US 20100275017A1

(19) United States

(12) Patent Application Publication

Iyer et al.

(10) Pub. No.: US 2010/0275017 A1

(43) Pub. Date: Oct. 28, 2010

(54) **PEER-TO-PEER FORWARDING FOR PACKET-SWITCHED TRAFFIC**

(75) Inventors: **Pradeep J. Iyer**, Cupertino, CA (US); **Keerti G. Melkote**, San Jose, CA (US)

Correspondence Address:
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040 (US)

(73) Assignee: **Aruba Networks, Inc.**, Sunnyvale, CA (US)

(21) Appl. No.: **12/429,981**

(22) Filed: **Apr. 24, 2009**

Publication Classification

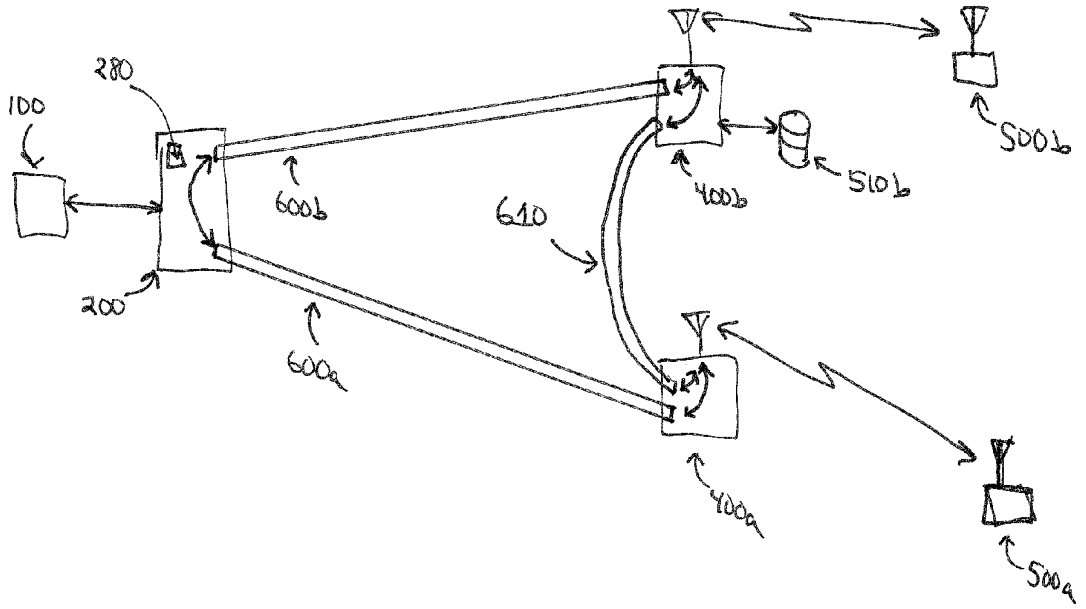
(51) **Int. Cl.**

H04L 29/06	(2006.01)
H04B 7/00	(2006.01)
H04L 12/26	(2006.01)
G06F 15/16	(2006.01)
H04L 12/56	(2006.01)

(52) **U.S. Cl.** **713/162**; 370/310; 370/253; 709/228; 370/400

(57) **ABSTRACT**

Establishing peer-to-peer tunnels between clients in a mobility domain. In normal operation, clients attached to a network having access nodes connected to a central controller transfer all traffic through the central controller. This traffic is passed using tunnels between the access node and the central controller. Tunnels may be encrypted, and GRE tunnels may be used. A mobility manager operating in the controller tracks access nodes connected to the controller, and clients connected to those access nodes. When the mobility controller recognizes traffic passing between clients in its mobility domain that is eligible for peer-to-peer forwarding, it instructs the access nodes supporting the clients to establish a peer-to-peer tunnel between the nodes, and direct the client traffic through this peer-to-peer tunnel. The peer-to-peer tunnel may be session based, or may be aged. Eligibility of traffic for peer-to-peer tunnels may be controlled by rules, such as limiting peer-to-peer tunnels by source or destination, by port or protocol, and the like.



