FIGURE 1A
before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))
IMPROVEMENTS IN WIRELESS NETWORKS

TECHNICAL FIELD

[0001] The present invention relates to improvements in wireless networks. In particular, the present invention relates to a method or server for encapsulating data, a method or routing module for determining a fastest route within a wireless network, and a method or server for providing additional HTML data.

BACKGROUND

[0002] Within many towns and cities, mobile communication devices may be connected to wireless networks through Wi-Fi hotspots using IEEE 802.11 communication protocols. This requires the user to access a specific Wi-Fi network and register the device with that network.

[0003] However, each Wi-Fi hotspot may be an individual network such that a user of a mobile communication device is required to register individually with each hotspot when they are on the move within the town or city.

[0004] Wireless mesh networks are provided that interlink multiple access points in a wireless network to enable mobile communication devices to connect to the wireless mesh network at any access point and to have continued access to the network when the device is moved around the network. These wireless mesh networks may use IEEE 802.11 communication protocols as well as other suitable IEEE protocols.

[0005] However, methods of controlling how requests and data are routed through the mesh network back to the backhaul of the network have problems associated with them in that the route determined for communicating the request and data is not always the fastest or most efficient route to use. This therefore reduces the efficiency of the whole
[0006] Further, current operation of wireless networks in transmitting data to mobile communication devices connected to the network is not efficient. Typically, every time a mobile communication device requests data such as a particular website, an individual packet is sent from the server back to the mobile communication device. This therefore results in a separate packet for each separate request.

[0007] Methods have been suggested whereby data may be compressed in order to reduce the amount of data being transmitted. However, this requires that additional processing is performed both at the server level in order to compress the data prior to transmission and also at the mobile communication device level in order to access the compressed data.

[0008] Further, it is also known to use cache capabilities in order to reduce the amount of data that is transmitted over the network. For example, mobile communication devices may store previously accessed data on the device in order to reduce the amount of data that is transmitted from the server. However, this only solves the problem of excess data on the network if the requested website has not changed since last accessed by the mobile communication device.

[0009] In order to provide free access to users of a wireless network, advertising may be incorporated into the requested data in order to generate revenue. For example, systems are provided that modify webpage data associated with a requested website in order to incorporate an advert within the webpage when it is displayed on the mobile communication device. This can lead to problems with regards to infringement of rights as modification and changing of webpage content may not be acceptable, allowable or legal. Further, these known systems do not take into account that the inclusion of an advert with a webpage may affect how the webpage is displayed. In a worst-case scenario, the
inclusion of an advert with a webpage may corrupt the webpage such that
the user is unable to access the requested data.

[00010] An object of the present invention is to provide an improved
route determination system and method in order to transmit data and
requests within a wireless network.

[00011] A further object of the present invention is to provide an
improved data encapsulation system and method in order to reduce the
load on the wireless network.

[00012] A further object of the present invention is to provide an
improved additional HTML display system and method to avoid corrupting
existing webpages.

[00013] Each object is to be read disjunctively with the object of at
least providing the public with a useful choice.

[00014] The present invention aims to overcome, or at least alleviate,
some or all of the afore-mentioned problems.

[00015] Further objects and advantages of the invention will be
brought out in the following portions of the specification, wherein the
detailed description is for the purpose of fully disclosing the preferred
embodiment of the invention without placing limitations thereon.

[00016] The background discussion (including any potential prior art)
is not to be taken as an admission of the common general knowledge in
the art in any country. Any references discussed state the assertions of
the author of those references and not the assertions of the applicant of
this application. As such, the applicant reserves the right to challenge the
accuracy and relevance of the references discussed.

SUMMARY
It is acknowledged that the terms "comprise", "comprises" and "comprising" may, under varying jurisdictions, be attributed with either an exclusive or an inclusive meaning. For the purpose of this specification, and unless otherwise noted, these terms are intended to have an inclusive meaning - i.e. they will be taken to mean an inclusion of the listed components that the use directly references, but optionally also the inclusion of other non-specified components or elements. It will be understood that this intended meaning also similarly applies to the terms mentioned when used to define steps in a method or process.

According to one aspect, the present invention provides a method of encapsulating data associated with uniform resource locator requests at a server for transmission over a Wi-Fi network, the method comprising the steps of the server creating a single page table by including in the single page table data associated with multiple uniform resource locator requests received at the server, wherein encoding of the page table into a datagram for transmission over the Wi-Fi network is delayed until a predetermined condition associated with the single page table occurs.

According to a further aspect, the present invention provides a method of encapsulating data associated with uniform resource locator requests at a server for transmission over a Wi-Fi network, the method comprising the steps of the server: receiving multiple uniform resource locator requests sent by multiple mobile communication devices connected to the Wi-Fi network, wherein the multiple uniform resource locator requests include a mobile communication device identification address uniquely identifying one of the mobile communication devices and a uniform resource locator (URL) identifying an Internet resource requested by one of the mobile communication devices; retrieving a first set of web page data associated with an Internet resource identified in a first uniform resource request received from a first mobile communication device; mapping, in a temporary storage medium, a first mobile
communication device identification address associated with the first
mobile communication device with the first set of web page data; retrieving
one or more further sets of web page data associated with one or more
Internet resources identified in one or more further uniform resource
locator requests received from one or more further mobile communication
devices; mapping, in the temporary storage medium, one or more further
mobile communication device identification addresses associated with one
or more further mobile communication devices with the one or more
further sets of web page data associated with further URLs; wherein the
mapping of the one or more further mobile communication device
identification addresses with the one or more further sets of web page
data continues until a predetermined condition associated with the
temporary storage medium occurs, whereupon said predetermined
condition occurring, the method further comprising the step of the server
enabling the encapsulation and encoding of identification addresses and
web page data mapped in the temporary storage medium into a datagram
for transmission over the Wi-Fi network.

[00020] According to yet a further aspect, the present invention
provides a server for encapsulating data in a Wi-Fi network, the server
comprising a URL request receiving module adapted to receive multiple
uniform resource locator requests, and a processing module adapted to (i)
create a single page table based on data in the received multiple uniform
resource locator requests and (ii) delay encoding of the single page table
into a datagram for transmission over the Wi-Fi network until a
predetermined condition associated with the single page table occurs.

[00021] According to yet a further aspect, the present invention
provides a method of determining the fastest route between a current
access point in a Wi-Fi network and an end point in the Wi-Fi network via
a plurality of further access points, the method comprising the steps of
determining ping rates and traceroute distances between the access
points, calculating a route determination value based on a combination of
the determined ping rates and traceroute distances, and determining the fastest route based on the calculated route determination value.

[00022] According to yet a further aspect, the present invention provides a routing module for use in a server within a Wi-Fi network, the routing module comprising a route determination module, a ping rate module, a trace route module and a memory module, the routing module configured to determine the fastest route between a current access point in the Wi-Fi network and an end point in the Wi-Fi network via a plurality of further access points, wherein the ping rate module is arranged to determine ping rates between the access points, the traceroute module is arranged to determine traceroute distances between the access points and the route determination module is arranged to (i) calculate a route determination value based on a combination of the determined ping rates and traceroute distances and (ii) determine the fastest route based on the calculated route determination value.

[00023] According to yet a further aspect, the present invention provides a method of providing additional HTML data alongside HTML data associated with an Internet resource request for transmission over a network, the method comprising the steps of: receiving an Internet resource request, retrieving the HTML data associated with the Internet resource request, executing, in a virtual environment, the retrieved HTML data to determine the original display dimensions associated with the retrieved HTML data, determining, within the virtual environment, modified display dimensions associated with a modified display which has additional HTML data displayed alongside the retrieved HTML data, determining an error rate based on the original display dimensions and modified display dimensions, and determining whether to provide the additional HTML data alongside the retrieved HTML data based upon the error rate.

[00024] According to yet a further aspect, the present invention
provides a server arranged to provide additional HTML data alongside HTML data associated with an Internet resource request for transmission over a network, the server comprising an Internet resource request receiving module arranged to receive an Internet resource request, an HTML data retrieval module arranged to retrieve the HTML data associated with the Internet resource request, a processing module arranged to (i) execute, in a virtual environment, the retrieved HTML data to determine the original display dimensions associated with the retrieved HTML data, (ii) determine, within the virtual environment, modified display dimensions associated with a modified display which has additional HTML data displayed alongside the retrieved HTML data, (iii) determine an error rate based on the original display dimensions and modified display dimensions, and (iv) determine whether to provide the additional HTML data alongside the retrieved HTML data based upon the error rate.

