the data flow is not useful from a snooper's perspective. Further community mail boxes agents, resident or remote, may be used to buffer recent broadcasts for the agent and obfuscate flow. Applying an agent/tune/flower female receiver oriented approach provides a transport mechanism which is inherently more secure and more versatile, without requiring any changes to legacy systems.

In some embodiments, the base level chirp/agent discovery process is multicasting/broadcasting. The pollen/flower search is driven by chirp ID and/or Agent ID, through Chirp Aware Routers. Extensions include inter agent communications within the Agent "Social" Network/Collaborative Eco-system. Thus, very private internal broadcasting clusters may form, within the outer layer of the base layers.

Two different network "trees" emerge. The physical network tree has chirp devices at the edge, edge router/relays to core router "roots". Similarly, the agent social network is, at their "root" level, big data agents. Below them is myriad agents massaging/filtering/integrating the small data chirps requested by them. Further, the big data "root" agents have access, at the root level, with other fellow roots. Each root agent has access to all of its agent in its sub trees. These include agents that change routing rules, and schedule the "buses," to remain in dynamic alignment with changing publisher and subscriber "blobs" of activity, see 61/555,400. Recall that chirp aware routers provide both chirp and IP based connectivity. Command and Control directives may thus be securely and speedily transported between agents. A hierarchical scheduling system emerges, where higher layers set the adaptive model parameters for lower layers.

Chirp capable nightlight embodiments exist in the form of smart phones, with Chirp Friendly sensors/transceivers e.g. IR, Ringtone, Light or Tactile Patterns/Tunes. Consider in one embodiment a secret rendezvous between "app" or agent "red" with agent "blue". In FIG. **39**, **3940**, the soldier has made contact and their smart phones exchange indecipherable chirps.

In this example, the soldier returns to his unit, **3940** and smart phones exchange information again, as part of buddy system-if the soldier did not return the data is not compromised. Distributed agents (local and remote) confer over the mesh network to decrypt the information and disseminate relevant snippets to individual smart phones. One information snippet may be the time and place for the next rendezvous. Only the "chosen" phone owner receives this message at a "schedule" defined by stacking agents, see 61/555,400 et al. Thus agent based "social" networks may form, merge, disperse with agents dynamically managing the "schedule" for dispersal of secret information.

In chirp agent worlds, pollen will find the right flowers, either through brute force (e.g. multicast, broadcast) or more subtle means, with chirp nods, winks and secret handshakes inside a distributed but secret agent referral and forwarding system. An agent meshed network forms on top of the mesh routing delivery platform of buses, winds, mail boxes and other featuresdescribed herein. Hierarchies within these smaller close knit self-sufficient communities include, in some embodiments, integrator agents, who assimilate chirps, regurgitate to produce "small" feeds and ship those, also using the same delivery mechanisms to upstream big data agents. The integrator agents may search and find other "blue" agents, FIG. **39** and collaboratively put disparate and diverse snippets together to feed "big" data.

Logical Radios and Tree Topologies

FIG. **37** maps the wireless transceiver having multiple logical radio "slots" of FIG. **16**, **17** to the Wired transceivers on the slots/ports of your typical switch/bridge/routers of FIG. **5**. The Uplink, **3702**, Downlink **3706** and Service Access **3704**, provide equivalent services in a switch stack hierarchy, see FIG. **5**. Note that a single access point (AP) radio services multiple wireless clients, hence it represents multiple "ports". Further, while multiple Service Access and Down link ports/slots are typical, there is always only one UP link, since a tree based (non cyclical) topology, is being maintained at all times by Mesh Control FIGS. **12**,**20**

FIG. **38** explores this equivalence further. **3810** is a view into what the "Routing Modifications" **3810** of FIG. **20** entails. Based on a strict tree based formalism, radio/wired slots/ports are equivalently either a uplink (U) or one of multiple downlinks (D), service access or APs (A) or scanners (S), **3830**, managed by the Scan Control, FIG. **12**. Those are the only four types of logical radio modes allowed. Each Transceiver "slot" must map to a physical transceiver device that performs the operation. Thus FIG. **12** shown a six slot/port switch, with four 802.11 radio and two wired ports. Slots could be U, D, S, A in non overlapping transceiver domains (e.g. Wi-Fi to 3G, Infra Red to Power-line). This embodiment allows for bridging across disparate mediums, see FIG. **9**, **22**, **23**.

FIG. **38**, **3840** depicts an embodiment of this principle for a four physical radio and six slot box, FIGS. **12** and **16**. Here, each U, D, S, A logical radio functions maps to a physical radio. In **3840** and its equivalent switch embodiment **3850** each slot is performing a dedicated function. This is desirable from a performance perspective but not a requirement: the mesh control layer is logical radio function based. For example, in FIG. **17**, **010**, AP or Service Access logical radio "A" is being supported by the same radio supporting BH downlink (D) services.

Similarly, in one embodiment using a purely logical radio functionality, a single physical transceiver/radio may share U, D and S responsibilities with dynamically allocated duty cycle for each, based on application and present circumstances, see FIGS. **14**,**15**. The physical radio may be directed to switch back and forth between distinct uplink or downlink channels, for example, thus emulating a two radio backhaul. Or they may share the same channel, collaboratively reducing "stacking" of packet queues, to stay in alignment with requirements, see FIG. **1** and 61/555,400.

FIG. **39**, **3920**, is an embodiment of a single Wi-Fi radio that is successfully employed to provide mission critical voice and video communication in military applications, as shown in FIGS. **14**, **15**. Since the embodiment uses a single radio, three U, D and S logical radios are shown supported by the same physical radio, **3920**. **3930** shows a single radio uplink connecting to a four radio downlink. In **3940** they form an isolated cluster, where each single radio mesh node performs U, D and S, logical radio functions/agents, see also FIGS. **14**, **15**. Further, the single radio connectivity linkage is a thinner "tree branch", (blue) **3950**, connecting back to the backhaul or trunk of the tree (red) **3960**. Thus, mesh node **3952**, join the "red" and "blue" lines of two otherwise isolated trees. It is providing the common router function described in Ser. No. 10/434,948, Appendix A.

The SCAN Agent, FIG. **12**, makes that bridge happen, in one embodiment. The red and blue "channels" are intentionally distinct, on non-interfering channels, possibly on different frequency bands e.g. IR and Wi-Fi, FIGS. **9** and **22**. Hence, mobile/temporal networks scan, using logical radio S, so they may join others on the "blue" line **3950**. They may also apply a portion of their scan duty cycle in search for a "red" node, as described in Ser. No. 12/696,947 and FIGS. **28**, **29**.

Further, if all the radios are operating the same "blue" channel, as shown, **3940**, **3950**, then throughout degrades with each hop, See FIG. **2**, **020**. However, the routing overhead remains O(n) in tree based topologies and hence low jitter and latency is maintained, see FIGS. **24**,**25**,**27**. For example, in one embodiment, clear VOIP has been demonstrated 44+ hops down in mining tunnels. Note that scalable tree based O(n) routing overhead applies to both multi-radio **3910** and single radio **3920** backhauls. As explained in a previous section, one distinctive benefit of the logical radio approach is Faster Routing Updates, because tree like structures are O(n)

In contrast, routing updates within prior art mesh architectures require O(n^2) resources. Ser. No. 10/434,948 Appendix A, reproduced herein, describes these mesh routing techniques and their limitations. Topologies are peer to peer, single physical radio backhauls. A minimal spanning tree must be maintained by each node. In a family of n siblings, each sibling must re-evaluate its relationships with all the other (n−1) siblings. For all n siblings, the routing update is O (n^2). In contrast, with tree based routing overhead is still O(n), even in single radio chains, see FIG. **39**, **3940**, **3950**.

This disparity between resource overhead in maintaining the mesh becomes apparent as the network grows—an inordinately large portion of system resources are devoted to managing mesh infrastructure in prior art embodiments. As such, fewer resources are available for its intended purpose—providing proactive connectivity in static and temporal mesh networks and their forming, joining, dispersal etc. Hence $O(n^2)$ networks simply cannot scale beyond a tipping point. The tipping point may vary as newer radio technologies evolve and provide better throughput capacity. But at some point the performance will be too sluggish to be relevant, especially in dynamic, mobile and temporal networks.

As smart phones proliferate, in some embodiments, the smart phones form their own mesh networks with both static (kitchen night light) and mobile (other smart phones) chirp aware devices. In some embodiments, a network is capable of maintaining communications with many such devices. A prior art single radio peer to peer network with $O(n^2)$ will be consumed with overhead from routing and other functions within a few hops and/or a few members, limiting its "Social Network" value. In contrast, O(n) systems can exploit advancements in radio technology further to stay "current". The slot based system FIG. **16**,**17**,**23**,**25**,**26** and their embodiments, FIGS. **38**, **39**, are future proofed—the radio cards, FIG. **16**, **010**, are removable and upgradable.

Smart phones provide wireless connectivity through Wi-Fi, Bluetooth and Cellular. Chirps may travel via both IP and Cellular networks. SMS messaging are used to communicate between smart phone agents and associated chirp devices, in one embodiment. Thus, two phones may be used to remotely operate, monitor, control, or diagnose devices securely and cost effectively via SMS also. In the example of remote video surveillance, described earlier, a terse SMS e.g. "Cat in Kitchen" and a snapshot, can cover the essence of an exception handling update.

The Tree based routing favors single radio systems for smaller networks because of its capacity degradation of ½ with each hop. The degradation is half because the duty cycle is shared between uplink (U) and downlink (D) logical radio functions FIG. **39**, **40**. With SCAN S duty cycle added, the throughput degradation could be ⅓ per hop. For a three hop single radio U+D network, it was $\frac{1}{2}^3$. For its U+D+S counterpart it is $\frac{1}{3}^3$ or three times worse. Thus, there are limits to the capacity scalability of all single radio mesh networks using single radio logical embodiments, over multiple hops.

The benefits of O(n) tree based topologies and routing are that even within long chains and degraded capacity, latency and jitter are still deterministic, even in dynamic, temporal or mobile environments FIGS. **24** through **29**. VOIP like lightweight Chirp packets are efficiently routed even in single radio versions of long chains (i.e. the "strings of pearls"), FIG. **43**. The distributed mesh control layer, **12**, **20** self corrects U, D, S resource allocations dynamically through heart beat updates and routing modifications, in one embodiment. FIG. **20** including using toll costs and hop costs, as described in Appendix A, manage the overall health of the network, FIG. **1**.

Thus, when recent history of scans and/or GPS readings by/from smart phone embodiment indicate that the soldier FIG. **39** has slowed down, the mesh control layer proactively reduces the duty cycle of the SCAN S function to mostly quick scans. Interspersed within those quick scans would still

be one or two periodic "detail" scans, in one embodiment. Thus the system would stay in dynamic alignment to changes in motion/situation, per methods also described in Ser. No. 11/818,889, 61/555,400 and 61/615,802, while using the radio for functions other than Scanning (uplink, downlink, client service/access).

FIG. **40** is a simulation of prior art mesh routing algorithm and its comparison to tree based routing for single radio mesh networks. The thicker blue lines in FIG. **40**, **4040** depict the minimal spanning tree. Note the dashed lines have to be additionally recomputed for each node in prior art mesh routing. Over multiple hops single radio backhauls suffer from both throughput degradation and faster routing updates. The former degrades by ½ with each hop, the latter with $O(n^2)$ where n is the number of nodes in the peer-peer network. Hence routing table updates will increase to $O((n+m)^2)$ with m additional new members. In sharp contrast tree based routing, with the logical radio abstractions in place, will still be linear: O(n+m))

Tree based mesh routing segments the collision domains, FIGS. **4** through **8**. Each BSS in FIG. **4**, **6** is operating on a non-interfering frequency/channel. Further bridging across transmission domains, FIG. **9**, **22**, is analogous to adding more frequency/channels for the BSS to operate in. In one embodiment, dynamic channel management manages channel changes, see FIG. **21**, all with the intent of reducing channel contention. Reduced contention enables CSMA/CA and CSMA/CD back off algorithms to be more efficient. Jitter and Latency become deterministic, as taught in Ser. No. 11/266,884, FIGS. 24,25.