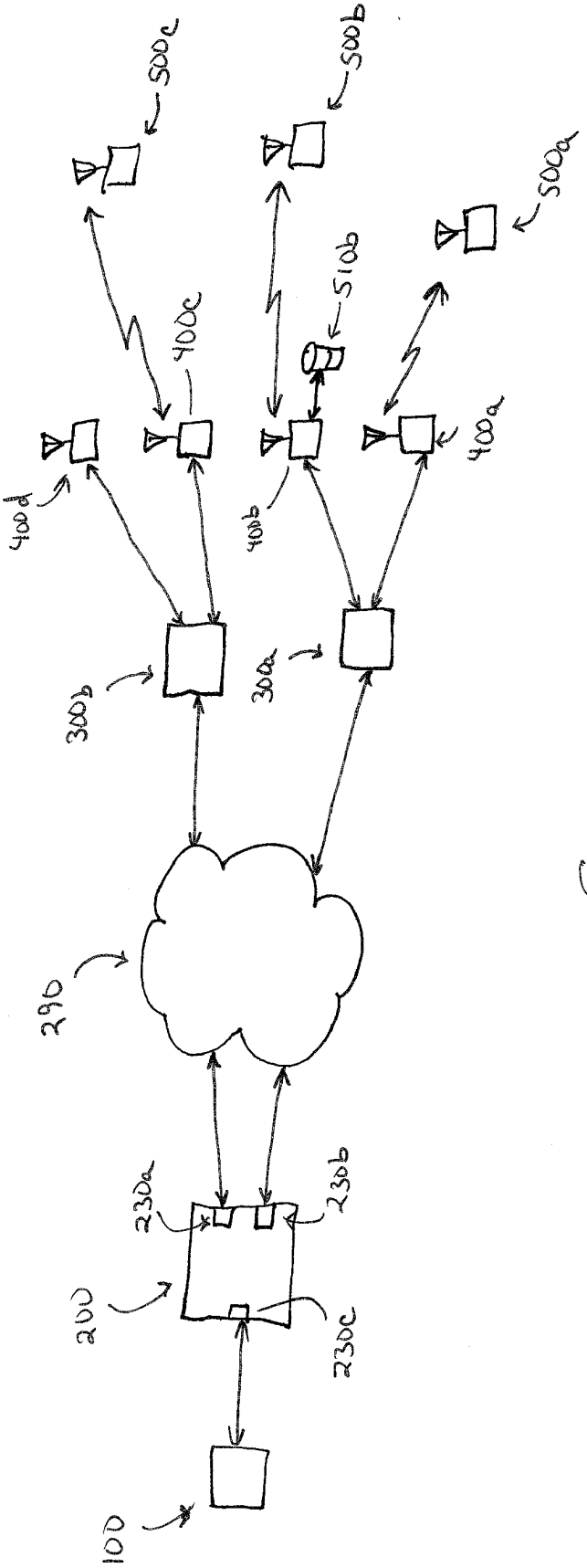
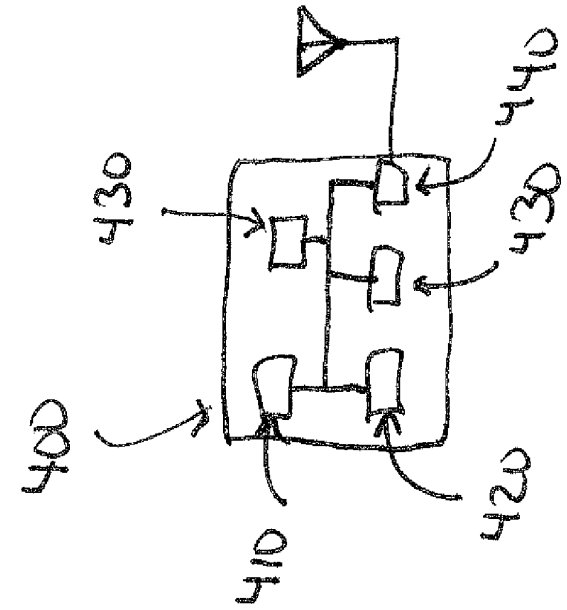