15 BRIEF DESCRIPTION OF THE DRAWINGS

[00025] Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

[00026] Figure 1A shows a wireless network according to an embodiment of the present invention;

[00027] Figures 1B and 1C show conceptual network diagrams according to embodiments of the present invention;

[00028] Figure 2 shows a server according to an embodiment of the present invention;

[00029] Figure 3 shows a processing module according to an embodiment of the present invention;

[00030] Figure 4 shows a routing module according to an
embodiment of the present invention;

[00031] Figure 5 shows a wireless network configuration according to an embodiment of the present invention;

[00032] Figure 6 shows a proxy server according to an embodiment of the present invention;

[00033] Figure 7 shows a GPU according to an embodiment of the present invention;

[00034] Figures 8A and 8B show smart packet header and data sections according to an embodiment of the present invention.

DESCRIPTION OF EMBODIMENTS

[00035] It will be understood that the system herein described may include one or more elements that are arranged to perform various functions and methods. The following portion of the description is aimed at providing the reader with an example of a conceptual view of how various modules and/or engines that make up the elements of the system may be interconnected to enable the functions to be implemented. Further, the following portion of the description explains in system related detail how the steps of the herein described method may be performed. The conceptual diagrams are provided to indicate to the reader how the various data elements are processed at different stages by the various different modules and/or engines.

[00036] It will be understood that the arrangement and construction of the modules or engines may be adapted accordingly depending on system and user requirements so that various functions may be performed by different modules or engines to those described herein, and that certain modules or engines may be combined into single modules or engines or the functions of the herein described the modules or engines may be
separated out into different modules or engines.

[00037] It will be understood that the modules and/or engines described may be implemented and provided with instructions using any suitable form of technology. For example, the modules or engines may be implemented or created using any suitable software code written in any suitable language, where the code is then compiled to produce an executable program that may be run on any suitable computing system. Alternatively, or in conjunction with the executable program, the modules or engines may be implemented using any suitable mixture of hardware, firmware and software. For example, portions of the modules may be implemented using an application specific integrated circuit (ASIC), a system-on-a-chip (SoC), field programmable gate arrays (FPGA) or any other suitable adaptable or programmable processing device.

[00038] Any processing modules described are arranged to perform the steps of a program stored as program instructions within a memory device. The program instructions enable the various methods of performing the invention as described herein to be performed. The program instructions may be developed or implemented using any suitable software programming language and toolkit, such as, for example, a C-based language. Further, the program instructions may be stored in any suitable manner such that they can be transferred to the memory device or read by the processor, such as, for example, being stored on a computer readable medium. The computer readable medium may be any suitable medium, such as, for example, solid state memory, magnetic tape, a compact disc (CD-ROM or CD-R/W), memory card, flash memory, optical disc, magnetic disc or any other suitable computer readable medium.

First Embodiment

[00039] Referring to figure 1A, a conceptual wireless network is shown according to this first embodiment.
A mobile communication device 101 is operated by a user in an environment that has a wireless network as described herein. The mobile communication device may be any suitable communication device that is able to operate using wireless communication protocols. For example, according to this embodiment, the wireless communication device operates using the IEEE 802.11 communication protocols. It will be understood that the wireless communication device and the network may be modified to operate using other similar or suitable communication protocols.

The mobile communication device 101 may be any suitable device such as a smartphone, mobile telephone, laptop, tablet, personal digital assistant or indeed any other communication device that may be carried and connected to a wireless network.

The wireless network consists of a number of access points (105A, 105B, 105C, 105D, 105E). A backhaul 107 is a final access point that connects this portion of the network to a proxy server 109 which is in communication with the Internet (World Wide Web) 111.

A user of the mobile communication device 101 connects to the wireless network by accessing the browser on the device. If the mobile communication device has been configured to detect wireless networks automatically, a window will appear on the device to identify the wireless network to the user. The user is then able to select the wireless network on the mobile communication device.

Upon selecting the wireless network, the mobile communication device 101 sets up a communication channel with the nearest access point (e.g. 105A) using the 802.11 communication protocols. The MAC address of the mobile communication device is transmitted to the access point and the network assigns a virtual address to the mobile communication device that uniquely identifies the mobile
communication device in the wireless network. A table of virtual addresses
recorded against all MAC addresses of all mobile communication devices
is stored at a proxy server on the network.

[00045] It will be understood, that as an alternative, the system may
utilise IP addresses (or any other unique addressing system) instead of
MAC addresses in order to assign a virtual address to a mobile
communication device.

[00046] The user of the mobile communication device may be
required to accept terms and conditions for accessing the wireless network
prior to the network registering the mobile communication device on the
network.

[00047] After the device has been registered on the network, the
user is able to access the Internet using the browser on their mobile
communication device in order to request Internet resources using URL
(uniform resource locator) requests. For example, when the user requests
a particular webpage to be displayed on the browser, e.g. either by typing
in a URL or selecting a hyperlink in the browser, the URL request is sent
from the user's device via the connected access point (e.g. 105A) through
the network via the other access points back to the proxy server 109. The
proxy server then retrieves the Internet resource and sends it back to the
mobile communication device via the network of access points to the
specific access point the device is connected to.

[00048] Figure 1B shows a conceptual diagram of the network.
According to this network topology, the proxy server is called an Access
Point (AP) master. Access Point (AP) primary slaves are able to connect
to the AP master utilising a number of different mechanisms.

[00049] According to one example, an AP primary slave (803A,
803B) may connect to an AP master 801 via a direct Internet connection
between the AP primary slave and the Internet. That is, the AP primary
slave may connect to the Internet via an Ethernet interface 807A, for example. Alternatively, the AP primary slave may connect to the Internet via a 3G or 4G data interface 809B. Regardless of which Internet connection is used, the AP primary slave may then set up a virtual private network (VPN) tunnel via a tunnel interface (813A, 813B) and the Internet connection back to the tunnel server 805 in the AP master. This tunnel connection may then be used to communicate all user traffic between the AP master and the AP primary slave, which then communicates with the mobile communication device 819A connected to the access point (811A, 811B) of the AP primary slave via Wi-Fi communication protocols.

[00050] Both the AP primary slave and AP master include 802.11s mesh interface modules (821A, 821B). Further, a number of other AP slaves (803C to 803G) also include 802.11s mesh interface modules (821C to 821G). These AP slaves are not termed primary slaves because they are not directly connected to the AP master. The mesh interface modules enable all the access points (811C to 811G) to create a mesh of interconnected access points depending on the relative proximity between each of the access points. Therefore, if an AP slave does not have a direct connection to the Internet, it utilises the 802.11s mesh interface module to create a connection between itself and a neighbouring access point.

[00051] The AP slave connects to all 802.11s interface modules that are in range. If one of those 802.11s interface modules is part of the AP master, then this connection is used to transfer data between the AP master and the AP slave using the 802.11s interface protocols. If the AP master is not within range, the AP slave uses connections to neighbouring AP slaves (preferably the AP primary slave) to determine a route back to the AP master utilising the herein described route determination techniques.

[00052] An AP slave (whether primary or not) may be configured to
either enable or disable other AP slaves (whether primary or not) to connect to it. In a configuration where an AP slave is configured to enable other AP slaves to connect to it (via the 802.11s interface), a neighbouring AP slave will only attempt to connect to that enabled AP slave upon detecting a 900MHz radio signal, according to this embodiment. The 900MHz radio signal is the signal used to provide the radio interface network structure via the 802.11s interfaces.

[00053] According to one example, an AP slave has a prioritised list of servers that it will attempt to connect to when it is first powered up. Initially, the AP slave will attempt to connect to the Internet through an Ethernet connection if available. If available, the AP slave can connect direct to the AP master via the VPN connection making the AP slave a primary slave.

[00054] If not, the AP slave will try and connect to a neighbouring access point utilising the 802.11s interface protocols, where the preference would be to connect directly to the AP master if in range (making the AP slave a primary slave), and if not another AP slave (preferable a primary slave). If no 802.11s connections are available, the AP slave will attempt to connect to the Internet via a 3G or 4G radio connection. If at any point one of the connections is lost, an algorithm is executed in the access point to run through the prioritised list in an attempt to reconnect.

[00055] Therefore, according to one example, a mobile communication device may connect to an AP primary slave via Wi-Fi communication protocols and requests data, such as a web page. The AP primary slave is connected to the AP master using any of the methods described herein and passes on the request to the AP master, for example via the Internet VPN connection. The AP master retrieves the web page data from the Internet and sends the data back to the AP primary slave via the Internet VPN connection. The AP primary slave transmits the web
page data to the mobile communication device using Wi-Fi protocols.