"Natural", healthy branch growth thus encourages "radios" operating in different "channels", forming non-interfering logical sub trees. Having more "channels" would favor smaller sub trees and more of them. Many would operate autonomously with the occasional need to chirp back status and receive email/firmware updates. Thus multi-transceiver chirp capable product may serve as embodiments of the slot based modular mesh framework, see FIGS. **16**, **17**, **23**, **26**. The smart phone is a candidate, in one embodiment. An IR chirp based transmission can be picked up on the IR "slot" and forwarded through IR (as in single radio mode, FIG. **39**, **3940**). Or the phone and/or receiving node in some embodiments may bridge IR and Wi-Fi Slots, see FIG. **22**, **23**. Or parts of the transmission may be over IR interspersed with Wi-Fi, where IR was not available. Further they may serve as temporal common routers **3952** to provide intermittent connectivity to otherwise isolated temporal or mobile networks, **3940**, operating on their own private channels and dialects.

The physics of wireless communications also favor smaller, close knit semi-autonomous "village" clusters, Reduced radio power reduces the range but also adjacent channel interference (adaptive power control is described in Ser. No. 10/434,948). A kitchen chirp aware nightlight/router embodiment thus supports a small, select chirp family, operating quietly on a common channel and possibly with their own machine Esperanto, in one embodiment. Common routers FIG. **39**, **3952**, and their agents provide intermittent connectivity to these largely self-sufficient clusters. A matrix of collaborating yet largely autonomous and scalable ant-like communities emerges.

In this embodiment, the routing overhead for all such rooted trees in the "park" would be O (n*r) where n is the size of a representative sub tree and r is the number of "root" nodes servicing them. In FIGS. **4**, **5**, r is two.

From the perspective of the mesh control, FIGS. **12** and **20**, distinctions between wired or wireless cease to be relevant. In one embodiment of the N-Logical radio concept, a bank of logical radios/transceivers **4130**, FIG. **41** supports multiple otherwise isolated trees through common router functionality. Each bank is a "switch" with dynamically reconfigurable slots. A slot in the switch is equivalent to a slot in FIG. **16**, **010**. FIGS. **12**,**16** has six such slots: four "radio" slots **010** and two Ethernet ports **020**. Similarly **4130**, **4270** depict six port configurable switches, in a switch stack hierarchy. Note that the two trees, wired and wireless, provide redundant fail over functionality. FIG. **16** shows a six slot switch in one embodiment. One of the 4 radio slots **010** will support wireless backhaul services. Separately, the two Ethernet ports **020**, are dynamically configured to provide the wire-based uplink and downlink backhaul, see FIG. **42**, **4260**. Thus, while in a tree based topology, routing is limited to North-South, adding another set of logical radios now includes "East", "West".

Switch port embodiments support (intermittent) wired and wireless connectivity. In one embodiment, a single-radio unit, **4110**, has been successfully reprogrammed to provide a U, D, S capabilities, singly and in combination. The switch ports themselves are also reprogrammed so that some ports may be configured to provide 24V Power Over Ethernet (POE) to the single radio units. Note that units **4140**, **4150**, **4145** logical radio agents U, D, S may be serviced by one physical radio **4110**, see FIGS. **14**,**15**,**39**, **40**.

In one embodiment, a logical radio agent S, **4150** hears uplink **4120** operating on a different channel than downlink **4140** is currently on—and therefore cannot "hear" uplink **4120**. The Scan Function, FIG. **12**, communicates this with the adjacent Mesh Control, FIG. **12**. Downlink **4140** is directed to change its channel temporarily to provide intermittent service to Uplink **4120**. Connectivity is intermittent: both uplink and downlink may also be servicing other clients, at other times, per the collaborative scheduling and queue/stack management, see 61/555,400. Buffering packets during scan requests is described in Ser. No. 11/818,889.

FIG. **42**, depicts an embodiment showing a "wired" equivalent tree to **4160**, Multiple wired and wireless links, **4160** and **4260** may concurrently exist, providing intermittent connectivity to isolated clusters. **4260**. In this embodiment, a common router **4270** has two uplinks, but operating in orthogonal domains of wired and wireless and hence permitted by the mesh control layer, responsible for ensuring tree based (non-cyclic) routing. **4270** may thus also provide bridging services e.g. for IR transceivers, see FIGS. **22**, and **23**.

FIG. **43** is a schematic of how the logical radio abstractions may be combined to create more complex abstractions. **4320** refers to two abstractions AP (also in FIGS. **12**, **17**). The "bridge" is a combined logical radio abstraction, similar to the U+D backhaul, FIG. **17**, but bridging over disparate frequencies and protocols. FIG. **44** shows the bridging function (as described in FIG. **43**). Mobile node **4455** switches from "blue" 5.8 G backhaul to a "pink" 2.4 G backhaul. The sub tree beginning with mobile node **4457** is thus operating on a non interfering channel/frequency/protocol. The static counterpart is **4460**. Thus, private networks are formed, occasionally bridge (FIG. **39**, **3930**, **3952**) but for the most part operate autonomously.

Mission critical mesh networks favor more "channels", for wired-wireless failovers, FIG. **42**. Embodiments installed in underground mining tunnels, for example, deploy a string of mesh nodes, FIG. **16**, to provide voice and video over multiple hops deep inside mines. Each mesh node supports both wire and wireless up links and downlinks at each node in the chain, see FIG. **39**, **3940**. Traffic is cloned to travel along both parallel pathways. On arrival at each node it will be forwarded on the most reliable link, wired or wireless and so on. Thus a packet may crossover from wired to wireless (where the wire has been cut) and back multiple times. The duplicate packets, like duplicate heart beat broadcasts, are discarded by the destination station's parent, a mesh node, FIG. **16**.

Embodiments employing "string of pearls" configurations are also used in mobile military applications. In FIG. **45** upper, a mobile unit makes intermittent connectivity to each static mesh node in turn. This is managed by the Scanning radio functions, FIG. **12**, described in Ser. No. 11/818,889. Note that mobile unit backhaul connectivity is intermittent, but the output is not. Real time video streams are being seamlessly "switched" to the mobile unit jitter free and without interruption, see unbroken throughput graph. This was "raw" video—the efficiency enhancements described in 61/117,502 would further improve performance. Chirp based control packets were also exchanged during this exercise, without interruption, using IP packets formats, see FIGS. **34-36**.

The process was repeated with single physical radios in a chain, FIG. **39**, **3920**. Bandwidth degraded, as expected, but the system still provided uninterrupted, jitter free, video, due to proactive SCAN agents, FIG. **12**, Logical Radio Abstractions, FIGS. **38** through **42** and the benefits of O(n) tree based routing, FIGS. **4** through **8**.

Extensible Network Management

FIG. **46**, a reprint of FIG. 10, Ser. No. 10/434,948, explains why network latency and topology are inter-related and hence relevant to latency sensitive VOIP/Chirp bus delivery schedules. Ser. No. 10/434,948, teaches a round robin approach where, the AP services each client in turn. In that example, 10 ms is the (equal) service duration for individual client services. Packets are buffered till a round robin cycle is complete. At the end of each cycle the container is sent, per up the tree, in one embodiment. The local latency upper bound would therefore be 30 ms for section **70**. By the same token, the root node is servicing 5 clients and hence the upper bound to reach the root node is 50 ms for sections **70**, **80**. Further, the example pointed out that, had all the nodes been connected directly to the root node, the latency would be 90 ms. 1 hop networks—all clients connected to a root node—are not necessarily "better". A five-hop string-of-pearls, O(n) routing scheme, FIG. **45**, may provide better service, than the 1-hop 5-client star, FIG. **46**, Section **80**.

Thus, in one embodiment, bus delivery schedules are driven by the round robin delay caused by servicing siblings, at each sub tree along a route. More siblings imply more latency and favor node/device migrations. Accordingly, network topology is dynamically modified based on toll costs of larger "families" see FIG. **1** and Appendix A.

In FIG. **46**, nodes operated independently and asynchronously but based on a common multiple of some service time interval e.g. 10 ms. Minimum Bus Intervals vary, based in the number of siblings. In Ser. No. 11/266,884, FIG. 24, the bus interval is set. Buses leave at preset intervals, regardless of whether the bus is full or not. In more efficient implementations, the departure time is flexible, and buses may wait till more passengers arrive, within a prescribed waiting limit, see 61/555,400. Thus the stream and CSMA allocations are based on "stacking", in dynamic alignment with "Customer Satisfaction".

In FIG. **47**, the bulk bus transport stream **4720** is first, during which all clients can listen but not talk, see FIGS. **12**, **24**. The remainder time **4750** is used to transfer data back from IP based clients to nodes etc. Further, FIG. **25** shows separate "channels" for concurrent transmissions. Contention is reduced during the Stream section, **4720** and possibly more. SCAN agent, FIG. **12**, measures overall activity of disparate packet types (a form of "channel list"). Note that in

Wi-Fi standard infrastructure mode, all communications are with the AP, hence tree based routing is inherent.

In one embodiment, regular chirps/heart beats are received by the node through one of its logical "slot" interfaces. IP traffic is also received from a slot interface. Both data types are then priority queued for onward transmission. Further, the data may be limited for local consumption, e.g. regional streams, FIG. **48**, or sent upwards and/or downwards e.g. Global streams, FIG. **49**. Thus, the amount of traffic flowing through the network is lumped together. The ability to identify different traffic regions, their locations and patterns is therefore of value to network administrators and VOIP/chirp device manufacturers alike. The ability to record and play back sequential snapshots of network topology changes is also relevant to simulation, diagnostics and adaptive learning.

FIG. **50** depicts the stream reader, in one embodiment. Special purpose Stream Readers are privy to data traffic queued for transmission at a node. Like post office sorters, they identify and sort the "mail" and therefore, help to collectively define bus schedules, reduce dead letter re-transmissions etc. Stream readers, resident at the node, feel the "pulse" and therefore provide early warning signals to the Mesh Control Layer. More "mail" from one node may increase toll cost for other child nodes to switch parents, using load balancing and adaptive power control methods, Ser. No. 10/434,948 and FIG. **1**. The use of resident agents in Access Points, to manage flow, was also first introduced in Ser. No. 10/434,948

Stream readers can feed multiple stream viewers, **5040**, sequential readers/agents, **5060** or a logging database **5080** community mailbox etc. A circuit of collaborating stream readers and subscribers emulate complex logistic supply chains, see FIG. **51**, **5120**. Disparate traffic data is sent to knowledge repositories **5180**. Secure Control lines from it, **5170**, drive sub-circuit behaviors and their outputs, **5125**, **5135**, **5155**. Repositories may also provide the secret handshakes needed by readers to correctly decode public network traffic **5150**. Thus bulk network traffic, **5150** may employ little or no encryption and thus be lightweight, like pollen. Further, the Network Viewer **5190** may be connected to the real time stream plug in circuit **5145** or run it in playback mode, **5165** from knowledge repositories **5180**. The same circuit based framework **5120** may also provide interleaved real time and historical trending, simulations and machine learning, FIG. **52**, **5290** and 61/555,400. Note that **5155** is not connected—it is in "connectionless" broadcast/multicast mode. Thus both direct and indirect subscription styles are supported within the same stream reader framework.