500c
500b
500a
510b
400c
400d
400b
400a
300b
300a
290
230a
230b
200
230c
100

Fig 1



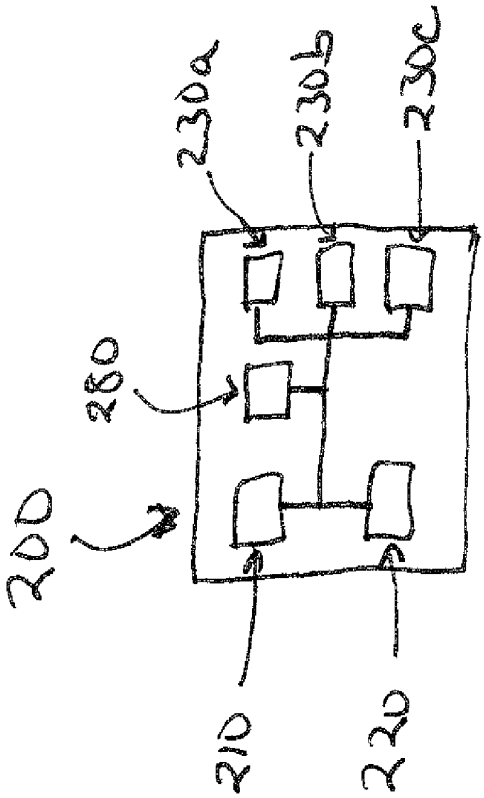


FIG 2

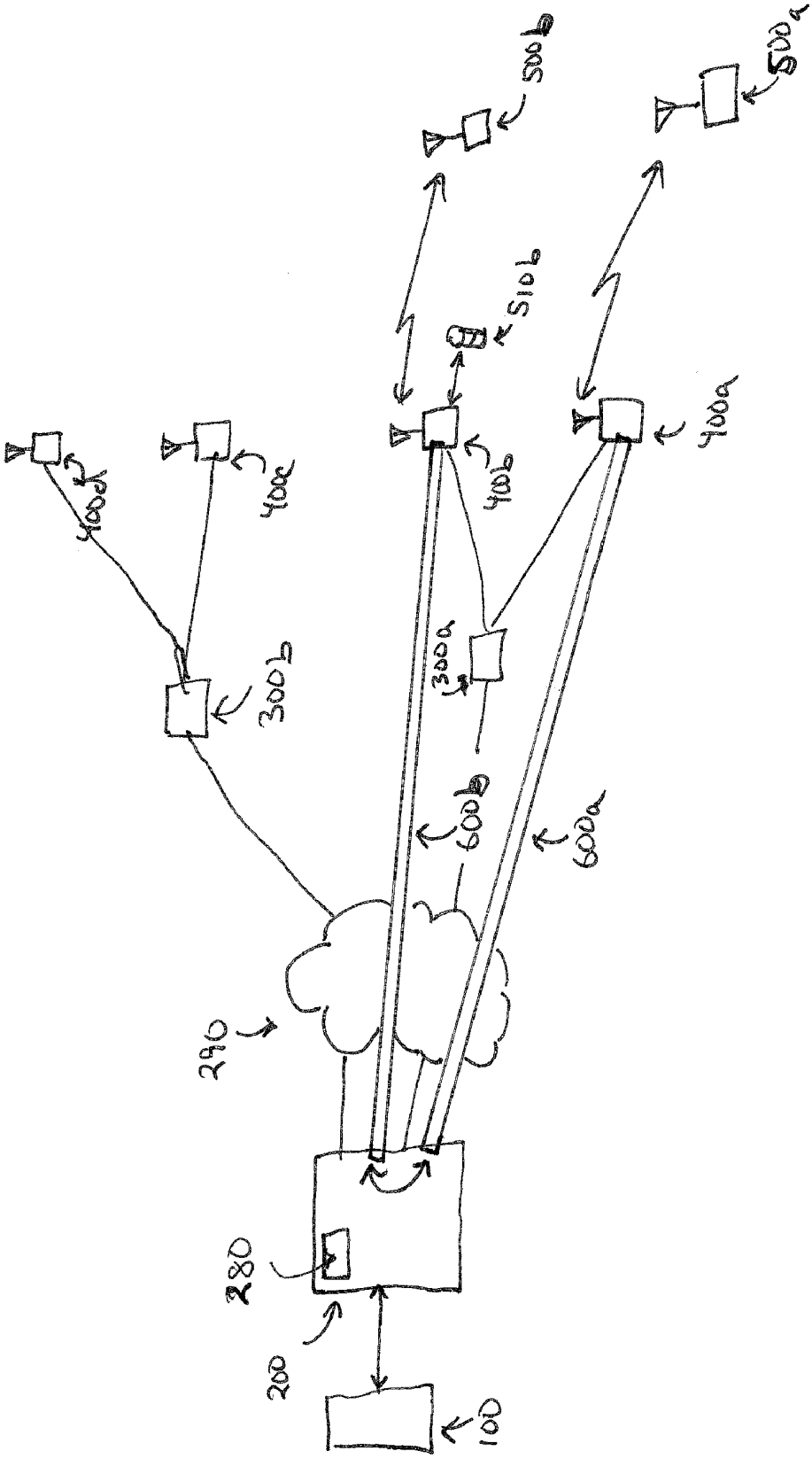

500b
500a
510b
400b
400a
400d
400c
300a
300b
600b
600a
290
280
200
100

Fig 3

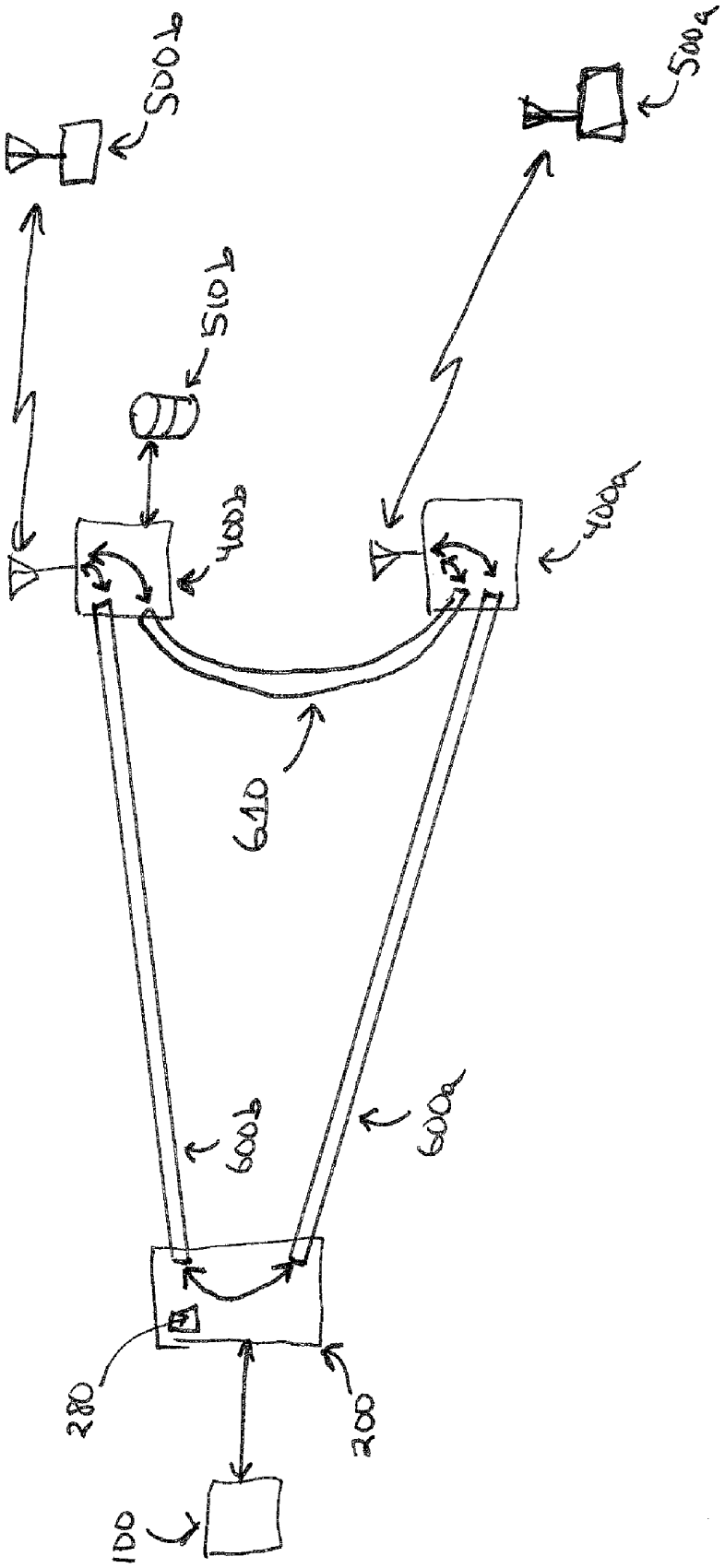


Fig 4

PEER-TO-PEER FORWARDING FOR PACKET-SWITCHED TRAFFIC

BACKGROUND OF THE INVENTION

[0001] The present invention relates to digital networks, and in particular, to the problem of routing traffic in controller-based digital networks.

[0002] Modern digital networks operating under IEEE 803.2 and 802.11 standards are called upon to support a wide range of wired and wireless clients.

[0003] Such systems usually comprise one or more controllers, each controller supporting one or more access nodes which provide wireless and wired network services to clients. In modern wireless systems, such access nodes may be located at some distance from the controller, communicating with the controller through the routed network. In operation, such systems provide secure network connectivity.