[00056] In a further example, if the AP slave that the mobile communication device is connected to is not directly connected to the AP master via a VPN connection, the web page data requested by the mobile communication device may be transmitted from the connected AP slave to another AP slave via the 802.11s interface modules until it reaches the AP primary slave, whereupon the request is forwarded to the AP master. The AP master retrieves the web page data and transmits it back to the AP slave via the AP primary slave. The AP slave then transmits the web page data to the mobile communication device using Wi-Fi protocols.

[00057] It will be understood that there may be more than one AP primary slave connected directly to the AP master using any of the herein described methods, such as an Ethernet VPN connection, 3G/4G VPN connection or 802.11s connection. Further, it will be understood that there may be more than one AP master providing a portal to the Internet for user data traffic.

[00058] Figure 1C shows a further example of a network configuration that may be employed. According to this example, an AP primary slave (803A) may connect to an AP master 801 via Ethernet interfaces (807A, 807B) to provide a layer 3 Internet connection between the AP master and AP primary slave. Alternatively, an AP primary slave (803B) may connect to the AP master 801 via a 3G/4G interface 809B. Using a tunnel interface (813A to 813G) in the AP slaves (whether primary or not), a compressed VPN tunnel connection is established with the tunnel server 805 in the AP master 801 via a layer 2 bridge.

[00059] The layer 2 bridge is provided between each access point (811A to 811G) in each AP slave (whether a primary slave or not) and its tunnel interface (813A to 813G). A backbone interface (817A, 817B, 817F) is provided to enable one or more AP slaves (803A, 803B, 803F) to
provide connections to a further AP slave (803C, 803D, 803G). The backbone interface connects to a leach interface (815C, 815D, 815F, 815G) within the AP slave (803C, 803D, 803F, 803G) to form an 802.11 a/b/g/n connection. It will be understood that the backbone interface and leach interface are effectively the same form of interface and that the terminology identifies what purpose the interface is being used for.

[00060] A layer 3 route or gateway is provided between each backbone interface and its associated leach interface, 3g/4g interface or Ethernet interface depending on which interface is being used for the VPN tunnel connection.

[00061] Interfaces are identified as being in one of two groups: service interfaces and structure interfaces. A service interface is one that provides the end user service to the mobile communication devices, i.e. the internet connection. A structure interface provides backbone connections and leach connections to create the mesh of access points. A 900MHz radio falls into the structure group. Whereas, backbone or leach connections will only be established on radios designated as structure interfaces. An interface can be specified as both a service and a structure interface if it is a mobile communication device that has the capability to operate at 2.4GHz and 5GHz.

[00062] Figure 2 shows a conceptual system diagram of the proxy server 109.

[00063] The proxy server 109 receives the URL request 201 (as well as other URL requests from the same device or other devices) at a URL request receiving module 203. The URL request receiving module 203 includes a buffer 205 that stores all URL requests being received at the proxy server that have been sent by all mobile communication devices connected to the wireless network. These URL requests are stored in the buffer in order of receipt at the proxy server and then provided to the rest...
of the system in order of receipt.

[00064] A first URL request 207 is retrieved from the buffer 205 and sent to a GPU (Global Processing Unit) 209 and an HTML data retrieval module 211. The HTML data retrieval module 211 then determines whether the URL has previously been requested and whether the associated webpage data is stored within a cache 213. If the data is not stored within the cache 213 or it has been determined that the data stored within the cache is not up to date, the HTML data retrieval module 211 retrieves the associated webpage data from the Internet 111.

[00065] The webpage data 210 (i.e. payload data) is provided to the GPU 209. The GPU 209 encapsulates the payload data according to the method described below. After encapsulation, the encapsulated data is forwarded to a data packet encoding module 215, which encodes the data into a smart packet 217 for communication via one or more access points in the wireless network back to the mobile communication device 101. Upon receiving the payload data in a communication via the wireless network, the mobile communication device displays the webpage data on its browser.

[00066] According to this embodiment, effectively a single GPU is controlled by a single central processing unit (CPU) to generate the encapsulated data packets. However, it will be understood that, as an alternative, multiple GPUs controlled by separated CPUs may be used in certain situations. For example, in order to improve throughput of data a first GPU may be used (as controlled by a first CPU) to handle UDP (User Datagram Protocol) traffic or indeed any other one way blind transmission of data. Whereas a second GPU may be used (as controlled by a second CPU) to handle other standard traffic. In this way, parallel systems are provided to generate multiple data packets to ensure that a bottleneck does not occur due to certain data requests by users. Each of the outputs from the GPUs may then be fed to a common RAM, which outputs the
encoded packets for transmission from an antenna array.

[00067] The GPU module 209 is shown in more detail in figure 3. The URL request 207 is retrieved from the buffer 205 by a URL request retrieving module 301. An encapsulation module 303 refers to the URL request and identifies incoming payload data 210 being retrieved by a payload data retrieval module 305. That is, the payload data retrieval module 305 retrieves data from the HTML data retrieval module in the proxy server.

[00068] The encapsulation module 303 generates a page table using an address identifying the requesting mobile communication device. The identifying address is determined from data within the URL request. According to this embodiment the identifying address is determined by cross-referencing the MAC address of the mobile communication device as retrieved from the URL request with the virtual address that was assigned to the mobile communication device upon the mobile communication device registering with the wireless network. Therefore, the virtual address assigned to the mobile communication device becomes the identifying address. This identifying address is essentially an IP address as generated by a DHCP server.

[00069] Alternatively, it will be understood that the identifying address may be the MAC address of the requesting mobile communication device, or indeed any other form of data that may be used to uniquely identify the mobile communication device.

[00070] The payload data associated with the URL request is also inserted into the page table.

[00071] The encapsulation module 303 continues to generate the single page table using further URL requests (including further virtual addresses of further mobile communication devices) along with further associated payload data as requested by the further mobile
communication devices. That is, a single page table is dynamically or continually generated or created to include webpage data that is associated with multiple uniform resource locator requests as they are being sent by multiple mobile communication devices and received at the proxy server.

[00072] The encapsulation module 303 adds a first virtual address and the associated payload data to a temporary storage module, which according to this embodiment is a translation lookaside buffer (TLB) 307.

[00073] The encapsulation module 303 continually adds further virtual addresses associated with URL requests sent by other mobile communication devices along with payload data associated with those URL requests to the TLB until a predetermined condition occurs.

[00074] According to this embodiment the predetermined condition is the expiry of a predetermined time limit that has been set for storing the data in the temporary page table. The encapsulation module 303 determines whether the predetermined time limit has expired by communicating with a timer module 309. For example, the predetermined time limit may be set to an arbitrary value of one second. It will be understood that any other predetermined time limit value may be set.

[00075] As an alternative, the predetermined condition may occur when the amount of temporary data being stored in the temporary storage module has reached a predetermined size limit. That is, the temporary page table being generated has reached a predetermined size limit and so has reached a maximum defined storage capacity.

[00076] Further, it will be understood that other predetermined conditions may be used to trigger when the temporary storage module data is to be used to create a single page table. For example, the predetermined condition may be the detection that a predetermined number of URL requests have been made.
Upon the predetermined condition occurring, the encapsulation module instructs the TLB to allow a page table generating module 311 to access the temporary data. Upon the page table generating module 311 accessing the temporary page table data, the page table generating module generates a single page table 313. The generated single page table maps multiple virtual addresses of multiple mobile communication devices to associated sets of requested webpage data. For example, the webpage data may include markup language data, such as HTML.

The single page table therefore includes a base address along with multiple virtual addresses for each of the mobile communication devices that have made URL requests. Further, the page table also includes an indication that a cache is enabled to hold information on request.

After the single page table 313 has been generated by the page table generating module 311 it is provided to the data packet encoding module 215 for encoding into a smart packet (datagram). The data packet encoding module 215 within the proxy server generates a HTTP smart packet in the form of an IP datagram delivered to the mobile user using standard communication protocols. That is, the datagram includes multiple mobile communication device identifying addresses associated with multiple mobile communication devices stored within the TEXT space of the header portion of the datagram. The TEXT portion also includes references to the payload data stored within the DATA space in the datagram in order to identify which identifying addresses are associated with which portions of payload data.

Therefore, the server generates the HTTP smart packet in the form of an IP datagram enabling the packet to traverse over various layer 2 technologies before reaching the access point associated with the mobile communication device. Once the smart packet reaches the access
point, the data within is delivered wirelessly using an IEEE 802.11 packet format to the mobile user device.

Further, the datagram also includes within its DATA space the multiple portions of payload data associated with multiple URL requests received from multiple mobile communication devices.

It will be understood that the datagram also includes other standard data as required for the communication protocol. Further, it will be understood that the datagram being transmitted may be either a TCP or UDP datagram including an encapsulated IP datagram, where the IP datagram includes destination IP address storage areas storing the identification addresses of the mobile communication devices along with data storage areas storing the HTML data associated with the webpages being requested.