From a control systems perspective, a network management system (NMS) receives node heart beat data and provides snap shot views for both human and automated agents. The circuit based approach engenders rewiring—swapping in/out data sources, or using consensual data from multiple readers to drive inferences, see 61/555,400. They may also be used to monitor different types of streams in addition to the Heart beat streams published by the nodes. For example, in FIG. **55**, **5520**, a custom heart beat was introduced in the Settings Viewport, for bidirectional Machine to Machine (M2M) data streams inside mobile/isolated man and machine clusters, FIG. **39**, **44**, **45**. This enables both human and automated agents to monitor and control remote machinery and their operators. Further, FIG. **55**, **5530** depicts the current mesh topology in dark lines. The lighter lines are alternatives gleamed from mesh node heart beats, which in turn was gleamed from SCAN agents, FIG. **12**. Also in FIG. **12**, the Packet Classifier **010** and VOIP Concatenation Engine, **020** are particularly relevant for terse M2M messaging. Their status/health is also monitored, **5520**. The client activity and alerts, **5540** are generated by M2M readers at the node interface, which in turn are received by subscriber agents. They generate the alert for the machine maintenance subscribers.

Since the health of a network is only as good as its participants, there exists a need for an extensible and open framework for rapidly developing means to view, within one dashboard, salient or related behaviors of complex man and machine networks—especially when they are intertwined. Ideally, a comprehensive open network management system manages the health of the network routers (and it components, FIG. **12**) and also the health of its clients: humans and devices. This closer relationship ensures dynamic alignment in fast changing pace of global supply/demand chain of data flow.

One embodiment of an open extensible Stream Reader Framework, FIG. **52**, is implemented in Java. A subset, JavaScript API **5230**, provides dynamic and customizable HTML based views. Custom Stream viewers define the GUI for different devices and form factors. More complex business logic applications use the Enterprise class Java API and Repository **5240**. Third party adapters and applications, **5250**, extend the network to consumers/providers of information and their viewers.

FIGS. **53** and **54** depicts the published interfaces for the Network Manager Streams API and the Heart Beat Entity relationships, respectively. Together, they enable speedy viewport development, FIG. **55**, for extensible human and agent collaborations e.g. FIGS. **1**, **14**, **15**, **20**, **24**, **27**, **28**, **39**, **42**, **44**, **45**.

Organic Chirp Protocols

Upper and Lower Control Loops

As networking continue to evolve, the burden on the IP packet routing infrastructure increases. Moore's law has helped with processing speed, but Moore's law is linear, while Metcalfe's law is exponential. As networks grow, O(n2) growth will overtake O(n) systems. It has not happened yet—while we hear talk of how the Internet is broken, business continues as usual. The growth of IoT is burgeoning and is occurring at the edge of the network. With IoT, a tipping point can occur, as the burgeoning M2M communication with cloud based services consume IP bandwidth at exponentially faster rates—both within local M2M social networks and the larger M2M social networks caused by multiple layers of publishers and subscribers.

Some new IP based packet handlers will need to be invented to stem the deluge of largely repetitive and non critical data. Since sensors do not typically have large memory, sending it up will be the common option. Periodic IP connectivity will be mandated.

The IoT version of the publish/subscribe world of pollen and flowers is significantly different from our traditional IP based networking with its roots in point to point communications:

Its usually unidirectional—many chirping devices, may not have listening capabilities, so the communication is one way only. Receiver address is then meaningless.

The IP packet header overhead is unacceptable for small chirp packets and the IP protocol was designed to be general purpose and static—you cannot use the bit stream to define your own genetic strands and its vocabularies. This is too restrictive for a burgeoning class of devices that simply chirp a few bits a day but want to be sure their chirps get to the right flower.

Since the same data source may be of interest to multiple subscribers, point to point communications make little sense—its simply kicking the can up to a central server that then still distributes data as part cloud services for subscribing Integrators. Contemporary thin client paradigms are based a more direct link between devices and integrators, via incumbent IP based traffic flow. Device (raw) data is forwarded up to big data cloud servers, where the economies of scale favor centralized processing. This is a top heavy architecture where point to point transport is convert into more usable publish-subscribe format only at the top.

A more balanced architecture, especially if device (raw) data is repetitive, is to provide some pruning/aggregation and exception handling closer to the publisher sources, rather than pay for it to travel upstream and then be discarded. If the devices in the thin client model can support it, it will be tempting to move some of that processing downstream, resulting in fatter clients. Putting more processing into every device does not favor economies of scale.

A more economic and efficient model would be to provide a central "hub" for device raw data to massaged into more useful forms for cloud servers. Big data subscribers are interested in massaged "small data" that fits into the format needed to feed the big data analytics. Intermediate agents fill this gap between raw data and big data. In short, some form of agent based local processing makes sense, regardless of whether devices are IP or chirp based. The control loop, is then split between the devices and agents and then the agents to their big data subscribers. In the thin client model, there was effectively one control loop between devices and servers. With the agent hub in place, devices may converse with the hub, in native dialects. The agents convert their bird speak in to a form more palatable to the vocabulary and formats of big data servers. The device publish stream is being converted into a small data stream, which big data servers are subscribing to, in addition to receiving raw data, if needed, on a case by case basis. The overall architecture is more scalable and more efficient.

In this distributed and balanced setting, a local agent of the big data server can manage what exceptions interest them. The task of pruning and aggregating is delegated to a lower level of control. Round tripping is obviated. Using the Mars Rover as an analogy, Houston is kept abreast of "interesting" developments but local control of sensors/actuators in handled autonomously by resident software agents. This obviates needless round tripping between the rover and Houston, providing a more equitable distribution of task and resources. This is more efficient since it also reduces both traffic and server load. The output is from agents is more palatable "small data"

Regardless of whether device communication is IP or chirp based, a layered control loop, (with agents as intermediaries acting as the translation mechanisms between the upper and lower control loops) is inherent more efficient than round tripping. Some devices, like smart phones, are inherently chirp capable (e.g. IR and Wi-Fi) and can participate in both control loops, acting a bridge between the two banks of the river, each with their own control loops.

Beyond round trip latency considerations, there is a more fundamental reason for this tiered control and communications model. The language and vocabulary of sensors is fundamentally diverse from that at the big data server level. Sensors publish their limited view of the world, while "Big" data provides insights into a more comprehensive world view, incorporating multiple sensor streams, past history, future trends etc. Since function dictates language and vocabulary, some form of translation is required—one cannot expect purpose built machinery to communicate directly without translation.

In the contemporary, thin client model, FIG. **56**, Left, that translation takes place in the cloud, where data is sent in a format palatable to big data consumers. Needless to say, that puts the onus on the machines and their M2M communication protocols to be intelligible. What was a terse purpose built dialect now has to be interpreted in a device abstracted language. Agents and their location within the lower control loop reduce this burden, FIG. **56** Right.

An intermediate agent based architecture is also closer to publish-subscribe frameworks that big data systems are familiar with. Cloud servers, through web services, subscribe to multiple sources of data. Big data systems may be viewed as market places where publishers/subscribers or data providers and consumers meet and exchange. The "exchange" is one service that enterprise middleware software provide at Layers 7 and above on the network stack. For example, Tibco (www.tibco.com) provides a platform where real time feeds are both published and consumed. Multiple and diverse applications employ generic and extensible real time publish/subscribe "exchange" infrastructure to conduct business.

Another type of "exchange" is burgeoning at the lowest edge of the network, see FIGS. **57** through **58**. M2M communications is rapidly evolving into its own form of localized publish/subscribe exchange, with its own function driven vocabulary. This community of sensors and actuators will need for their own, potentially private/isolated exchange. Many to one relationships, like any publish/subscribe framework needs to be supported. The smaller communities coalesce into a community tree of communities, but with flattened information flow, akin to bus routes along the network of mesh nodes and their agents.

FIG. **59** is an example of a small community of sensors, providing, vital sign inferences and warnings gleaned from pattern/trends of multiple sensor streams. A field of such local integrators, generating small data, feed into progressively larger streams, see FIG. **60**. Here, a composite view of the current ground moisture level is integrated with weather forecasts of likely rain to direct whether some areas of the field need additional water.

Simplifying the transport mechanisms of sensors and actuators has profound and compounded network effects. If sensors are no longer burdened by an incumbent IP protocol, not designed to be inherently publish-subscribe or exchange based, then they than free chirp is terse, purpose built organically grown, collaborative communities. The Dewey Decimal system, for example, enables terse content classification shared across all library "exchanges". Its analogous implementation could function as pollen category classifiers. Now, raw sensor data immediately becomes sort, search and publish worthy. Further, the decision to combine multiple sensors into a composite firmware package can be validate by subscriber trends indicating their interest in a local integrator/small data generator. The circuitry for the Chirp transmitter on the propagator/integrator is then shared, see FIG. **61** and the combined device provides additional functionality, FIG. **62** shows an integrated multi-sensor package.

FIG. **63** shows a network of chirp propagator nodes. Each of the elements in the four leaf clover configuration are transceivers. The transceivers, in one embodiment, are identical, so that uplink, downlink and scanning functionalities may be logically assigned. The network topology is flexible, when required, see FIG. **1**. One layer of this 4 element propagator services chirp IR traffic. Layers are stackable, since they are operating on different, non interfering wire-less medium (e.g.

IR and Wi-Fi). When Chirp and IP traffic are operating in the same unlicensed band, then the propagator proactively reserves time slices for chirp devices, using well understood reservation techniques (e.g. RTS/CTS).

Control Loop Polling Intervals

The social network or community of M2M participants, will, over time form their own real time market place, with links to the enterprise level market places in the cloud. Some form of linking the two market places is needed, without forcing either to coalesce into a single control loop, thin client model. As an analogy, the Mars Rover is a single entity functioning semi-autonomously with at least two control loops. The lower control loop, operating every few milliseconds, closes a tight, purpose built, isochronous loop between sensors and motors, keeping the Rover on track with its planned destination. The higher control loop, operating in minutes (between Earth and Mars), cannot be used for tight motor control. It is, instead, used to provide planned destination coordinates, for example, which is then translated to lower level commands and activities by the intermediate translation layer between the two control loops.

Compare, in FIG. **58**, the overhead of centralized process biased approach on control loops and associated latencies. In the thin client model, latencies are compounded by those the non deterministic latency of IP CSMA/CA and CSMA/CD protocols, over multiple hops from source to destination. In the dual control model, there is a membrane separating the upper and lower control loops of IP traffic and Chirp traffic (includes IP devices operating as chirp devices, when needed). Data flow between the membrane is publish/subscribe based. The Propagator hub manages bus schedules in both directions so both control loops are satisfied. Ideally, neither should be waiting on each other. For this to work, the two control loops can run on different frequencies but both need to be aware of the timing diagram needs of the other.

One implementation requires hubs to publish their incoming traffic schedules upstream. Upstream ascendants on the mesh network or routing path, can then back calculate when a chirp bus of aggregated packets needs to arrive so their "stock" feeding into their business processes, is maintain in a regular, non-disruptive rate. It can communicate that "demand" to the "supplier" node. The aggregated demand at the supplier drives it to be in dynamic alignment with its subscriber. This may involve more frequent smaller shuttles. Conversely, it may dictate convoys but will lower frequency. The two control loops are collaboratively changing their polling intervals if needed.