[0004] In operation, traffic to and from client systems connected to an access node passes to the central controller, commonly through the use of a tunnel. As understood in the art, IP tunneling is a method of connecting two disjoint Internet Protocol (IP) networks using encapsulation. In such tunneling, every IP packet with addressing information of its source and destination IP networks, is encapsulated within another packet prior to being sent through the intermediate network. Such encapsulation allows traffic between, for example, the controller and an access node, to be routed through the larger switched network. As is known in the art, such tunnels may be encrypted, such as with GRE tunnels, providing additional security. Tunnels are described, for example, in RFC 1701, RFC 1702, RFC 2784, and RFC 2890.

[0005] While the use of tunnels allows remote access nodes to provide seamless access to services, this architecture does impose a price, routing all traffic through the (central) controller. If a remote user is connecting to a corporate server, the necessity of routing all that traffic through the controller does not impose much of a penalty. But if a user at a remote location is trying to send a large file, or open a multimedia stream such as SIP connection, to another remote user located ten feet away, the central controller architecture means that all that traffic must be routed through the central controller, introducing the potential for delays and bottlenecks.

[0006] What is needed is a way to maintain the benefits of the central controller architecture while not restricting local traffic.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The invention may be best understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention in which:

[0008] FIG. **1** shows a network,

[0009] FIG. **2** shows details of network devices,

[0010] FIG. **3** shows tunnels in a network,

[0011] FIG. **4** shows an additional network.

DETAILED DESCRIPTION

[0012] Embodiments of the invention relate to methods tunneling traffic in a digital network. A digital network has one or more central controllers to which a plurality of access nodes are connected. Each access node provides a combination of wireless and/or wired access to resources available through the central controller. The access nodes may be directly connected to the controller, or they may connect to the controller through routed networks such as the corporate Intranet, wider Internet, through private networks such as VPNs, or through a combination of wired and wireless backhaul.

[0013] In operation, the access nodes establish communications with the controller using tunnels. An example of a tunnel is a GRE tunnel. Traffic to and from clients connected to an access node is routed through the tunnel and through the central controller to which the access node is connected.

[0014] In accordance with the invention, a mobility controller process runs in the controller, monitoring traffic to and from clients. The set of access nodes known to the controller and other associated controllers is defined as the mobility domain. When the mobility controller recognizes that traffic is being sent from a first client in the mobility domain to a second client in the mobility domain, the mobility controller evaluates whether the traffic is eligible for peer-to-peer forwarding. If the traffic is eligible for peer-to-peer forwarding, the mobility manager instructs the access node to which the first client is connected to establish a peer-to-peer tunnel to the access node to which the second client is attached, and to direct the traffic through the peer-to-peer tunnel.

[0015] Peer-to-peer tunnels may be established between any two clients in the mobility domain, or may be established on an authenticated basis. Peer-to-peer tunnels may be limited or authenticated on a per-user basis, and may be limited to certain protocols, ports, and/or destinations. Peer-to-peer tunnels may be kept alive on a session basis, or may be aged and terminated automatically if unused for a predetermined period of time.

[0016] FIG. **1** shows a digital network. Router **100** connects to a network, not shown. Router **100** also provides services to controller **200**. Controller **200** has a plurality of ports, **230***a*, **230***b* for supporting devices such as access nodes **400***a*, **400***b*, **400***c*, **400***d*.

[0017] As shown in FIG. **1**, these ports **120***a*, **120***b* connect through switched network **290** to routers **300***a* and **300***b*.

[0018] Access nodes **400***a*, **400***b*, **400***c*, **400***d* provide wireless and possibly wired services to clients. As shown in FIG. **1**, wireless client **500***a* is connected to access node **400***a*. Access node **400***b* supports wireless client **500***b* and wired client **510***b*. Access node supports wireless client **500***c*.

[0019] As shown in FIG. **2**, controllers **200** are a purpose-built digital devices having a CPU **210**, memory hierarchy **220**, and a plurality of network interfaces **230**. CPU **210** may be a MIPS-class processor from companies such as Raza Microelectronics or Cavium Networks, although CPUs from companies such as Intel, AMD, IBM, Freescale, or the like may also be used. Memory hierarchy **220** includes read-only memory for device startup and initialization, high-speed read-write memory such as DRAM for containing programs and data during operation, and bulk memory such as hard disk or compact flash for permanent file storage of programs and data. Network interfaces **230** are typically IEEE 802.3 Ethernet interfaces to copper, although high-speed optical fiber interfaces may also be used. Controller **200** typically operates under the control of purpose-built embedded software, typically running under a Linux operating system, or an operating system for embedded devices such as VXWorks. Controller **200** may have dedicated hardware for encryption, and/or for routing packets between network interfaces **230**.