Therefore, the encapsulation module effectively delays or stalls the transmission of a datagram, which includes payload data associated with a URL request, thus enabling multiple portions of payload data associated with multiple URL requests from multiple mobile communication devices to be processed or included within the datagram. The process occurs constantly and in a chain effect emulating a traditional packet broadcast but in a more efficient manner as the processing throughput at the server is increased as the decision-making process of accepting or rejecting data packets is moved down stream to the mobile communication device.

The stalling of the transmission or transmission delay is realised by utilising a temporary storage module to capture the virtual addresses and multiple payloads associated with URL requests sent from a number of different mobile communication devices. The end of the delay is then triggered by an event such as the expiry of a time limit or an indication that a set storage capacity has been reached for example.
[00085] Once the datagram has been generated by the data packet encoding module, the datagram is transmitted by the proxy server over the wireless network via the access points to the access point the mobile communication device is connected to. The datagram is then transmitted wirelessly to the mobile communication device using 802.11 protocols.

[00086] As the number of data packets being sent through the wireless network is reduced, because multiple requests are combined within a single data packet, the data transmission speed associated with the network is maximised.

[00087] It will be understood that the system may encapsulate data associated with any number of requests sent by multiple mobile communication devices. For example, one data packet (datagram) may include up to 90 or more data portions associated with up to 90 or more URL requests. Effectively, the virtual addresses stored within the datagram are stored as an array, where the virtual addresses identify the associated payload data stored within the DATA portion of the datagram.

[00088] Further, it will be understood that if the requested URL is either not valid or the webpage is not available, the data provided back to the mobile communication device will include a "page not present" message instead of the actual webpage data.

[00089] A standard stack layer includes 7 layers as follows: Application layer; Presentation layer; Session layer; Transport layer; IP layer; MAC layer; Physical layer. According to various embodiments, a modified stack layer has been created as follows: Application layer; Presentation layer; Session layer; Transport layer; IP layer; Smart Packet layer; Modified MAC layer; Physical layer.

[00090] Various database entries are used by the modified stack as follows:
<table>
<thead>
<tr>
<th>Mac Address for Client</th>
<th>IP associated to client</th>
<th>Access-point MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC address client 1</td>
<td>IP address client 1</td>
<td>MAC address AP1</td>
</tr>
<tr>
<td>MAC address client 2</td>
<td>IP address client 2</td>
<td>MAC address AP1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>MAC address client x</td>
<td>IP address client x</td>
<td>MAC address APx</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>MAC address client N</td>
<td>IP address client N</td>
<td>MAC address AP1</td>
</tr>
</tbody>
</table>

[00091] The information from the external source (Internet) is received by the server that is requesting the information on behalf of the mobile device via a proxy server. The information/data follows the existing network stack to pass through Layer 6, Layer 5, Layer 4 (formed into a TCP/UDP segment) and then into Layer 3 where the packet is created which associates the mobile device data with the particular IP address assigned to the mobile device when the mobile device initially connected to the network.

[00092] A data aggregation process is driven by storing IP packets for multiple clients into buffers for each AP associated to the mobile device. The buffers are created when payload for a mobile device connected to a particular AP is received in the form of an IP Packet to the smart packet layer. Each buffer expires over time to ensure effective ongoing communication. To achieve this, the network stack and drivers for Ethernet or wireless are modified to store IP packets in the buffer. The buffer contains the destination device's MAC address and the IP packet created/formulated at IP layer for the destination device. The MAC address information is stored within a database on the server and can be retrieved on demand using the IP address of the destination device. The buffer is driven by a customisable timer (e.g. value <2ms) or a maximum size limit, e.g. up to 9000/18000 bytes. The hardware abstraction layer [HAL] includes modified drivers to these increased packet sizes to be handled.
On expiration of the timer or when the defined size limit is reached, the information stored within the buffer is ready to be parsed into the smart packet before being sent on to the MAC Layer (Layer 2). According to this embodiment, the information is parsed into the header and data section as shown in figure 8A and 8B.

As can be seen, the MAC address for a mobile device along with the size of the IP payload (stored in the buffer) and the offset for its location is stored within the header of the smart packet. The Type field allows the smart packet to indicate to the MAC Layer whether the packet is a 9000/18000 byte payload to be encapsulated within a frame.

Once the smart packet is created and the data has been encapsulated the smart packet is passed onto the MAC Layer. The MAC address of the AP associated with the particular buffer that has created the smart packet is retrieved from the database and assigned as the destination MAC address within the frame at MAC layer. The information is then passed along to the AP.

Once the frame is received at the AP, the AP checks to ensure that the whole frame has been received. After verifying the whole frame has been received, the AP passes the frame payload to the smart packet layer. At this layer a decision is made as to whether to transmit a smart packet to all of the mobile devices that are connected or separate the information for each individual mobile device.

If the AP determines that the mobile devices connected do not have the modified network stack to discern and retrieve information pertaining to them from a smart packet, the AP retrieves information pertaining to each mobile device and repackages the information through the following process.

The frame passes the smart packet to the upper layer, where this layer of the network stack parses the header and retrieves the
information for each payload (IP packet) for a particular mobile device. The information the AP uses from the header to retrieve this information is the offset and size values associated with the MAC Address.

[00099] Once having retrieved the IP packet for the mobile device, the smart packet layer sends the IP packet on to the MAC layer. Since the IP packet is coming from Layer 3, the MAC layer does an ARP request to find the MAC address for the mobile device IP address as seen in the IP packet that it receives. The MAC layer then adds this MAC address as the destination MAC address, computes the checksum creating a Layer 2 802.11 frame and transmits it over radio.

[000100] The mobile device will normally process the received frame first ensuring that it is destined for itself and then pass the payload to an upper layer (IP layer).

[000101] If the mobile devices connected have a modified network stack, the AP sets a flag for the MAC layer and passes the smart packet back to the MAC Layer. The flag indicates to the MAC layer that it needs to set the destination MAC address as the broadcast address as every mobile device connected to the AP needs to receive and parse this information. The process a mobile device receiving the smart packet performs is as follows.

[000102] The frame passes the smart packet to the smart packet layer once the frame has been checked for errors and can successfully pass along the payload (smart packet) to the upper layer. The packet, once received, is parsed and the MAC address associated with the mobile device is compared to the entries within the Header. If there is information for the MAC address for the mobile device, the relevant information (offset, size) to retrieve the IP packet is utilised.

[000103] The IP packet received is then passed along to the upper layer IP layer and the normal network stack operates henceforth.
Therefore, as described above, the smart packet is transmitted over a smart packet layer, which is effectively a layer 2.5 residing between the IP layer (layer 3) and MAC layer (layer 2). This 2.5 layer provides the ability for equipment e.g., server/access-point, to combine data destined for multiple client devices connected to the network in one large packet. This involves increasing the current Maximum Transmission Unit (MTU) from 1500 bytes to a large value such as 18000 bytes. By broadcasting one large packet, the access point need not send out single packets for each client device connected to the access point. The smart packet layer further requires the use of buffers to hold the incoming packets from the IP layer and then process the packets into the smart headers and payload for the packet.

Therefore, when communicating between each other, access points are able to exchange large chunks of information without polling for airtime in the wireless spectrum. Also, when an access point is communicating to multiple clients connected to the network, the access point is able to broadcast one large packet rather than sending one packet to each client connected which in a dense wireless environment causes interference and loss of packets. The smart packet layer is able to send information directly on to the MAC layer from the IP layer without modifying it, if required for mobile devices that do not specifically support this type of network stack. The MAC layer is modified for the purposes of this network to accommodate a larger smart packet as payload. The smart packet layer parses the IP packet as a payload for the client and associates that information with the MAC Address of the client to which the information is destined. Therefore, the network communication system is able to operate with mobile devices that have been specifically adapted to handle a 2.5 layer communication through the provision of specific drivers and coding, as well as mobile devices that can only handle layer 2 and layer 3 communications.

Each access point within the wireless network includes a
routing module which determines the route for transmitting URL requests sent by mobile communication devices back to the proxy server (or backhaul). Figure 4 shows a conceptual diagram of a routing module 401 forming at least part of a router configured inside a current access point. It will be understood that the router may be implemented in software, hardware or a combination of the two. The current access point is understood to mean the access point with which a mobile communication device is currently in communication. That is, the current access point is a reference point in order to understand the concept of the routing methodology according to this embodiment.

[000107] The routing module 401 includes a route determination module 403 that determines the destination (i.e. the next access point) for data packets being transmitted from the current access point. A data packet receiver and transmitter module 405 is arranged to receive data packets, and transmit the data packets based upon the determination made by the route determination module 403.