Layered control systems often have set polling intervals, generally a multiple of the lowest loop e.g. the sensor/actuator level. In thin client situations, with no pruning/aggregation and exception handling, the need to close loops quickly is understandable. Hence the historical need for tighter polling at upper control layers. With agent based delegation to the lower control layers, Mars Rover successfully converted Task level commands, sent asynchronously, with minute delays, to drive low level isochronous control sensor-actuator loops, operating in milliseconds. Upper control loops no longer have to be "tight"

When inevitable packet transport failures to accommodate do occur, the degradation in performance of the overall system is still graceful, because the small data generation uses pattern matching and inferences, sometimes operating in a composite device, FIG. **62**. Hence early warning systems are activated sooner, at the local level. Local evasive action within local M2M sensor/actuator communities, obviates adverse network effects. Disruptions are both graceful and contained.

A Classification Based Protocol for the Edge Network

With intermediary agents and membrane in place, individual sensors and actuators may simply evolve their language and vocabulary for what they were designed for them to do. This intermediate agent service then bridges the gap between raw data and big data. Now, devices, using their own, proprietary pollen formats can chirp in their own dialects—no standard common language is needed. Devices can remain simple, agents can be arbitrarily complex.

Further, nothing stops devices operating in both networks and forming their own bridges. IP agents can decipher the IP encapsulation where the payload is a aggregation of chirps and perpetuate the publishing beyond chirp membranes.

Beyond efficiency (large packet formats etc) there is a more fundamental reason to support a different transport protocol, rather than couch a new description language inside the payload section of an IP packet. But it is still a destination based routing protocol. It is not inherently publish/subscribe

Packet type ID in the packet header provide the information needed to drive the routing according to associated packet handlers. Adding new packet handlers vocabulary and protocols for IP based M2M communities poses a challenge. Routers have to know how to route these new types of packets. The new "agents" need to be accessible across the entire router network core and edge routers, including legacy routers.

Further, while there are known classifications for sensor types, but its an evolving field—providing specialized packet handlers within the routers, to handle their routing needs is not practical. Standards committee processes are long, tedious and largely controlled by the larger networking companies, with a vested interest in maintaining the status quo with IP for everything. New devices, such as composite sensor arrays FIG. **62** and private, semi autonomous communities, FIG. **63** are just beginning They will require their own private, terse, tight control loop. Some organic evolution is called for.

The primary reason for chirp based devices is their inherent simplicity and that protocol may organically evolve to support device categories not yet dreamed of, let alone how they interact with us and the world. Burdening these publisher/subscribers with the detritus and restrictions of a solely IP based transport is simply too small a canvas for the developers of these brave new products to create in.

IP formats were designed for coarse classifications of packet type routing handlers: e.g. Voice, Video, Web browsing, File transfer. Application specific granularity (Devices→Sensors→Moisture→Device-Type-A) cannot be easily expressed in a format intended to address sender oriented communications based on IP addresses and MAC-IDS. The type of data may be expressed in the payload section, but peering in payloads, slows down traffic. There are inherent limitations to sender oriented traffic flow.

Nature uses a Receiver oriented network. Pollen publishers have no receiver address per se, nor do they know where their ultimate destination will be. A pollen category based identification scheme is receiver oriented—the pollen simply travels in all applicable directions, it is not destination based. It is the onus of the subscriber to accept (flower) or reject (sneezing) the pollen. Category Classification publishing connect pollen to flowers in an inherently more direct manner.

Routing Based on Category Classifications.

What would such an extensible protocol look like? In Nature's DNA sequencing, there are strands of genetic code that are recognizable. Sometimes there is a marker that helps align samples so we can see the strands repeat across specimens. Genetic fingerprinting is extensible.

In Nature's world of publish/subscribe, pollen is being published for subscriber flowers. Pollination is essentially a selective pattern match. The same logic will now we applied to the IoT Publish/Subscribe world. Here chirp pollen, eventually arrive at a bus station. Chirp pollen have no idea where to go—this is a female/receiver oriented world. What does the chirp bus station hub need, in order to perform its function? It needs to know:

Where the flower in search of this pollen/passenger is and

When the flower requires the pollen/passenger to arrive there.

It can then drive the bus schedules to ensure an equitable compromise between bus size, its frequency and cost of IP during different times. Smaller bus loads will leave more frequently, for passengers in a rush, others will be compensated by a lower bus price for travelling on the larger but less frequent convoys.

Thus supply and demand of the chirp packets and its arrival is driven by dynamic subscriber demands. Scheduling of packets is covered under Collaborative Scheduling, in a referenced application. It relates the moving packets closer to delivery schedules ahead of others less in a rush. This is a dynamic form of prioritizing, based on pollen life and subscriber demand.

For now, we return to challenge of putting pollen/school children on the correct buses. At the bus station hub, pollen/passenger will collect and should be directed the bus best suited for them. This must be determined largely by public information, provided by the chirp/pollen, as its genus type, depicted by a DNA like strand of data in its public category bit stream.

Like DNA markers, only the flower knows where to look for the markers and what they are, to determine if there is match. Further, there can be small changes in the data—it does not have to be error free. As long as the markers are not corrupted, the faulty data will still find its way to bus station. There, it may be examined further, to determine if it should be put on the bus or discarded.

If Markers are corrupted, they may still be relevant, if a gradation scheme is used. Thus, consider a 4 bit marker 1.0.1.1. Assume that it is mistaken read as 1.1.1.1. (e.g. **15** instead of **11**) Pattern **15** requires Byte 1 in the category field to be A, B, C or D. Thus matching sequences in the DNA strand will eventually result in this pollen pruned and sent to a dead letter section. Since chirps are repetitive, this loss is not critical. Note however that false positives are being progressively pruned—before chirps board IP buses.

Using markers in a relative manner, as opposed to fixed settings in an IP packet is a light form of security, like pollen—only the intended flower knows how to decipher the payload. Note that "markers" between "packet header" fields is fuzzy, unlike IP packet headers. This makes security "light".

Propagators simply need to know what direction to send the data—up or down the network tree. This is not complex in a tree structure with O(n) routing. Recall, this is not a peer to peer network, requiring O(n2) computation of routing paths, as suggested by traditional sensor networks, e.g. Zigbee. Thus, the direction (up/down) suffices in tree structures. And the direction should point to where subscribers are.

In addition to aggregation, propagators, in conjunction with their subscriber preferences may also be needed to perform pruning. Traffic flowing upwards from remote moisture sensors in the Wine country in France, to an Amazon hosted cloud service in the United States, could well be small, but, given the number of such sensors, the IP traffic is significant. IP traffic is not free: some means to control what is sent over IP is needed, specifically, prune repetitive data, close to its source (as opposed to the cloud server/Integrator).

Local pruning and aggregation favor placing agents closer to the sensor raw published stream. Here the subscribers/their agents have more control over what they want sent to them. A subscription model would defray the cost of transport and pruning.

Skeletal Architecture of Chirp Packets

Locating the subscribers efficiently and developing the correct bus schedules and routes, is of common interest to both pollen and flowers. As in spring, there is a finite time of life for the pollen to have value to flowers.

Propagators, need some pollen genus category description to enable the match making What does this descriptor look like? As an analogy we return to bird chirps. We have categorized bird chirp sounds, based on studying individual bird categories. We can identify the bird type from its chirp/tune/melody. Hence those subscribers interested in melodies from doves, can now receive those recordings, based on bird category. The categories will have to support different levels of granularity—some bird enthusiasts are only interested in doves near their homes. Hence the category field, should be sufficiently flexible in design, to support further drill down.

In both melody/tunes and DNA structures, there exist "Marker" strands of information that provide a common pattern across members of the category. These markers occur at specific locations and are of specific patterns. Thus, some categories may be described as those that have a 8 bit marker, always in the 4th byte of the bit stream. Thus the classification could be 4.8.XX.XX, where XX are more levels of granularity that may be gleaned from knowing a specific pattern type and what it entails, in terms of how to further classify the public (no security but not necessarily universally decipherable) category field.

Consider the 8 bit pattern is 1.1.1.1.1.1.1.1 or 255. 255 may indicate a pattern where the 4 bytes are 4 one byte classification sub categories. Thus a 4 byte genetic strand may now be interpreted as A.B.C.D, where the letters occupy one byte each. The complete category is thus 4.8.255.A.B.C.D. In one embodiment 4.8.255 is publically accessible pattern information and at the start of the bit stream, the location and size of the marker (255) is specified, as opposed to at the end of the public category shown in FIG. **56**. This enables a quick bit mask to look for all publishers in categories 4.8.255 etc. Those capable of providing further granularity in the chirp signature will need to access pattern that provides the map or implicit field markers for A.B.C.D within the category strand. Thus, in one embodiment, all of the following provide deeper classifications of the pollen/chirp/uniform:

4.

4.8.

4.8.255

4.8.255.A

4.8.255.A.B . . .

Thus Propagators, depending on their access to internal field data, can always provide at least 3 levels of addressing (e.g. 4.8.255) and potentially the complete DNA strand/signature. That may be sufficient for coarse aggregation: pollen of the same feather may be flocked together. At least three levels of granularity in bus schedules are supported. Larger, infrequent convoys cover 4.8.XX categories, while smaller shuttles for more frequently requested data can specify precisely what is of interest e.g. 4.8.255.A.B.C.XX Pollen categories thus drive the bus loads, their contents and their frequency, at differing levels of granularity.

Note that A.B.C.D is distinct from B.A.C.D. In general there are 4*3*2*1 or 24 non null combinations for a 4 letter vocabulary A, B, C, D.

In another embodiment the set of 255 8 bit pattern markers denote eight 4 bit markers. Now the pattern arrangements are 8*7*6 . . . =40,320 arrangements of 8 letter vocabulary: Since 4 bits support up to 15 non null numbers, 8 distinct letters are easily supported within 4 bits. Like DNA, the vocabulary may be terse and small, but the patterns depicting the category are not. Small data can be rich in content.

Incognito Pollen:

Some pollen may need to travel incognito. That is, they expect propagators to rebroadcast them, potentially in all directions, till an agent discovers them.

A "4.0" category pollen implies a marker at byte 4 but its length is not specified. Agents with bit mask filtering can locate such semi-incognito pollen, they know what the marker is. Note the marker can be arbitrarily long or short. Short markers increase the occurrence of false positives with other marker types (e.g. 4 bit markers 1.0.1.1 sharing same 4 bits as 8 bit marker 1.0.1.1.0.0.1.1). Agents that have this level of information can also glean other data from the packet melody/strands, to filter them out.

A "0.0" category pollen does not specify either the location or size of the marker. This is completely incognito and the propagator may continue to rebroadcast the pollen both up and down the tree till it reaches either roots or leaves of the tree. Chirp devices have no access to the IP network except through the bridging propagators. An agent at the chirp interfaces is either present or the pollen dies. It cannot cross the Chirp Membrane. Hence IP traffic congestion is obviated.

In some situations, a 0.0 pollen may wish to specify a direction and nothing else—e.g. up or down the tree. Thus, in one embodiment 0.0.1 pollen are upwards, while 0.0.2 pollen are downwards. Since each category has its own vocabulary and language, 0.0 chirp families may choose to use the next two bits to define direction (0.0.0.1, 0.0.1.0), as opposed to marker size. Languages defining how the strands of DNA comprised by bit streams are both versatile and secure—since it is generally receiver oriented.

IP based sensors and their traffic may also use category patterns as part of their data classification schemes. In that case IP based packet headers will specify the sensor MAC_ID or serial number within the payload, in addition to the category classification for IP based agents in the cloud servers or local to the IP interface of the propagator. By the same token chirp devices, may, in their private payload or public category type, contain an IP address where they wish their pollen to be sent. The chirp interface of the propagator receives this chirp, local agents/apps decipher it, prune it, repackage as needed for IP highways.