[0020] Similarly, as understood by the art, access nodes **400***a*, **400***b*, **400***c* and **400***d*, are also purpose-built digital devices. These access nodes include CPU **410**, memory hier-

archy **420**, wired interface **430**, and wireless interface **440**. As with controller **200**, the CPU commonly used for such access nodes is a MIPS-class CPU such as one from Raza Microelectronics or Cavium Networks, although processors from other vendors such as Intel, AMD, Freescale, and IBM may be used. The memory hierarchy comprises read-only storage for device startup and initialization, fast read-write storage such as DRAM for holding operating programs and data, and permanent bulk file storage such as compact flash. Wireless access nodes **300** typically operate under control of purpose-built programs running on an embedded operating system such as Linux or VXWorks. Wireless interface **340** is typically an interface operating to the family of IEEE 802.11 standards including but not limited to 802.11 a, b, g, and/or n. Multiple wired interfaces **430** may be provided, with one wired interface **430***a* being used to connect the access node to its controller, and the other wired interfaces **430***b* used to host wired devices as clients. While wired interfaces such as 802.3 Ethernet may be used, USB may also be used to support printers, mass storage devices, and wireless back-haul links such as 3G or WiMAX modems.

[0021] While FIGS. **1** and **2** depict a wired backhaul connecting access nodes **400** to controller **200**, a combination of wired and wireless backhauls may also be used, for example, using WiMAX, 3G, or other high-speed wireless connections. While a wired connection to a modem such as an ADSL modem or a cable modem may be used, such a modem may also be built into access node **400**.

[0022] Routers **300** are also purpose-built digital devices, and similar to controller **200**, they contain a CPU, memory hierarchy, and a plurality of interfaces. Routers typically run dedicated software devoted to the tasks required. Routers are commercially available from a number of companies such as Cisco-Linksys, Hewlett Packard, D-Link, and others.

[0023] Wireless clients **500** are also digital devices, similarly having CPU **510**, memory hierarchy **520**, wireless interface **530**, and I/O devices **540**. As examples, wireless device **500** may be a general purpose computer such as a laptop, or may be a purpose-built device such as a Wi-Fi phone or a handheld scanner. In a general-purpose computer, CPU **510** may be a processor from companies such as Intel, AMD, Freescale, or the like. In the case of purpose-built devices, Acorn or MIPS class processors may be preferred. Memory hierarchy **520** comprises the similar set of read-only memory for device startup and initialization, fast read-write memory for device operation and holding programs and data during execution, and permanent bulk file storage using devices such as flash, compact flash, and/or hard disks. Additional I/O devices **540** may be present, such as keyboards, displays, speakers, barcode scanners, and the like.

[0024] In operation and as shown in FIG. **3**, access nodes **400***a*, **400***b*, **400***c*, **400***d* establish communications with controller **200**, in the case of FIG. **3**, through routers **300** and switched network **200**. As shown for access nodes **400***a* and **400***b*, tunnels **600***a*, **600***b* such as GRE tunnels are established between the access node and controller **200**. Such tunnels **600***a*, **600***b* may be established on a per-access node basis, or on a per network basis, with one tunnel established for each advertised wireless network (BSSD) or one tunnel established for each wired port on an access node.

[0025] Assume wireless client **500***a* is connected to access node **400***a*, and client **500***b* is connected to access node **400***b*. When client **500***a* establishes a connection to client **500***b*, traffic from client **500***a* passes through access node **400***a*, tunnel **600***a*, to controller **200**. Controller **200** identifies the traffic destination as client **500***b*, and sends the traffic though tunnel **600***b* to access node **400***b* and client **500***b*.

[0026] This routing is performed by controller **200** using the IP addresses of clients **500***a* and **500***b*, as well as the MAC (media access controller) addresses of clients **500***a*, **500***b* and access nodes **400***a* and **400***b*. When client **500***a* wishes to send data to client **500***b*, it in essence forms an IP packet with client **500***b*'s IP address as the destination, and with client **500***a*'s IP address and MAC address as the source. This information is encapsulated and sent to controller **200**.