[000108] The route determination module 403 determines the fastest or most efficient route based upon data provided to it from a ping rate module 407 and a trace route module 409, as will be explained in more detail below.

[000109] It will be understood that the terms ping and traceroute have the following meaning. Ping is the speed and capacity of the link/connection between two access points, an access point and a server, or an access point and the Internet. Traceroute is the route taken taking into account the number of hops traversed and the load on each hop.

[000110] A memory module 411 is used within the routing module 401 to store a table holding the ping rates and traceroute distances calculated by the ping rate module 407 and trace route module 409 respectively.

[000111] The table within the memory module is arranged to store the
unique addresses of access points within direct communication distance from the current access point. That is, in this embodiment, the unique addresses are the MAC addresses of the neighbouring access points that are within the communication range of the current access point.

Further, the table also stores communication frequency values that are used to transmit and receive the traceroute distance and ping rate data. According to this embodiment, the communication frequency used for transmitting and receiving the traceroute distance and ping rate data is set to 5 GHz, i.e. a different frequency band to that used (2.4 GHz) for communicating the webpage data packets. Alternatively, the same communication frequency may be used for transmitting the webpage data packets and the ping rate & traceroute data.

Through the use of the table, the routing module 401 is able to determine the best route through the further access points for transmitting data packets back to the proxy server.

Referring to figure 5, a conceptual network diagram of interlinked access points is shown to describe an example of the methodology in determining the best route for data packets to be transmitted back to a proxy server or backhaul.

A current access point 501 is in communication with a mobile communication device 503. That is, the mobile communication device 503 is within the communication range 505 of access point 501. The mobile communication device 503 sends a URL request 507 to the access point 501.

Within direct communication distance of access point 501 are further access points (509, 511, 512 and 514). It will be understood that there may be additional further access points within direct communication distance of access point 501. The number of additional further access points will depend upon the proximity of the access points
and the communication range of the current access point.

[0001 17] The router module in access point 501 determines the best route back to the end point (proxy server) as follows.

[0001 18] Stored within the table within the router module are ping rates and traceroute distances for each of the further access points that the current access point is able to directly communicate with, i.e. each access point that the current access point is within direct wireless communication range. For example, according to figure 5, the current access point 501 is within communication range of the further access points 509, 511, 512 and 514. The ping rates between the current access point and each of these further access points are calculated by the ping rate module and stored in the table along with the traceroute distances between the current access point and each of these further access points. Therefore, according to this embodiment, there are four sets of ping rate values and traceroute distance values associated with the four further access points. It will be understood that the table within the current access point may store any number of sets of ping rate values and traceroute distance values associated with any number of access points.

[0001 19] According to this embodiment, the route determination module 403 identifies two preferred sets of ping rate values and traceroute distance values from all four sets. To identify the two preferred sets, the route determination module adds the traceroute distance (ms) to the ping rate (ms) for each of the further access points to calculate a route determination value (ms). According to this embodiment, the route determination value is termed a "heartbeat" value.

[000120] It will be understood that the route determination module may identify two or more preferred sets of ping rate values and traceroute distance values.

[000121] Further, it will be understood that the route determination
value may be calculated using any other suitable algebraic calculation such as one or more of averaging, addition, subtraction, multiplication and division or any combination thereof.

[000122] According to this embodiment, once the two identified preferred routes have been selected based on the calculations described above, the current access point 501 checks to see whether the identified further access points (i.e. the further access points associated with the identified preferred routes) are still in communication with the current access point. If the current access point 501 determines that the first preferred further access point is no longer in communication with the current access point, for example if there is a problem with the wireless router associated with that further access point, the current access point 501 selects the second preferred further access point.

[000123] In a situation where neither of the two preferred routes is available to the current access point, the current access point may either recalculate the ping rate values and traceroute values for all neighbouring access points to determine new preferred routes, or alternatively, the current access point may utilise a third, fourth or further route identified in the table.

[000124] After the route determination has been completed, the data packets are then transmitted from the current access point to the selected preferred further access point. Subsequently, the further access point that receives the data packet runs the same algorithm in order to determine the next access point that is to receive the data packet in the same manner as described herein.

[000125] The following provides a simple example of the method described above based on the network shown in figure 5.

[000126] The routing module within access point 501 determines that the traceroute data and ping rate data associated with access points 509
and 511 provide the two preferred fastest routes back towards the proxy server 510. For example, the table stored within the memory module of the routing module may include ping rate data and traceroute distance data for each of access points 509, 511, 512 and 514 as follows:

TABLE 1

Example of current Access Point table data

<table>
<thead>
<tr>
<th>AP MAC address</th>
<th>Trace route distance (ms)</th>
<th>Ping rate (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>509</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>511</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>512</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>514</td>
<td>12</td>
<td>5</td>
</tr>
</tbody>
</table>

[000127] It will be understood that for ease of understanding, the reference numerals used in figure 5 have been used as the MAC addresses for the access points in table 1 above.

[000128] Using the data in the example table 1 above, the route determination value is calculated for access point 509 = 11ms, access point 514 = 17ms, access point 511 = 13ms and access point 512 = 16ms. Therefore, the access points may be listed in preference as follows with the most preferred access point listed first: 509, 511, 512, 514. From this list, it can be seen that access points 509 and 511 are the two preferred access points as they have the lowest route determination value i.e. 11ms and 13ms respectively.
It can also be seen from this example that although access point 511 has a lower ping rate than access point 509 it is not the fastest route back to the proxy server (backhaul). This is because access point 511 has a higher traceroute distance than access point 509. The higher traceroute distance may be caused, for example, by increased loading on access point 511 due to the number of mobile devices connected to the access point.

Next, the access point 501 determines whether the access point 509 and access point 511 are still able to communicate with the current access point. If so, the access point 501 sends the data packet to the access point associated with the fastest route determination value (i.e. access point 509). If the access point associated with the fastest route determination value is not currently able to communicate with the current access point, the current access point sends the data packet to the access point associated with the second fastest route determination value, i.e. access point 511.

The table data is continually updated and so the two preferred further access points may continually change based on the updated ping rate and traceroute data values. For example, the table may be updated every 30 seconds or so. It will be understood that the update rate may be less or more than 30 seconds.

Further, after the route determination module in access point 501 has determined the next hop for forwarding the data sent by the mobile communication device 503, the receiving access point (e.g. 509) utilises its own routing module and route determination module along with its own table to determine the next hop. This continues as the data is transmitted between access points until the data is forwarded to the proxy server 510.

For example, referring back to figure 5, if the receiving
access point were access point 509, this access point has a communication footprint 513 enabling it to communicate with the previous access point 501 as well as two further access points 515 and 517. The route determination module within the routing module on the router within the access point 509 makes the determination whether to transmit the data packets to access point 515 or to access point 517 in the same manner as described above with reference to access point 501.

[000134] However, if the receiving access point were access point 511, this access point has a communication footprint 519 enabling it to communicate with the previous access point 501 as well as two further access points 517 and 521. The route determination module within the routing module on the router within the access point 511 makes the determination whether to transmit the data packets to access point 517 or to access point 521 in the same manner as described above with reference to access point 501.

[000135] By utilising a combination of the ping rates and traceroute values as described above to calculate a heartbeat value, the system takes into account not only the communication distance between access points but also the load being applied to each of the access points. Therefore, this provides an improved method for determining the fastest route back to the proxy server.

[000136] In order to provide free access to the wireless network, the proxy server injects advertising into data packets prior to them being transmitted back to mobile communication devices that have requested Internet resources.

[000137] Figure 6 shows a conceptual diagram of a proxy server 601 according to this embodiment.

[000138] As can be seen in figure 6, the proxy server 601 includes similar components as those shown in figure 2 discussed above. In
particular, the proxy server includes a URL request receiving module 203 which includes a buffer 205 for receiving and storing URL requests 602 in order of receipt from mobile communication devices within the wireless network. Further, a GPU module 209 receives the URL requests from the buffer 205. An HTML data receiving module 211 retrieves data from the Internet 111 based on the URL request received from the URL request receiving module 203. The GPU module 209 outputs page table data to a data packet encoding module 215, which encodes the page table data into a smart packet (datagram) 217 for transmission over the wireless network.

According to this embodiment, the proxy server further includes an advert storage module 603 that stores adverts for injection into datagrams being transmitted over the wireless network according to the method described below.

It will be understood that, as an alternative, the advert storage module 603 may be placed in a location away from the proxy server.

It will be understood that, as an alternative, the proxy server is not required to have the buffer 205. For example, according to a further embodiment, the proxy server may perform the following methodology without relying on the buffer 205.