Individual Information within Chirp Signature

Beyond category information, bird chirps carry individual/private information, see FIG. **56**.

Nature's random number generator changes the individual birds chirp tonal qualities, governing each bird. This serves as a form of identification. Thus, mother birds know each of their children's distinctive chirps, though all are using the same chirp family packet format and its associated shared vocabulary.

The IoT counterpart of this individual identification may include:

1. Chirps at distinctively different patterns (e.g. tunes)
2. Public category classification, include some ID e.g. last 4 digits of the manufacturer SKU number of the device.
3. Lineage based—e.g. child of kitchen router.
4. Location based: e.g. located in kitchen, close to toaster. (from signal strength analysis)

Note the combination of chirp tune, its last 4 SKU, location and lineage *collectively* define a distinct bird/sensor. While none are unique, their combination is sufficiently distinctive. Uniqueness is not required.

The combinations have inherent randomness since their constituent elements (e.g. transmission pattern of chirps) are random. They are not unique, like IP or MAC ID addresses, so there is no burden of maintaining a global database. Pretty good distinction in the bird chirp is sufficient for the mother. By the same token, pretty good distinction is sufficient for the bus drivers e.g. propagators.

Note that individual data, typically in the private section, may also be present in the public section. Thus, some common types of sensors, e.g. Moisture, may not need to a private section: the data may not need to be secured.

Transmission Agility Information

In the event chirps share the same Wi-Fi medium, one part of the public category section needs to contain chirp transmission characteristics. In other words the "uniform" should support network agility. Smarter, more agile devices can become aware that simpler chirp devices will be active at intervals specified. Thus, data related to when and how often the birds chirp and what pattern they use (as in melody/tunes or rhythm) is needed by both propagators and agile devices to ensure they the network can anticipate and hear chirps distinctly and without co-located interference on the same "channel" from other devices. Note that in nature, bird chirps are syncopated—birds are cognizant of each other. This data also gives propagators the option to shift smarter, more agile chirp devices to other times. Or the propagator, after review of local client device transmission patterns, can request a change to the dipswitch settings of a device. The devices that support such flexibility, is again, part of the Pattern Marker et al. Thus, after some tuning, there may be sufficient distinction in the melodies so mother propagators can easily recognize individual offspring.

Extensible, Non Unique, Pattern Driven Chirp Signatures

A skeletal view of the Pollen "Uniform" emerges. It contains patterns, defining other patterns, each of which provide a more refined level of detail. Access of that detail can be controlled to answer:

1. What type of pollen is being transported (pollen category)
2. How often is this data published
3. What is its publishing frequency pattern (may be dynamic, may need observation over time, implying learning and discovery)
4. Distinguishing features of individual chirp devices e.g. serial number, location, lineage)
5. Information on transmission pattern so agile devices can share the same medium without interference.

Note that all of the above is easily discernable by rudimentary bit masks—that is, if you know what pattern you are looking for. Thus, in one embodiment a propagator agent is instructed to look at bit location **13** in all 4.8.11 packets. If it is set to 1, that is a universal flag for "unit malfunction, type 1". The agent is required to convert that information into an IP packet and forward to the manufacturer specified in a byte 1 (e.g. A in 4.8.11.A.B.C.D).

The public section defines the chirp/pollen category, needed for bus scheduling and packaging of packets. This is sort of like the bird category. Without this category information, the propagator does not know which direction to send the packets, as in which bus route to employ going up, down trees and where to clone more packets for multi-cast transmissions when multiple subscribers exist.

The second, typically private, section is the message—what a particular bird is saying and some (typically private) information about this particular bird. It uses the same concepts as the public section, but it has its own locations of where markers are, what those pattern signify and hence where the implicit field markers are. The 4.8.11.A.B.C.D family may use a completely different scheme for the private section than the 4.8.11.A.B.D.C strand/family.

The public and private parts of the chirp packet are separated by a publically known Public-Section-End-Marker. It is of variable length. In one embodiment, it may be 4 bits or 8, depending when whether 15 or 255 different types of (public) chirp patterns are needed. Note that if only 7 different markers are needed, then a 3 bit marker suffices (2**3=8).

Some default public markers will be provided, through consensus or standards bodies. These will be reserved for common use by sensor manufacturers of a specific category e.g. all moisture sensors manufacturers will use a category A.B.C.D.E.F, a 6 byte category address).

In that exemplary 6 byte genus family a end-of-public-section-marker will always appear at the end of the 6th byte. It will be one of the defined types. New types may be added as within the size of the marker in bits. Thus, if manufacturers feel that 255 distinct markers are needed to define a second level of granularity in the category classification, then they would jointly agree to support a 1 byte (8 bit) marker and its associated overhead.

Note that with 6 bytes there are a total of (2 exponent 8) exponent 6 or (248 −1) non null unique category assignments. Further, with an eight bit marker type, the sub type classification now supports (256 −2) category classes.

All chirp packets may not need 6 bytes of classifications. If it 4 bytes, then the marker appears at the 4 byte partition. How does one know where to look for the marker quickly and efficiently? In one embodiment, the first part of the public classification category contains both marker location and its size. For example, the classification category pattern with 6 bytes would have the 8 bit marker position 6.8 stated so before the public section begins. Without knowledge of the location and size of the marker, the entire packet is gibberish, recall classifications are based on public marker location.

One embodiment reserves the first three bits of public category field to define the end marker location only and the fourth bit is the size of the marker. If both are specified then both marker location and contents are extractable. The category field can then be deciphered based on the marker pattern description. The Marker pointer is shown in the start of the packet but is also a variable. If some propagators do not know of this new type of bird chirp, they will simply send it upwards. This is one way pollen can direct the winds that carry it.

In one embodiment, the first three 3 bits for the Marker position support **7** marker locations. These may adequately express the location of the marker, e.g. 0, 1, 2, 3, 4, 5, 6 and 7 bytes long.

Note that the same Marker Number (e.g. #**200**) provides diverse interpretations simply by its location. This is similar to DNA sequencing, where the location of the distinctive sequence helps recognize and align patterns.

Category Byte Size

Many simple devices may require only one byte for category (255 variants) and another 4 bit marker for the pattern type, see FIG. **56**. Thus each of the 255 category numbers may be interpreted up to 15 distinct ways. This allows for close to 212 interpretations of a 1 byte category field. Similarly, a 6 byte category/genus would allow for (248 −1) variants, each supporting 255 patterns (8 bit Marker). The "genetic code" describing a sensor category, may be expressed in multiple ways using this extensible pattern based format.

For non zero byte public sections, the marker type provides all the information needed to interpret it. The pattern defines where the content subsections/fields reside within the public section. Hence simple devices, may use a larger public section to include data, also public. Here no private section is needed/used.

A zero byte location is defined, in one embodiment, to mean that there is no public section. The Marker type points to a data pattern which provides the information needed to interpret the private section, following the (empty) public section. The Marker pattern is then being used to interpret what follows generally as payload. Thus flexible use of the Marker Pattern is supported, beyond its intended use. Thus a marker pattern and associated classification of the data packet may together constitute an IP packet payload. This is relevant to IP based sensor streams that prefer IP connectivity over Chirp to IP bridges.

Marker Pattern Templates

Sharing the same marker type, at the specific locations (e.g. 1 byte) engenders collaboration between manufacturers of the same sensor type. They may agree to jointly use a range of Marker types (e.g. **200**-**220**), which share common fields, but use other fields, both in the public and private sections to provide more detailed and/or secure information. An shared used marker pattern template emerges through this collaboration.

Creating a new marker type (#**221**), may not require the traditional central standards body review process because the repercussions are local. For example, introducing a new marker type in location **1**, affects only the 1 byte public category users. Within that, it affects those who wish to use the same marker pattern number. Contrast this with defining an new IP header format. IP headers must comply with IP requirements in order to be readable. This is potentially affecting all users.

The Marker Template is therefore a organically evolving pattern masking scheme that helps consumers delve deeper into the public section/category classification ID. As such, it loosely resembles IPV4 or IPV6, subsections of the entire IP address. Note however that IP addressing is destination based. After the packet reaches its destination, the payload is deciphered, in the cloud. Then the information, perhaps still device specific, is device abstracted. Then comes pruning and aggregation in the generation of small data. The small data is now publishable, within the distributed processing of big data servers (e.g. Hadoop based). It must now be inserted into the publish/subscribe framework of web baaed services.

With Chirp Category templates, small data is generation closer to the source, where it can have more impact in tighter sensing-control-actuation loops. And since it is category based, finer granularity is simply a matter of loading the appropriate agents, at any level within the chirp network or chirp-aware IP agents—that know how to look into IP encapsulations of aggregated small data.

In one embodiment, the category section this a bit stream with contiguous fields, like strands in a DNA sequence. Knowing how to look into it, helps decipher the bird chirp category better. But this requires more processing and is therefore intended for subscribers interested in finer granularity. As the pollen progresses up stream, it continues to be disseminated with finer targeting to the interested subscribers, who can also drill down themselves, if they prefer, by requesting broader category searches.

The lowest level of granularity needed for the bus drivers is simply the marker location and its number. Hence Byte 6, 4 bit Marker, value 1.0.1.1 is sufficient to get the pollen to Bus type 06.4.11. This is sufficient to get the pollen/chirp to a 6.X bus station.

At the bus station for 6.4 buses, specialized 6.4.11 travel agents may peruse the category pattern to uncover two more sub categories each of which, are specified by the pattern description to be 1 byte each. Now it is known that category is 6.4.11.250.250. Subscribers willing to pay for this level of detail are alerted. Thus chirp/pollen can be very specific in terms of where they want to go, using a variable pattern template structure.

Again, pollen is implicitly managing the winds that carry it, since manufacturers can decide where those agents are placed along the route starting with 6.4.11 and getting progressively finer.

Bus schedules are now driven by the bus load and when the packets need to delivered. This is related to the collaborative scheduling engine, where the coffee order and its delivery are aligned to provide lower stacking and better quality of service/Customer satisfaction. The size and content of the packets is being managed to ensure timely delivery in dynamically changing scenarios. This becomes a more tractable problem as more refinement into the pollen category is possible closer to the pollen publishers. However having pattern matching agents 6.8.001 through 6.8.255 (8 bit marker) resident at the bus station requires CPU processing—this may be an enterprise chirp router but overkill for the home.

Hence multiple types of propagators emerge, each with their own category types. Or SIM cards slots are provided, so categories of chirp packet handlers may be supported. Some packet handlers will be secured to specific hardware; others may be software agents/apps.

Thus, yet another means available to pollen, to direct their wind, is to specify the propagator category or agent type it is in search of Agents reside on propagators or in the cloud. Their existence is known internally within the propagators tree like mesh network. Hence going up or down the tree will eventually move the pollen towards its handler within the chirp network. The agent handler can then provide the small data that needs transport over the IP highway, via the chirp buses.

If the propagator has no category pattern agent then it will kick the can upstream. Alternately, in one embodiment, a public pattern template (e.g. 255, 15) specifies how detailed navigation details are in the category section, e.g. UP 3, DN 1, Left 2). Here the network topology is being used to find the flower this is relevant to static, secure networks, where the topology is managed. Note that the directions can encompass both the chirp network topology and it parent IP based network tree, in an hybrid mesh network comprising of both.

Routing Agents/Apps and Network Loading

The marker number is essentially a pointer to a look up table of patterns, see FIG. **56**. Device manufacturers have multiple options regarding how to use the marker patterns. In one embodiment, by mutual agreement, 15 and 255 denote navigation based information for 4 and 8 bit markers respectively, e.g. (North 4, East 1) Further in another embodiment, 14 and 254 denote IP addresses in 6.X classification schemes for 4 and 8 bit markers. Thus two public means are made available for pollen to specify their intended flowers. IoT pollen can drive the winds that transport them, all the way into IP land, without agents.