[0027] Controller **200** keeps tables of all access nodes it controls, and all clients associated with those nodes, including IP and MAC addresses. In this way, when it examines the packet from client **500***a*, it can determine that client **500***b*, the destination, is connected to access node **400***b*, and direct the traffic through tunnel **600***b* to that access point, and the destination device.

[0028] Even if clients **500***a* and **500***b* are sitting in the same office suite, ten meters apart, traffic between them is routed through controller **200**.

[0029] According to the present invention, mobility manager **280** is a process running in controller **200**. By accessing controller **200**'s tables of access nodes and their clients, mobility manager **280** can detect when a client is exchanging data with another client in its mobility domain.

[0030] As shown in FIG. **4**, when mobility manager **280** detects that client **500***a* is communicating with client **500***b*, also in the mobility domain of controller **200**, mobility manager **280** evaluates if this traffic is eligible for peer-to-peer forwarding. If the traffic is eligible for peer-to-peer forwarding, mobility manager **280** instructs access node **400***a* to establish peer-to-peer tunnel **610** between access node **400***a* and access **400***b*, and to route that traffic between clients **500***a* and **500***b* through tunnel **610** rather than through tunnel **600***a*. While the peer-to-peer tunnel is being established, traffic between clients flows through the controller. In this manner traffic between clients **500***a* and **500***b* rather than traveling through tunnels **600***a* and **600***b* and controller **200**, instead travels through tunnel **610** once the tunnel is established.

[0031] A peer-to-peer tunnel may be established any time mobility manager **280** detects connections and data exchanges between clients in its mobility domain. Or, peer-to-peer tunnels may be evaluated and only established on an authenticated basis according to pre-established rules. Peer-to-peer tunnels may be limited by client identity, including but not limited to client IP address, client MAC address, client authentication, and the like, destination identity, port, traffic type, and so on. As an example, assume a high-speed printer is connected as a client to access node **400***a*. Appropriate rules for the establishment of peer-to-peer tunnels for a printer would be limited to ports and protocols needed for printer use for local authorized users, with no access allowed for guests. Similarly, traffic to e-mail servers would not be eligible for peer-to-per forwarding, so that such traffic would always pass through controller **280** and be subject to firewalling, virus detection, deep packet inspection, and the like. As another example, network time protocol traffic on port **123** would be eligible for peer-to-peer forwarding to reduce transit delays for time data.

[0032] It should be understood that which end of the traffic causes the tunnel to be established is immaterial. As an example, consider a user sending queries to a remote database server. It does not matter if the traffic triggering the formation

of a peer-to-peer tunnel is the transmission of a query from the client to the database server, or the transmission of the query result from the database server to the client.

[0033] Peer-to-peer tunnels may be established on a session basis, or may be aged. As an example, for a device such as a high-speed printer, a peer-to-peer tunnel with a timeout of thirty seconds may be appropriate; if no activity passes through the tunnel for that predetermined period of time, the tunnel is discontinued. If bursts of traffic between two clients exceed the time-out period, the peer-to-peer tunnel will be discontinued, but the next traffic between the clients, which will once more be routed through controller **200**, causes the peer-to-peer tunnel to be re-established.

[0034] Assume as an example file/database server **510***b* is connected via a wired connection to access node **400***b*. Peer-to-peer tunnels may be permitted for authorized users of the database for the specific protocols and ports used for database access, with all other traffic routed through controller **200** for filtering, firewalling, and authentication. As an example, while database traffic using port 3306 between server **510***b* and client **500***a* may be routed through a peer-to-peer tunnel **610**, traffic on port 80 between client **500***a* and server **510***b* is still routed initially through controller **200**.

[0035] When multiple controllers **200** are present within a mobility domain, mobility managers **280** operating in each controller may cooperate in supporting peer-to-peer tunneling within the mobility domain. In one embodiment, a mobility manager **280** broadcasts updates of connected clients to other mobility managers in the mobility domain. These updates may be made on a periodic basis, may be event-driven, such as on client connection or disconnection, or on a combination. By providing the ability for a mobility manager to identify clients attached to a different controller that are still within the mobility domain, peer-to-peer forwarding may be extended to cross controller boundaries.

[0036] In another embodiment involving multiple controllers, mobility managers **280** may send queries to other mobility managers within the domain to inquire if a destination is a client of another mobility manager within the mobility domain. It may be useful in some embodiments to apply additional authentication when controller boundaries are crossed. As an example, consider an enterprise network spread over many locations, perhaps over many time zones. While establishing a peer-to-peer tunnel between a streaming media device such as a security webcam and a monitoring station offloads that streaming traffic from passing through the controller, other policies may wish to restrict access to such cameras to only users connected to the controller at the particular site, not allowing access across controller boundaries, or only allowing access across controller boundaries to certain classes of users.