Regarding the features of data encapsulation described above with reference to figures 2 and 3, it will be understood that these features may also be applied in conjunction with the methodology of advertising injection as described below. Alternatively, the advertising injection methodology may be applied without including the features of the data encapsulation method described above with reference to figures 2 and 3.

Further, it will be understood that the features of the route determination methodology described above with reference to figures 4
and 5 may also be applied in conjunction with the methodology of advertising injection as described below. Alternatively, the advertising injection methodology may be applied without including the features of the advertising injection methodology described above with reference to figures 4 and 5.

[000144] Figure 7 shows a conceptual diagram of a GPU 209 according to this embodiment. The GPU 209 includes an HTML source code retrieval module 703 which retrieves the HTML source code 701 from the Internet based on the URL request received at the proxy server. The HTML source code is forwarded to a webpage analysing module 705, where the HTML source code is analysed to determine the effects upon injecting advertising into a data packet that includes the original requested HTML source code. The webpage analysing module 705 may alternatively retrieve the HTML source code from a cache memory 707, as described above.

[000145] If the webpage analysing module 705 determines that the advertising is to be injected into the data packet including the HTML source code, the advertising is retrieved from an advert storage module 603 and instructions are provided to a data packet adjusting module 709. The data packet adjusting module 709 provides instructions to the encapsulation module 303 to enable the advertising to be injected into the data packet, as will be explained in more detail below. The encapsulation module then forwards the data to the data packet encoding module 215 to create the datagram 711 that is to be transmitted over the wireless network.

[000146] The following describes the advertising injection methodology implemented by the herein described system in more detail.

[000147] The webpage analysing module 705 determines whether the webpage that has been requested by the mobile communication device
can be displayed with the injected advertising without causing a display error above a predetermined error rate. The following section describes how the error rate is calculated by the webpage analysing module 705.

[000148] The webpage analysing module 705 retrieves webpage data from the HTML source code received by the HTML source code retrieval module 703 or the cache 707. The webpage data is analysed in a virtual environment to determine the original layout dimensions of the requested webpage. In particular, the X and Y dimensions of the webpage are determined.

[000149] Further, the webpage analysing module 705 creates a modified version of the webpage in the virtual environment where a space is inserted alongside the original HTML data that would be suitable for an advert intended to be supplied with the requested webpage. That is, the webpage analysing module is able to calculate the X and Y dimensions of the webpage as if an advert is being provided with the HTML of the datagram being transmitted, where the datagram includes the original HTML of the requested webpage alongside the HTML of the advert. Using the original dimensions calculated using the original webpage HTML and using the modified dimensions where the additional space has been provided, an error rate is determined based on the two sets of X and Y dimensions (i.e. the original and modified dimensions).

[000150] If the error rate is above a predetermined error rate value, such as 5% or 10% for example, the webpage analysing module instructs the data packet adjusting module not to inject the advertising alongside the original webpage. However, if the error rate is determined to fall below the predetermined error rate value, HTML used to display the advert is injected or encoded within the datagram so that it is displayed along with, e.g. above, below or next to, the requested webpage defined in the original HTML. In this way, the original HTML of the requested webpage is not affected or adjusted.
It will be understood that the predetermined error rate value may be more or less than 5%. Further, it will be understood that an error value may also be used that is a nominal value based on the dimensions measured rather than a percentage value.

In a very simple system, in order for the webpage analysing module to determine the original X and Y dimensions of the requested webpage, the source code of the requested webpage is parsed. During the parsing step, the webpage analysing module detects the styles, such as CSS references for example, within the HTML. Using the styles, the webpage analysing module detects the X and Y dimensions being used to display the webpage. Therefore, the webpage analysing module determines the width (X dimension) and height (Y dimension) of the requested webpage from the styles.

For more complex webpages, the webpage analysing module also takes into account any "div" tags, i.e. <div> tags or division tags, as well as table dimensions within the HTML of the requested webpage.

To calculate the original display dimensions associated with the requested Internet resource in situations where the webpages are more complex as described above, the webpage analysing module 705 obtains the HTML data from the HTML source code retrieval module 703 and executes one or more HTML dimension related scripts found within the retrieved HTML data. That is, the HTML dimension related scripts may include one or more of a stylesheet reference, division tag and a table reference, for example.

Within a virtual environment, the webpage analysing module utilises the HTML dimension related scripts to determine the original display dimensions associated with the requested webpage.

By referencing a Cartesian starting point, the webpage
analysing module executes each of the HTML dimension related scripts found within the retrieved HTML data to determine a Cartesian ending point. The ending point will have an X and Y position that will provide the webpage analysing module with the original width and original height dimensions of the requested webpage. Therefore, by rendering the retrieved HTML data within the virtual environment, the true width and true height dimensions of the original webpage may be determined by the webpage analysing module.

[000157] In order to calculate the modified display dimensions as if additional HTML in the form of an advert were provided alongside the original HTML data, the webpage analysing module inserts a space above, for example, the original HTML where the space dimensions directly correlate with the advert display dimensions.

[000158] The webpage analysing module then determines the modified dimensions within the virtual environment to calculate the X and Y values of the webpage in the same manner as described above using Cartesian starting points and Cartesian ending points as if it were to be displayed on a browser alongside the additional HTML. That is, a web browser is emulated within a virtual environment to determine in real time within the server the original dimensions of the requested webpage and the modified dimensions of the requested webpage with an additional advertising space to see whether the modified dimensions are within a predefined limit compared to the expected dimensions, where the expected dimensions are the original dimensions modified by the dimensions of the space.

[000159] For example, the webpage analysing module determines that the retrieved HTML creates a webpage in the virtual environment having (X,Y) dimensions of (600,1200), i.e. 600 pixels wide and 1200 pixels high (or deep).
[000160] If the additional HTML (e.g. an advert) is to have the additional dimensions (Xadd, Yadd) of (600, 60), i.e. 600 pixels wide and 60 pixels high (or deep), then the space inserted into the retrieved HTML webpage also has the dimensions (600, 60).

[000161] Based on the original dimensions and the additional dimensions, the expected dimensions may be calculated by the webpage analysing module. That is, as the X dimension is intended to be the same for both the original X dimension and the additional X dimension, the expected X dimension is also determined by the webpage analysing module to be equal to the original dimension X value and the additional dimension X value. Whereas, for the Y dimension the webpage analysing module adds the original Y dimension (Y) to the additional Y dimension (Yadd) to determine an expected Y dimension (Yexp).

[000162] For the X dimension the webpage analysing module checks to see whether the original X dimension (X) is the same as the additional X dimension (Xadd) to determine the expected X dimension (Xexp), which in this example would be the same as the original X dimension (X) as well as the same as the additional X dimension (Xadd).

[000163] Therefore, the webpage analysing module determines whether the modified webpage in the virtual environment has the expected modified dimensions (Xexp, Yexp) of (600, 1260), i.e. the same width as the original webpage and the same width as the advert, as well as the combined height of the original webpage and advert.

[000164] The webpage analysing module determines the actual modified dimensions (Xact, Yact) and compares one or more of the X and Y dimensions with one or more of the expected modified dimensions (Xexp, Yexp) to determine the error rate. If the error rate is above a predetermined value, the additional HTML is not provided with the original requested HTML. However, if the error rate is below the predetermined
value, the additional HTML is provided with the original requested HTML.

[000165] It will be understood that the webpage analysing module may only calculate an error rate based on the Y dimensions, therefore ignoring the X dimensions. Alternatively, the webpage analysing module may calculate an error rate based on both the X and Y dimensions.

[000166] For example, the webpage analysing module may calculate an error rate by determining the percentage difference between the actual modified dimensions and the expected modified dimensions.

[000167] As a first example, if the expected Y dimension (Yexp) was determined to be 1260 and the actual measured modified Y dimension (Yact) was determined to be 1260 then the error rate would be calculated by the webpage analysing module to be 0%, i.e. 100% - (Yexp/Yact * 100%).

[000168] Therefore, if the predetermined limit set for the error rate was 5%, the webpage analysing module would send instructions to the data packet adjusting module 709 to enable the data packet adjusting module to adjust the data packet in order to insert the additional HTML into the data packet. The additional HTML would be encapsulated within the data packet by the encapsulation module 303 so that the additional HTML is displayed above the original HTML without adjusting the original HTML.

[000169] Whereas, as a second example, if the expected Y dimension (Yexp) was determined to be 1260 and the actual measured modified dimension (Yact) was determined to be 1360 then the error rate would be calculated by the webpage analysing module to be 7.4%, i.e. 100% - (Yexp/Yact * 100%).