Pollen intentions can thus be explicitly encoded. Or it may be fuzzy, as in: "look for an agent that knows about 6.8.11.A.B". Here, the propagator network and its routing tables are needed. The routing table keeps track of where the clients are, includes chirp devices and chirp routing agents. Some chirp routing agents/apps may be on the Chirp to IP bridge, and capable of securely accessing the entire category fingerprint, peruse the contents and decide what to do with it.

The efficient path of the pollen is thus gated, filtered and then redirected at progressively finer sieves, akin to Zip Code classifications for mail. Letters that fit "standard" patterns (size and weight) are processed efficiently. Others will be dealt with after the simple stuff is completed—this is how greedy algorithms work. The price paid for the flexible chirp format is that non standard package types will emerge and must be handled albeit less efficiently.

As an example, in one network, there are a handful of byte 4.8.XX categories, others are all 2.4.XX or 6.8.XX. It would make sense to move the 4.8.XX agents to a propagator that handles more 4.8.XX buses. A 4.8.XX bus central hub emerges, at least temporarily, based on the center of gravity where 4.8.XX pollen and their subscribers are. Some chirps may have more hops to travel but by economies of scale, 4.8.XX bus deliveries and scheduling become easier and less costly. The system can support declining margins.

Dynamic loading on the network is examined by nodes of the hybrid mesh tree (of both IP and chirp devices), from the root of the IP tree, downwards. System administrator are notified, who can move the agents residing on the network nodes. This will alter the pollen path and streamline flow. Further, if the agent is mobile (as in not locked to a particular physical device) then the network can automatically move the agents to where overall traffic flows best. This is akin to changing mesh network topology to meet changing latency and throughput requirements.

Both the physical network topology and the logical network (based on where agents reside), eventually stabilize and learn to adapt the topology to provide stable, tunable bus schedules and routes. At each hop along the network tree, pollen in being pruned and aggregated. It is being pruned along the path, at each bus station, since pollen has a relevant life and may have outlived it. Or the subscribers have lost interest. It is being aggregated at each bus station hub along the route to meet bus schedules and economic passenger quota. Publish and Subscribe sides of the demand/supply chain are thus in dynamic alignment.

Propagator Node Networks and Operation

Trees come to mind when we think of Nature's scalable creations. Trees are older than Man and have a highly evolved networking structure that is both efficient and adaptive. The structure is recursive: Any part of the tree replicates the same structure. The underground roots are an inverted tree, branches are horizontal tree, all connected through the trunk. A network of trees, some "rooted" to the tree trunk, others through relay nodes. The logical and physical network of branches all follow one simple rule: the "uplink"—the head of the branch is always one. An pitch fork branch (one with three roots to the tree trunk) would be considered a freak of nature. It is this simple rule—one uplink only—that ensures O(n) routing. Scalable Networks are possible.

The tree structure is emulated in the IoT world envisaged. For example consider the propagators P**0**, P**1** . . . P**3** see FIG. **64**. P**0** is the "root" node since it has an access to the IP network bridge. P**0**-P**1**-P**2** form a string of pearls relay for chirp clients C**3** and C**4**. The both share the lineage P**0**.P**1**.P**2** and hence are identified as siblings. Lineage is part of their identity.

Propagators form sections of a sub tree, the simplest of which are a string of pearls e.g. P**0**→P**1**→P**2** forming a link in the chain requires at least two interfaces e.g. uplink and downlink transceivers. For example, P**1** slot **0**, is an uplink connecting to P**0** Slot **0**. P**2** slot **3** is a downlink providing connectivity to P**2** uplink slot **0**. By convention, Slot **0**, refers to the uplink, except for root nodes (P**0**). Root nodes have only downlinks—their uplink is the IP bridge.

The propagators are shown with 4 transceivers e.g. Infrared LEDs or other short range wire-less transceivers. They are placed in the general vicinity and with arbitrary orientation. Propagators periodically scan the environment and reorient/re-assign the slots so there is always one uplink connecting to a parent propagator node. The choice is based on the best available effective throughput, all the way back to a root node. The parent selections are not always smaller hop based. For example, P**2** may be able to "see" P**0** but throughput P**0**→P**2** is inferior to P**0**→P**1**→P**2**. In other words, Min (throughput P**0**→P**1**, throughput P**1**→P**2**) is still better than (throughput P**0**→P**2**). In the event it was not, P**2** would logically reorient its uplink, so Slot **3** would now be the Uplink facing Slot **0** of P**0**.

Thus, a 4 slot transceiver, with arbitrary orientation, may logically reassign slots **0** through **3** to ensure connectivity back to an upstream root node. The slots **0** to **3** are thus being dynamically reassigned to maintain an effective tree based network topology.

Managing Latency in Network Tree Topologies

Propagators are placed in locations where they can form tree branches as shown above, all the way to a "root" node, that bridges to the IP edge node. Chirp networks extend the edges of the IP network without adding additional overhead to the IP end of the network and also serving as pruning and aggregation agents for IP based subscribers. In effect, they present alternatives to the current IPV6 thin client model, which, even if intermittent connectivity is assured, requires data to travel up the cloud before its essence is extracted. The cost of IP transport is minimized. And the cost of more expensive and power hungry IP aware devices is concurrently avoided.

Some chirp data is more time sensitive than others. Kicking the can upstream to big data cloud servers, simply burdens both the edge and core network infrastructures. Further, Moore's law is linear, while Metcalfe's in O(n2). As the sensor data at the edge grows, processor speeds (for both routers and cloud servers) will not keep up. Hence M2M traffic latency will deteriorate or become more expensive. Hence some relief is needed at the source of the deluge—the edge of the network. Here pruning and aggregation of only "relevant" data brings things back into balance. Metcalfe Effects are kept in check.

The primary function of the propagators/relays nodes is to send upstream "relevant" data, which agents, residing in their firmware, have expressly requested. The agents know the categories of interest. Additionally there could an exclusion list provided—for categories of no interest. The nodes will record the existence of those categories in the network, but will not forward. This reduces up stream traffic.

One cannot always know, a priori, the type of categories of interest, any more that winds can always provide focused beams of pollen to their awaiting subscribers. Some discovery is needed, through, at the very least, notification, from "mother" propagators, that a new category of sensor "bird" chirps have become "active" in a geo location under its care (e.g. sub tree of network). Notification summaries of sensor activity would therefore be periodically provided. Interested subscribers can then direct their agents/apps to provide the level of granularity/aggregation/pruning/exception handling needed.

Over time, an agent based social network emerges, Agents would logically reside at main branches of sub trees, where there is sufficient data flow/analysis to justify to additional processing power needed in the propagator. It is thus tree based, in accordance with O(n) scalability. There would be consumer and enterprise versions of such propagators. In consumer versions, there would be limited agents at the node—most data may be pushed upstream to parent nodes and their agents, all the way to cloud servers. This multi-hop path and its associated latency may be inacceptable for some mission critical enterprise applications. In a previous era, Programmable Logic Controllers, (PLC) wired to sensors and actuators on the factory floor, managed the deluge of real time, latency sensitive M2M traffic, escalating only that which fell out of their rule based relay ladder logic diagrams used by PLCs. Today, that same approach may be applied to rule based agents residing on propagators, close to the sensor/actuators. This reduces latency for enterprise class M2M communications.

Switching Parents and Scan Management

At the base level, propagators are relays, connecting to a "root" node, thus forming a tree. On power up, its primitive behavior is to become associated with a parent, which provides a path up stream to the root node, the closer to the root node, the better. The preference may therefore be, at a rudimentary level, to connect to parents with a low hop count, 0 for the root, 1 for one removed etc. However, M2M traffic is moving mostly upstream, hence, there is more contention closer to the root. Hence, in addition to noting the candidate parent nodes within its wireless radio "zone", propagators must also be able to send a "probe request", to determine the signal quality for transmission. Additionally, it would need to know how many siblings it will need to contend with. Since siblings are part of their own sub trees, the descendants of those siblings are also, indirectly competing for the "mother" parent node's attention. In short, that is a lot of information to sift through before selecting a parent. And it would change in a few minutes, in dynamic or mobile environments. A simplified notification of the presence to a candidate parent is required. At the base level, connected Nodes transmit, through heartbeats, their "lineage" and "costs" of connectivity, e.g.

1. Their name
2. Their Parent's name
3. How many hops they are from the root node ("hop cost")
4. The "Toll Cost" of using this node—e.g. its availability
   a. Based on current processing power usage at node
   b. Based on number of active chirp and propagator descendants
   c. Its overall link quality of path back to root node
   d. Etc

NAME-PARENTNAME-HOPLEVEL-TOLLCOST thus defines a broadcast beacon, in one embodiment. Recall names are not unique, simply unique within a lineage sub tree. Hence node names, all the way up to the root, may be duplicated—as long as the lineage path remains unique. Thus two sibling nodes may not share the same node name. Hence a new node with the same name as a current child node, will not be allowed to join that sub tree.

The decision to join is then simplified to whether a prospective parent toll cost/hop cost ratio meets desired characteristics of current chirp packets that the orphan node would be transporting. The orphan node, does not actually know what that data profile would be—it has not yet joined a network.

It does have access to chirp devices in its vicinity and can perform rudimentary profile analysis, with the presumption that this is a representative sample set. Based on the profiling, if more latency is acceptable, higher hop cost would be acceptable. Else, a switch to a node closer to the root, but at a higher toll cost, will be initiated.

This is, at best, an approximation to an actual link quality, when connected, and then having actual chirp devices connected to it.

One is tempted to suggest that the propagator make a hasty connection and perhaps later, switch parents but this is costly. Nodes may switch parents constantly, causing local oscillations (switching back and forth between sub trees), which eventually percolate to the top.

"Mother" Nodes can therefore not leave the nest, while descendants are switching around—it would simply feed the chaos. Hence decay functions are built into the hierarchical control system that manages the network tree topology. Permission to switch parents travels at least as far up as the parent of both sub trees—since both are being affected by the switch. There, if there nodes have settled down from the previous switch, permission is granted.

In order to discover candidate parents, each relay node needs to scan its environment periodically, preferably a broad scan covering multiple frequency channels available to the transceivers. If it has a dedicated scanning radio then its normal function of transporting chirps is not interrupted. Else, it must request a scan "lunch break" from its parent, in order to use its radios to scan on other frequencies than the one it is using. At that point, it will need to tell its incoming link from its parent, to "hold all messages". During that period, the clients are effectively temporarily detached, see FIG. **57**.

A parent node would have multiple scan requests, which would be permitted in some weighted, round robin manner, weighted in favor nodes with more clients, for example. Using such a round robin scheme, each sibling node of a patent would be granted a timed lunch break, so that no two siblings scan at the same time, thereby missing each other. The siblings may know of each other but without mutual probe requests, have little knowledge of the signal strength and tested link quality. Further, since the current "mother" parent's siblings (e.g. aunts) are also potential parent candidates, none of them may be in scan mode either. Hence the scan request is being permitted by a parent's parent or grandmother. By the same token, the decision to allow a node to switch is therefore also addressed by at least a grandparent to the requester node.

In general, changes within a sub tree (child moving from mother to aunt) will not affect the grandparent aggregated upstream throughput—since both the aunt and mother are its children. Hence, if the shift request is within parent's siblings, the perturbation is contained and temporal. In general it is at least grandmothers of the intended parent candidate provide the final permissions.