[0037] While the invention has been described in terms of various embodiments, the invention should not be limited to only those embodiments described, but can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is this to be regarded as illustrative rather than limiting.

What is claimed is:

1. A method of operating a packet-switched digital network having at least one controller supporting a plurality of access nodes connected to the controller, each access node capable of supporting clients, comprising:

monitoring, by a mobility manager in the controller, traffic flowing to and from the access nodes connected to the controller,

the mobility manager deciding if traffic flowing through the controller from a first client connected to a first access node to a second client connected to a second access node is eligible for peer-to-peer forwarding, and

if the traffic is eligible for peer-to-peer forwarding, the mobility manager commanding the first access node to open a tunnel from the first access node to the second access node and to route the traffic between the first client and the second client through the tunnel.

2. The method of claim **1** where the connection between the first client and the first access node is a wireless connection.

3. The method of claim **1** where the connection between the first client and the first access node is a wired connection.

4. The method of claim **1** where the connection between the second client and the second access node is a wireless connection.

5. The method of claim **1** where the connection between the second client and the second access node is a wired connection.

6. The method of claim **1** where the tunnel from the first access node to the second access node is encrypted.

7. The method of claim **1** where the tunnel from the first access node to the second access node is a GRE tunnel.

8. The method of claim **1** where the tunnel from the first access node to the second access node is aged.

9. The method of claim **1** where the connection between the controller and the access nodes connected to the controller uses tunnels.

10. The method of claim **9** where the tunnels are GRE tunnels.

11. The method of claim **1** where the step of deciding if traffic flowing through the controller from a first client connected to a first access node to a second client connected to a second access node is eligible for peer-to-peer forwarding is based on authenticating one or both clients.

12. The method of claim **1** where the step of deciding if traffic flowing through the controller from a first client connected to a first access node to a second client connected to a second access node is eligible for peer-to-peer forwarding is based on the protocol and/or ports in use to transfer traffic between the clients.

13. The method of claim **1** where the step of deciding if traffic flowing through the controller from a first client connected to a first access node to a second client connected to a second access node is eligible for peer-to-peer forwarding is based on the IP address of one or both clients.

14. The method of claim **1** where the step of deciding if traffic flowing through the controller from a first client connected to a first access node to a second client connected to a second access node is eligible for peer-to-peer forwarding is based on the MAC addresses of one or both clients.

15. A method of operating a packet-switched digital network having a plurality of controllers forming a mobility domain, each controller supporting a plurality of access nodes connected to the controller, each access node capable of supporting clients, comprising:

monitoring, by a mobility manager in each controller, traffic flowing to and from the access nodes connected to the controller

the mobility manager detecting traffic from a first client connected to a first access node, the traffic flowing through the controller,

the mobility manager identifying the destination of the traffic from the first client is a client connected to a second access node connected to a second controller in the mobility domain,

the mobility manager deciding if traffic flowing from the first client to the second client is eligible for peer-to-peer forwarding and

if the traffic is eligible for peer-to-peer forwarding, the mobility manager commanding the first access node to open a tunnel from the first access node to the second access node and to route the traffic between the first client and the second client through the tunnel.

16. The method of claim **15** where identifying that the destination of the traffic is a client in the mobility domain comprises:

mobility managers in each controller publishing information on connected clients to other mobility managers in the mobility domain.

17. The method of claim **15** where identifying that the destination of the traffic is a client in the mobility domain comprises:

the mobility manager in the first controller sending queries to the other mobility managers in the mobility domain to determine if the destination is a client of a controller in the mobility domain.

18. The method of claim **15** where the step of deciding if traffic flowing from the first client to the second client is eligible for peer-to-peer forwarding is based on authenticating one or both clients.

19. The method of claim **15** where the step of deciding if traffic flowing from the first client to the second client is eligible for peer-to-peer forwarding is based on the protocol and/or ports in use to transfer traffic between the clients.

20. The method of claim **15** where the step of deciding if traffic flowing from the first client to the second client is eligible for peer-to-peer forwarding is based on the IP address of one or both clients.

21. The method of claim **15** where the step of deciding if traffic flowing from the first client to the second client is eligible for peer-to-peer forwarding is based on the MAC addresses of one or both clients.

* * * * *