[000170] Therefore, if the predetermined limit set for the error rate was 5%, the webpage analysing module would not instruct the data packet adjusting module 709 to adjust the data packet by inserting the additional
It will be understood that it is not always necessary when inserting the space intended for the additional HTML to have a starting point that is at the X,Y dimension (0, 0) and that the starting point may be any other predetermined starting point. For example, the starting point may be halfway down the webpage, so the space is inserted at this point. For example, if the height of the webpage is determined from the style sheets to be 1200 pixels, the halfway (centre) point is calculated as being 600 pixels from the top of the webpage at (0, 600).

The space may be created within the virtual environment by including HTML such as:

```html
<script> function runTableDisplayO; </script>
```
where the function runTableDisplayO displays a space having a predefined dimension.

In order to analyse the modified dimensions, the following scripts may be used, for example:

```php
FUNCTION findWidthO

$width = output website to arraystring [Width]

A phpFunction may then be used to grab the total width from the output string arraystring [Width].

$height = output website to arraystring [Height]
```
A phpFunction may then be used to grab the total height from the output string arraystring [Height].

Example code or pseudo code for retrieving the original html.
and sending it to the website analysing module for analysis is provided below:

```plaintext
$Website = "google.com"; //Get website

$Compiled_Site = hardwareGenerator($Website); //Call hardware to output script to array

$Total_Width = $Compiled_Site($Width); //Get total width from hardware

$Total_Height = $Compiled_Site($Height); //Get total height from hardware

//Website Error Check

siteCheckWidth($Compiled_Site + (Total_Width + 600));

siteCheckHeight($Compiled_Site + (Total_Height + 60));
```

**Further Embodiments**

[0001 81] It will be understood that the embodiments of the present invention described herein are by way of example only, and that various changes and modifications may be made without departing from the scope of invention.

[0001 82] It will be understood that there may be one or more backhaul connections on the wireless network. Further, it will be understood that there may be any number of access points that are interconnected in the network.
Claims

1. A method of encapsulating data associated with uniform resource locator requests at a server for transmission over a Wi-Fi network, the method comprising the steps of the server creating a single page table by including in the single page table data associated with multiple uniform resource locator requests received at the server, wherein encoding of the page table into a datagram for transmission over the Wi-Fi network is delayed until a predetermined condition associated with the single page table occurs.

2. The method of claim 1, wherein the predetermined condition is the determination that the single page table has reached a predetermined size limit.

3. The method of claim 1, wherein the predetermined condition is the determination that a predetermined time limit for storing data in the single page table has expired.

4. The method of claim 1, wherein the step of creating a single page table comprises the step of storing the data associated with the multiple uniform resource locator requests in a temporary storage module.

5. The method of claim 4, wherein the data associated with the multiple uniform resource locator requests is continually added to the temporary storage module until the predetermined condition occurs.

6. The method of claim 5, wherein the method further comprises the step of, upon the predetermined condition occurring, transferring the data added to the temporary storage module to the single page table.

7. The method of claim 4, wherein the data associated with the multiple uniform resource locator requests is received from multiple mobile communication devices connected to the Wi-Fi network.
8. The method of claim 4, wherein the temporary storage module is a translation lookaside buffer (TLB)

9. The method of claim 1, whereupon the predetermined condition occurring, the method further comprises the step of encoding the data in the page table into a datagram for transmission over the Wi-Fi network.

10. The method of claim 1, wherein the data associated with the multiple uniform resource locator requests comprises multiple mobile communication device identification addresses and data associated with multiple uniform resource locators.

11. The method of claim 10, wherein the mobile communication device identification addresses is mapped to the data associated with multiple uniform resource locators in the page table.

12. The method of claim 10, wherein the data associated with multiple uniform resource locators comprises web page data associated with the multiple uniform resource locators.

13. The method of claim 12, wherein the webpage data comprises one or more forms of markup language data.

14. The method of claim 10, wherein the mobile communication device identification addresses comprises at least one of an IP address or MAC address associated with the mobile communication device.

15. The method of claim 10, wherein the data associated with the multiple uniform resource locators is retrieved from either a cache or the Internet.

16. A method of encapsulating data associated with uniform resource locator requests at a server for transmission over a Wi-Fi network, the method comprising the steps of the server:
receiving multiple uniform resource locator requests sent by
multiple mobile communication devices connected to the Wi-Fi network,
wherein the multiple uniform resource locator requests include a mobile
communication device identification address uniquely identifying one of
the mobile communication devices and a uniform resource locator (URL)
identifying an Internet resource requested by one of the mobile
communication devices;
retrieving a first set of web page data associated with an Internet
resource identified in a first uniform resource request received from a first
mobile communication device;
mapping, in a temporary storage medium, a first mobile
communication device identification address associated with the first
mobile communication device with the first set of web page data;
retrieving one or more further sets of web page data associated with one
or more Internet resources identified in one or more further uniform
resource locator requests received from one or more further mobile
communication devices;
mapping, in the temporary storage medium, one or more further
mobile communication device identification addresses associated with one
or more further mobile communication devices with the one or more
further sets of web page data associated with further URLs;
wherein the mapping of the one or more further mobile
communication device identification addresses with the one or more
further sets of web page data continues until a predetermined condition
associated with the temporary storage medium occurs,
whereupon said predetermined condition occurring,
the method further comprising the step of the server enabling the
encapsulation and encoding of identification addresses and web page
data mapped in the temporary storage medium into a datagram for
transmission over the Wi-Fi network.

17. A server for encapsulating data in a Wi-Fi network, the server
comprising a URL request receiving module adapted to receive multiple
uniform resource locator requests, and a processing module adapted to (i) create a single page table based on data in the received multiple uniform resource locator requests and (ii) delay encoding of the single page table into a datagram for transmission over the Wi-Fi network until a predetermined condition associated with the single page table occurs.

18. The server of claim 17 further adapted to perform the method steps of any one of claims 1 to 15.

19. A method of determining a fastest route between a current access point in a Wi-Fi network and an end point in the Wi-Fi network via a plurality of further access points, the method comprising the steps of determining ping rates and traceroute distances between the access points, calculating a route determination value based on a combination of the determined ping rates and traceroute distances, and determining the fastest route based on the calculated route determination value.

20. The method of claim 19 further comprising the step of determining the ping rates and traceroute distances between the current access point and one or more of the plurality of further access points in the Wi-Fi network that are in direct communication with the current access point.

21. The method of claim 20 further comprising the step of calculating the route determination value by combining the determined ping rates and traceroute distances.

22. The method of claim 21 wherein the ping rates and traceroute distances are combined by performing one or more algebraic calculations on the ping rates and the traceroute distances.

23. The method of claim 22 wherein the one or more algebraic calculations comprise one or more of averaging, addition, subtraction, multiplication and division of the ping rates and traceroute distances.
24. The method of claim 19 further comprising the step of storing the ping rates and traceroute distances in a table at the current access point, wherein the stored ping rates and traceroute distances are associated only with further access points that are in direct communication with the current access point.

25. The method of claim 24 further comprising the step of refreshing the stored traceroute distances and ping rates at predefined intervals.

26. The method of claim 24 further comprising the step of accessing the stored ping rates and traceroute distances in the table to determine a selected two or more further access points that have the lowest ping rates and traceroute distances, and based on said determination determining the fastest route.

27. The method of claim 26 further comprising the step of determining from the selected two or more further access points whether the two or more further access points are still in direct communication with the current access point, and based on said determination determining the fastest route.

28. The method of claim 19 further comprising the step of storing a unique address identification of the one or more of the further access points in a table at the current access point, wherein the stored unique address identifications are associated only with further access points that are in direct communication with the current access point.

29. The method of claim 19 further comprising the step of storing a communication frequency associated with the communication channel used to determine the ping rates and traceroute distances.

30. A routing module for use in a server within a Wi-Fi network, the routing module comprising a route determination module, a ping rate module, a trace route module and a memory module, the routing module
configured to determine the fastest route between a current access point in the Wi-Fi network and an end point in the Wi-Fi network via a plurality of further access points, wherein the ping rate module is arranged to determine ping rates between the access points, the traceroute module is arranged to determine traceroute distances between the access points and the route determination module is arranged to (i) calculate a route determination value based on a combination of the determined ping rates and traceroute distances and (ii) determine the fastest route based on the calculated route determination value.

31. The routing module of claim 30 wherein the route determination module is further arranged to determine the ping rates and traceroute distances between the current access point and one or more of the plurality of further access points in the Wi-Fi network that are in direct communication with the current access point.

32. The routing module of claim 30 wherein the route determination module is further arranged to calculate the route determination value by combining the determined ping rates and traceroute distances.