For network topologies with less than a few hops, it is more efficient to let the root node address both switch and scan requests. The root node is generally more processor and memory equipped, since it also handles the IP to chirp interface. As one of many "hubs" for the chirp buses, it is also the logical place for small data subscribers and agents to reside. Some of those agents, Navigational agents, may wish to have a say in the changes in network topology. Thus agents are part of the control plane managing the physical network. Since the physical network and logical network map to each other, the only option is to change the network topology, by moving nodes around, based on the global (root level) toll cost/hop cost criteria. The network topology is thus managed to be in dynamic alignment to subscriber demand.

As in insect colonies, the primary function of every node, all the way from the edge/branch nodes to the root node is identical. Each node wishes to improve its lot, but with a view to long term network stability. This is akin to ant or bee colonies, where the common good affect all positively. Thus a node may be directed by the root node to switch parents because it would streamline the publishing stream. Or nodes may be directed to disassociate a chirp child and have another sibling (aunt) adopt the orphan. Thus each of the sibling nodes may, over time, become specialist hubs for category clusters and the social network coalesces towards more efficient routing. This is akin to trees changing their growth to favor changes in sun direction. Adaptive Network trees, like natural trees are driven by the common good of the entire tree, including all constituents, down to the chirp devices at the edges.

Beacon Heart Beats

The extensible chirp protocol may be used by propagators also, to provide information at various levels of granularity. Within the propagator community there will be specialists who will connect only to other specialist relays. They may limit their relay efforts to specific chirp categories or devices thereby forming a private and exclusive logical chirp network. These specialists may use other nodes to provide their transport and routing, but in effect, the meaning of the data being routed in intra specialist.

In order to support routing requests from the wider community, all propagators collaborate when possible—in service of the larger network. Thus basic routing is part of a common protocol and language, extensions are specialist/navigation agent based.

The basic routing is akin to layer 2 wired switch stacks and their wireless mesh node equivalents, see FIG. **1**. In both cases, tree topology ensures scalable O(n) routing overhead. In each case, there is only one uplink.

The very basic "beacon" information that relays may transmit must minimally include the hop cost, toll cost and parent name. The parent name is needed, because then a prospective child can talk directly to the parent. Recall the grandparent manages scan and switch events. Hence, they know if a better parent is available but is out to lunch, performing a scan. Thus, a propagator node may be left awaiting association permission from a prospective parent node's parent till a scan is over. This delay ensures that connections once made do not have to switch to a better candidate, discovered after some later scans. The grandparent is being proactive.

Thus, the very basic heart beat from all nodes must contain:

1. My name
2. My Parent Name
3. My hop level—from the root node
4. My Toll Cost.

The orphan, during its scanning period after power up, sends and receives probe requests from multiple connected nodes in its vicinity.

It can surmise which candidates are siblings, based on their parent name. Should it join either sibling, it is assured of collaborative alternatives within the same sub tree (the aunts). This engenders its "survival", in terms of redundant paths with minimal changes—the rest of the sub tree back to the root node is unchanged, between sibling switches. Routing updates are only needed at the last hop. In contrast, switching between entire sub trees is more onerous, especially if that sub tree's siblings are not available as back ups. Survival favors joining sub trees with multiple accessible sibling mother/aunts.

Latency and Throughput Tradeoffs

Beacon heart beats enabled orphans to discover potential parent presence. Their relative proximity is measured during probe requests, to determine effective link quality and include that in its selection process. The total available throughput in a string of pearls link is simply that of the weakest link—the link with the worst "performance".

Candidate parents may thus receive pings to test aggregate link quality all the way up to the root node. In general, each node has an inherent predilection to choose the best "lineage" to connect with. But there are tradeoffs. Ideally, all things being equal, nodes would wish to connect as close to root nodes as possible—since M2M traffic is largely upstream. However the link quality to a distant node, may be a lot worse than going through intermediary nodes.

In the simulations above, overall back haul throughput, from all upstream traffic to the root degrades as the tree topology (Caption **1**) is modified by Toll Cost/Hop Cost ratios favoring low hop cost. Notice that Caption **4** is the familiar hub and spoke, single hop topology—with the worst overall throughput.

In addition to overall link quality, ascertains through pings, the availability of the candidate parent to service additional requests drives the final decision. Great overall backhaul throughput is academic if the node is already saturated, based on its limited processing power.

Toll Cost provides the nodes level of availability. Higher toll cost nodes are being selective, mindful of their own limitations and therefore protecting their existing clients for being crowded out. Thus allegiances are formed, where propagators develop preferences to belong to particular sub trees that demonstrate healthy characteristics (e.g. multiple sibling accessibility etc)

Routing Table Updates

Having joined the network, nodes must now relay chirp broadcasts in their vicinity. Nodes would have one uplink, to maintain the tree structure, though multiple uplinks, servicing disparate trees (to avoid cycles) is also permitted. Multiple downlinks can service both chirp and IP traffic on both the same and distinct wireless interfaces. The uplinks, could be either chirp or IP based (e.g. Wi-Fi).

For each disparate uplink, routing tables are maintained that provide layer 2 switch stack functionality. Packets are moved either up the tree to ancestors or down the tree to descendants. The decision is based on a condensed routing table, updated by each node, based on comprehensive routing heart beat sent periodically and circulated within the relevant sub tree.

Thus nodes, operating on different frequencies, send out heart beats received through its uplink to a parents downlink. It then rebroadcasts it, both on its downlinks and uplinks, operating on different frequencies. Nodes now receive updates on their own uplink and downlink channels and disseminate the information to their own sub trees. In effect

Nature's equivalent of allergy season is in effect. It is seasonal, because each heart beat is tagged with a counter number. Since heart beats will travel multiple paths in broadcasts, nodes ensure that the same counter number is not rebroadcast. Further, each node and its agents may decide how far up or down the tree to provide the broadcast. For example, a parent switch to sibling aunt requires no further broadcasts than the last hop routing table.

Eventually, each node is aware of

1. its own immediate children
2. In the case of propagator nodes, their children etc
3. Adjacent nodes that may serve as alternate parents
4. Its current overall current link quality and throughput
5. Through scanning, the overall link quality of alternate parents

Over time, the cost of switching back and forth is reduced by developing more data on both the current parent and its alternatives. This information leads to stable networks at the local levels.

The routing table is available to all members of the current sub tree to (at least) the level of a grandparent. Thus each node is aware of is entire sub tree of descendants below it, at least two hops down. After that, it is somewhat irrelevant, since its grandchildren, on having the packets delivered to them, will know how to relay it further. All the grandparent needs to know is *roughly* where the chirp parent node, resides e.g. which portion of its descendant sub tree—a general direction of routing suffices. In the event chirp devices move around, one or two packets intended for them, will be lost (there is no retry or retransmission in chirp world). But most traffic is upstream. It suffices, for each chirp descendant under its care, nodes need only be aware of:

1. Chirp device Descendant's immediate parent propagator
2. Location of that parent node within its sub tree (e.g. lineage)
3. Hence a lineage path from root node to chirp device exists.

Some chirps will be picked up by multiple propagators. They will all rebroadcast the packets, in the directions specified (up/dwn). However in each case it will tag the packet with the chirp device's immediate propagator, which is the last part of a lineage tree. Multiple chirp packets will thus travel separately up stream through different relay paths, from multiple propagators that pick up the chirps in their vicinity. Multiple lineage paths are available.

Multiple paths is useful when redundancy is desirable. Such is not the case with chirp sensor data and the relative unimportance of any single chirp. Hence pruning of multiple paths is performed at the grandparent level. Chirp packets are relayed through one node only, typically the node closest to it and therefore the best link quality. Others, also picking up the chirps directly from the devices, are directed to ignore those chirps. The chirp device is now assigned a unique lineage or relay path back to the root. Thus, even in the case of unidirectional chirp streams, an association is made, to prune redundant traffic.

A tree based, scalable, hierarchy driven control system emerges, where filters are applied to progressively reduce redundant data upstream. Here is the beginning of small data, as data being sent upstream continues to be more refined as it passes through multiple rule-based logical sieves. The agents may be simple, ant like, but their strength also lies in numbers and their ability to support multiple interpretations of the same data and provide that perspective with significantly lower latency than if everything was pushed up to the cloud. There are situations where raw chirp data needs to send all the way up, but like the Mars Rover, when latency matters, some level of local autonomy is essential to the survival of a network burdened at the edges.

Thus, in this model of IoT, As schools of chirps travel upstream up the river to the root, agents at strategic locations along the chirp route may perform local pruning, aggregation and exception handling, thereby reducing the traffic and improving load performance. Since multiple agents can be operating on the same data, some form of collaborative scheduling and sharing of timing requirements is needed. This will be discussed in a later chapter.

Returning to pruning lets examine the savings in IP traffic for a 100 sensor network. Consider, for simplicity, a 10 node string of pearls chain, each propagator/relay supporting 10 sensors, all the way back up to the root. For example, these sensors could be part of an underground coal mining tunnel, with propagators forming the life line for both IP and chirp traffic.

Simple Rule based logic watches the methane gas occurrence across the tunnel. As you would expect, methane exposure at one region, would also affect adjacent regions and hence a blob of methane gas publishers may appear abruptly and unexpectedly.

Sending such "exception handling" upstream to big data servers is clearly valuable. It is questionable whether routine and acceptable readings would be sent over the cloud. But without some local handling, there is no way of defining an exception, without a base line of routine readings. Hence agents may also maintain some short history.

Implementing Large Scale Chirp Based Control Systems

Discovery of Duplicate Classifications

When a new device joins the network and is mistakenly using a classification scheme already mapped to a different type of device then agents for both the new and incumbent devices will initially receive both new and existing device broadcast based on their subscriptions pattern "match". Thus both agents discover and report the disparity so chirp classifications may be amended upstream. However, in a distributed framework, with local intelligence at the edge, agents at hubs/nodes may also confer internally, and based on differences in their chirp streams, FIGS. **47** through **49**, redefine the subscription bit mask patterns they are interested in, as a sequence/pattern/script of bit mask matching(s). This requires no change to the chirp device transmissions, FIG. **64**. Thus chirp re-classification for blind devices—that cannot change their transmission—is supported at both local/edge and enterprise levels of the logical network comprising of both IP and non-IP devices (e.g. Chirp).

Robust Dynamic Chirp Category Assignment.

Edge networks, by definition, are at the extremes, part of a localized and often partially isolated cluster/region, FIGS. **14** through **15**, **28** and **39**. A temporary chirp based classification/addressing scheme, operating within the local zone is akin to the distributed DHCP servers for isolated clusters see FIGS. **27** through **29**. Recall each node of the network maintains its own DHCP/SIP registry contention free address space based on the distributed DHCP framework. Thus each node is part of a distributed, contention free addressing scheme at each node of the dynamic tree based logical mesh network.

Similarly, chirp classification assignments may use the same distributed DHCP framework to temporarily assign new chirp classifications (akin to DHCP addresses) without confusion. Further, the DHCP approach may also be used to assign chirp classifications or mappings, specific to the application/zone, based on subscriber preferences—hence the agents, working with the distributed DHCP scheme, can confer internally to review renaming options. The distributed DHCP framework FIG. **27**, **28**, **29** also provides a "gasket" between chirp based devices using similar/identical chirp classification and/or protocols conflicting with other existing chirp devices operating in the same local network/zone.

Application Driven Renaming

Agent managed renaming/reclassification reduces confusion when devices are employed in applications where incumbent current chirp classification is misleading. For example, a wearable fitness device with a personal fitness running/tracking classification may be repurposed for an industrial application, where the device is now monitoring vibration on a portable/remote diesel generator. The chirp category may then need to be reclassified in accordance with industrial sensor categories, since agent subscribers are looking for that specific bit mask/pattern. Doing this at the agent level, with support from the distributed DHCP services is simpler than recoding the chirp devices thus supporting non-programmable/blind chirp devices

Changing Device Locations for Unique Routing Lineage

Two devices may be separated so their chirps are received on a distinct interface/port of the wireless switch stack network, FIGS. **1** through **8**. Propagators, may be inserted, FIG. **57** or the router interface rotated/repositioned as shown in the 4 leaf clover design FIG. **63**. Other discrimination patterns e.g. frequency of intervals, physical location et al, may also be used by the chirp aware router to generate a logically distinct classification for the new devices and route the chirp packets without ambiguity and without requiring further modifications upstream. These lineage based discriminators apply to the routing topology of the network. Additionally, the agent based network, operating at the nodes, may redefine/alter subscription bit mask patterns they are now interested in, as a sequence/pattern/script for the forwarding table entries in the routers. Thus, multiple options support the ability to support devices with completely identical chirp classifications.

Simulation Support

In designing the distributed control system, these methods in simulation, with virtual chirp devices, may automatically suggest where the propagators nodes should be positioned, to avoid ambiguous chirp classifications (without changing the agent level bit mask matching). It may also be used to direct some port/radio interfaces to accept only chirp traffic and disassociate with any IP clients on that wireless medium in the event blind devices, chirping on their random/preset patterns are adversely affecting other network traffic on the same interface/port/radio/LED etc. Further, the agents are positioned judiciously along the control node paths, in order to meet the bus schedules and stem the deluge of unsubscribed/uninteresting upstream data. Thus the latency/throughput model FIG. **1**, is applicable to both the physical (nodes) and virtual (agent) tree networks.

Chirp Registry for Discovery:

Ser. No. 12/352,457 expanded on the DHCP approach with the isolated distributed SIP registry functionality as described in that invention. Terse and periodic VOIP chirps are routed by the local SIP server. Further, when the VOIP network joins an external network, SIP registry information is exchanged. The Chirp registries may thus also exchange chirp device classifications and protocol specifics. Other (restricted) registries may contain location of agents dispersed at local nodes based on bus scheduling requirements. This enables discovery using the same approaches as described in Ser. No. 12/352,457 for devices, agents and logistics hubs. A comprehensive framework for discovery and routing for both Chirp and IP devices are thus supported by the DHCP and SIP registry modules, specifically designed for mobile, temporal wireless mesh networks.

Collaborative Scheduling at the Edge.

Blind chirp devices may be low cost, throw away consumer items. Industrial/enterprise networks may prefer the "upgraded" version supporting bi-directional communications. This enables better sharing of the wireless/wire-less mediums (e.g. WiFi, Audio, Sonar, LED). Like polite children, they are taught when to speak and listen, unlike their more unruly blind cousins. The time slices allocated for Chirp transmissions—the silent periods for IP devices—can then be proactively managed (shifted/reduced/fragmented/coalesced) to ensure that latency sensitive IP traffic (e.g. VOIP) is not affected adversely, see FIG. **24** through **25**.

FIGS. **65** through **67** relate to the collaborative scheduling engine. FIG. **65** is the starting condition where each task is taking just a little longer than expected. The usual first in first out queue, see FIG. **66**, results on excessive delays further out for entities at the tail of the queue. FIG. **67** employs collaborative scheduling to balance out and maintain overall "customer satisfaction". "Customer satisfaction" here includes setting up the timing and task intervals of multiple chirp packet deliveries and their aggregated forms.

When chirp and wifi share the same medium, dynamic scheduling of the silent periods (for chirp transmissions) becomes necessary using the NAV, RTS and CTS control signals see FIGS. **34** through **36**. More deterministic chirp transmission elapsed time, combined with some flexibility in chirp transmission initiation intervals provides the Task Duration, Start and Task End time slot reservations for the Collaborative Scheduler. As shown in FIGS. **65** through **67**, stacking is reduced in favor of overall customer satisfaction, see FIG. **67**, as opposed to conventional FIFO or other priority based queuing algorithms, FIG. **66**. Recall that the collaborative scheduler shifts adjacent time reservation slots to the blue (before) or red (after) regions to reduce FIFO effects, FIG. **66**. The mesh control layer, FIGS. **12**, **20** may thus use Collaborative scheduling to manage collaborative sharing of wireless medium in terms of setting up silent period reservations. To do this, it subscribes to the Collaborative scheduling engine, managing the chirp bus schedules. The schedule is then provided to the chirp routers/Access Points communicating with chirp devices. The device chirp schedules are thus coordinated through small blue/red shifts. Jitter and Latency performance objectives are thus proactively managed downstream from the source devices. Access Points sharing the same medium with IP devices then set up periodic NAV directives and RTS/CTS procedures to ensure that simple chirp devices can take control of the medium without IP based techniques e.g. CSMA/CA.

Since the chirp transmission schedules feeds the inter-node chirp bus schedules, the Collaborative Scheduling, FIGS. **65** through **67** may be used also by mesh nodes for the bus schedules for the node-node transport system. This in turn, also drives when chirps need to arrive and is another input to the mesh control layer, FIGS. **12**,**20**. The schedule knows when the buses leave—t may also thus request the devices to chirp more (missing buses) or less frequently (waiting in line). Chirps should occur with the timing constraints of the bus scheduling system so that chirp packets arrive in time to climb aboard the bus in time to meet the inter-bus schedules of the logistics hubs.

Ideally, Chirp transmissions should occur sufficiently frequently for the level of granularity requested by the subscribers/"Customers" schedules. In other words, the bus schedules is driven by the subscriber demanding more frequent updates. But this is dynamic and hence the need to manage smart chirp transmission frequency and blue/red shifts at the source (e.g. sensors). Listen capable Source chirp devices with a lower level MAC layer capability are capable of this proactive response. Note that a lower level MAC can run on low cost microcontrollers and requires far less power than low power IP alternatives (e.g. Bluetooth). Further, the chirps tunes/directives are specific for the device—non standardized communication format/protocol is supported in an software agent based network.

Small blue/red shifts are thus managed by listen capable chirp devices. The Access Point receiving the chirps, monitors them to determine the range of the transmission "task" start/end times and their blue/red regions. It can thus profile the task and its variants. The bus schedule for the subscribers are then examined to determine the optimal bus size and interval. This, in turn drives how often the chirps must be transmitted and at what time. Synchronization may be approximate, measured in terms of a non dimensional parameter e.g. the fraction of the current chirp interval. This ensures complicity since both parties understand what that means regardless of whether they represent the time in nanoseconds or primitive register counters, Thus intelligent chirp devices may be controlled to adapt to the subscriber interests. Recall that chirp agents subscribing to the device chirps are also publishers to upstream bus schedule latency/throughput requirements, See FIGS. **1**, **20**. This is particularly relevant to mission critical remote control systems operating at the edge of the network, in the physical world e.g. the outdoor enterprise (Military and Mining, Emergency response), Industrial Automation (Factory and Warehouse) and Home Automation (Intelligent Energy Usage).

The Larger (logical) Network

The Layer 2 mesh network see FIGS. **4** through **15** and FIGS. **57** through **58** is logical switch stack tree based. Agents may reside anywhere on the network, and each mesh node can support an agent friendly application layer, see FIG. **12**. The application layer, FIG. **12** may also access the VOIP concatenation engine **020** and the packet classification engine **010**, both of which support both chirp and IP protocols, through an hardware abstraction layer, FIG. **20**. The application layer FIG. **12** may need to access distributed DCHP/SIP services, FIGS. **27** through **29**, and localized chirp streams, see FIG. **48** through **49**.

When a new device is registered on the network, its agent/app is "loaded"-. This agent/app then provides the subscriber-based chirp-to-chirp translations and chirp-to-IP translations and filtered "small" data. Based on the direction of the chirp, agents may also direct propagators to provide intermediate translations see FIGS. **22**,**23** through applets operating at the routing and "port" forwarding tables at each mesh node port/interface of the dynamic switch stack tree. Techniques to do this with Software Defined Switches (e.g. OpenFlow) may then be easily adopted to support Chirp Extensions at the edge. A logical SDN network emerges from core/enterprise routing, where Big Data servers reside, all the way down to local edge network clusters where "Small" data is being generated for their subscribers operating at the application layer, see FIG. **12**, **20**.

A software driven MAC (e.g. MAC802.11) supports building flexible, extensible chirp to IP and chirp to chirp bridging and software based switching at any node of the network. Within this layer, agents may subscribe to support services (e.g. DHCP, routing tables, switch level port forwarding tables) that enable efficient bus scheduling. The scheduling supports the jitter and latency constraints of small packet flow over IP networks see FIGS. **10**, **12**, **24**.

Agents register on the Chirp IP membrane, and subscribe to specific chirp/IP streams. Any expressible bit mask pattern or combinations thereof is possible with a software defined routing e.g. managing the forwarding tables based on both simple and complex bit mask patterns/scripts not just on IP based headers. An agent based chirp to IP membrane, operating within a soft MAC layer can thus support the communication with external networks that together support a global publish-subscribe chirp an IP exchange. Each port of the network is thus logically addressable with intermediate agents providing the security and translations. Software driven switches, operating within the factory/enterprise, may thus easily support registered external agents authorized to subscribe to any node or chirp bus logistics hub. The forwarding tables on the switches controlling those ports is then modified accordingly. Thus both chirp and IP traffic can be subscribed to with software defined agents providing the necessary exchange services, all the way down to the switch forwarding tables in the data plane. Note that the collaborative scheduling engine, operating at the management control plane of bus schedules, is also operating all the way down to the routers, within the Mesh MAC layer, FIG. **12**,**20**. Thus, managed latency/throughput sensitive hierarchical control system emerges (FIGS. **56-57** at both the data planes and the control planes, all the way down to the edge of the chirp network.

The network becomes the control system, comprising of listen capable chirp devices, chirp aware meshed routers/Access Points, a shared vocabulary collaborative task scheduler operating through a open and extensible chirp based publish-subscribe exchange with agents logically/physically residing at any control point/node within the mesh network. This scalable distributed control system thus enables that the data flow from publishers stays in alignment to subscriber requirements within deterministic latency/jitter for packet delivery

The invention claimed is:

1. A tree-shaped mesh network comprising:

a mesh of wireless nodes forming a tree shaped network with one root node having a connection to an external network; chirp clients; and wireless network clients;

wherein chirp clients comprise low cost chirp devices wherein said low cost chirp devices transmit short duration messages;

wherein at least one wireless node of the mesh of wireless nodes is a designated chirp-aware node and said chirp-aware node further comprises a bridge between the short duration messages and IP based devices wherein said bridge includes a wireless receiver to receive said short duration messages;

wherein all remaining wireless nodes, consisting of wireless nodes other than said at least one wireless node, disregard the short duration messages as random and transient noise by adaptively filtering out the short duration messages using Automatic Gain Control, Error Correction or noise cancellation, wherein the short duration messages are sufficiently short in length so that said adaptive filtering by all remaining wireless nodes disregards the short duration messages as random and transient noise;

wherein each wireless node further comprises two logical radios and a service radio wherein each wireless node uplink and downlink operates on distinct non-conflicting frequencies; and

wherein said wireless network clients communicate with said wireless nodes using the

said service radios; and

wherein the designated chirp-aware node is at an edge of the tree-shaped mesh network and wherein said chirp-aware node containerizes the short duration messages into containerized packets by adding an IP header to each said short duration message.

2. The tree-shaped mesh network of claim **1** wherein the designated chirp-aware node assigns a destination address to each containerized packet.

* * * * *