33. The routing module of claim 32 wherein the route determination module combines the ping rates and traceroute distances by performing one or more algebraic calculations on the ping rates and the traceroute distances.

34. The routing module of claim 33 wherein the one or more algebraic calculations performed by the route determination module comprises one or more of averaging, addition, subtraction, multiplication and division of the ping rates and traceroute distances.

35. The routing module of claim 30 further comprising a memory module arranged to store the ping rates and traceroute distances in a table at the current access point, wherein the stored ping rates and traceroute distances are associated only with further access points that


are in direct communication with the current access point.

36. The routing module of claim 35 wherein the routing module is arranged to refresh the stored traceroute distances and ping rates at predefined intervals.

37. The routing module of claim 35 wherein the routing module is arranged to refresh the stored traceroute distances and ping rates at predefined intervals.

38. The routing module of claim 37 wherein the route determination module is further arranged to access the stored ping rates and traceroute distances in the table to determine a selected two or more further access points that have the lowest ping rates and traceroute distances, and based on said determination determining the fastest route.

39. The routing module of claim 30 further comprising a memory module arranged to store a unique address identification of the one or more of the further access points in a table at the current access point, wherein the stored unique address identifications are associated only with further access points that are in direct communication with the current access point.

40. The routing module of claim 30 further comprising a memory module arranged to store a communication frequency associated with the communication channel used to determine the ping rates and traceroute distances.

41. A method of providing additional HTML data alongside HTML data associated with an Internet resource request for transmission over a network, the method comprising the steps of:

receiving an Internet resource request,
retrieving the HTML data associated with the Internet resource request,
executing, in a virtual environment, the retrieved HTML data to determine the original display dimensions associated with the retrieved HTML data,
determining, within the virtual environment, modified display dimensions associated with a modified display which has additional HTML data displayed alongside the retrieved HTML data,
determining an error rate based on the original display dimensions and modified display dimensions, and
determining whether to provide the additional HTML data alongside the retrieved HTML data based upon the error rate.

42. The method of claim 41 further comprising the step of determining the original display dimensions of the retrieved HTML data in the virtual environment by executing one or more HTML dimension related scripts within the retrieved HTML data.

43. The method of claim 42 wherein the step of determining the original display dimensions further comprises the steps of determining a reference dimension point in the virtual environment, executing the one or more HTML dimension related scripts, determining an end dimension point in the virtual environment and determining the original display dimensions based on the end dimension point.

44. The method of claim 42 wherein the one or more HTML dimension related scripts include one or more of a stylesheet reference, a division tag, and a table.

45. The method of claim 41 wherein the step of determining the modified display dimensions comprises the step of executing an HTML script with the retrieved HTML data within the virtual environment to create a space having dimensions associated with the additional HTML data.
46. The method of claim 45 further comprising the step of determining the modified display dimensions by calculating the width and height of the modified display.

47. The method of claim 46 wherein the width and height of the modified display are determined by executing a findWidth function and executing a findHeight function.

48. The method of claim 41 further comprising the step of determining expected display dimensions associated with the modified display dimensions and determining the error rate based on the expected display dimensions and the modified display dimensions.

49. The method of claim 41 further comprising the steps of encoding the additional HTML data in the same datagram as the retrieved HTML data when the determined error rate is within a predefined limit.

50. A server arranged to provide additional HTML data alongside HTML data associated with an Internet resource request for transmission over a network, the server comprising an Internet resource request receiving module arranged to receive an Internet resource request, an HTML data retrieval module arranged to retrieve the HTML data associated with the Internet resource request, a processing module arranged to (i) execute, in a virtual environment, the retrieved HTML data to determine the original display dimensions associated with the retrieved HTML data, (ii) determine, within the virtual environment, modified display dimensions associated with a modified display which has additional HTML data displayed alongside the retrieved HTML data, (iii) determine an error rate based on the original display dimensions and modified display dimensions, and (iv) determine whether to provide the additional HTML data alongside the retrieved HTML data based upon the error rate.

51. The server of claim 50 further arranged to perform the method steps as claimed in any one of claims 42 to 49.
52. The method of any one of claims 1 and 16 substantially as herein described with reference to the accompanying figures.

53. The server of claim 17 substantially as herein described with reference to the accompanying figures.

54. The method of claim 19 substantially as herein described with reference to the accompanying figures.

55. The routing module of claim 30 substantially as herein described with reference to the accompanying figures.

56. The method of claim 41 substantially as herein described with reference to the accompanying figures.

57. The server of claim 50 substantially as herein described with reference to the accompanying figures.
FIGURE 3
### Figure 8A

<table>
<thead>
<tr>
<th>Smart Packet Header</th>
<th>Payload &amp; CRC Client 1</th>
<th>Payload &amp; CRC Client 2</th>
<th>..........</th>
<th>Payload &amp; CRC Client N</th>
</tr>
</thead>
</table>

### Figure 8B

<table>
<thead>
<tr>
<th>Type ID</th>
<th>Packet Size</th>
<th>Reserved for future use</th>
<th>Mac Address of Client 1 and associated offset for payload and size</th>
<th>Mac Address for Client 2 and associated offset for payload and size</th>
<th>....</th>
<th>Mac Address for Client N and associated offset for payload and size</th>
<th>Header CRC</th>
</tr>
</thead>
</table>


INTERNATIONAL SEARCH REPORT

INTERNATIONAL application No.
PCT/AU2012/001552

A. CLASSIFICATION OF SUBJECT MATTER

H04W 88/00 (2009.01)  H04W 84/10 (2009.01)  H04W 40/00 (2009.01)  G06Q 30/00 (2012.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Google Patents, EPDOC, WIPO, INSPEC, EPODOC, TXTUSO, TXTUSI, TXTUS2, TXTUS3, TXTUS4, TXTEP1, TXTGB1, TXTWOI. Keywords: (Wi-Fi, wireless, network?, Uniform_resource_locator?, encode+, encapsulate+, encrypt+, map+, data+, delay, critera+, condition?, UDP, lookaside w buffer/storage/queue/memory, IP_address, identifi+, mark_up data, virtual_address, proxy_server, access_point?, UE?, mobile?, smart_phone?, access_point?, gateway?, internet, rout+, (fast+ or quick+ or short+ or optim+ or efficient or best) w route, access w point?, AP, ping w rate?, trace_route, route w determin+, table, direct w communication, address w identification, access w identification?, communication w channel?, interface w modules?, 802.11s w mesh w interface, heartbeat, route w determination w value, html or web page or pay_load, additional or extra or supplemental or associate or advert+ or commercial, internet w resource w request?, modified or target+ or updat+ or expect+ or estima+, display, dimension? or size, error or discrepancy or difference) and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

Documents are listed in the continuation of Box C

Further documents are listed in the continuation of Box C

See PCT/ISA/210 I family annex

* Special categories of cited documents:
  * "A" document defining the general state of the art which is not considered to be of particular relevance
  * "E" earlier application or patent but published on or after the international filing date
  * "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  * "O" document referring to an oral disclosure, use, exhibition or other means
  * "P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
22 April 2013

Date of mailing of the international search report
22 April 2013

Name and mailing address of the ISA/AU

AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
Email address: pa@ipaustralia.gov.au
Facsimile No.: +61 2 6283 7999

Authorised officer

Kusha Punta
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No. 0262832833

FormPCT/ISA/210 (fifth sheet) (July 2009)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 2008/0214175 A1 (PAPADOGLOU ET AL.) 04 September 2008 abstract, FIG. 1, paragraphs 20-21 and 27</td>
<td>1-18</td>
</tr>
</tbody>
</table>
This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.:
   because they relate to subject matter not required to be searched by this Authority, namely:

2. X Claims Nos.: 52-57
   because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
   See Supplemental Box

3. Claims Nos:
   because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a)

This International Searching Authority found multiple inventions in this international application, as follows:

See Supplemental Box for Details

1. X As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. □ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. □ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. □ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

□ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

□ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

X No protest accompanied the payment of additional search fees.
Continuation of Box II
The claims do not comply with Rule 6.2(a) because it/they rely on references to the description and/or drawings.

Continuation of Box III
This International Application does not comply with the requirements of unity of invention because it does not relate to one invention or to a group of inventions so linked as to form a single general inventive concept.

This Authority has found that there are different inventions based on the following features that separate the claims into distinct groups:

- Claims 1-18 are directed to a method/server for encapsulation/encoding of data associated with uniform resource locator requests at a server for transmission over a Wi-Fi Network. The feature of delaying the encoding data into a datagram for transmission over the Wi-Fi network is until a predetermined condition is specific to this group of claims.

- Claims 19-40 are directed to a method/device for route determination between an access point and an end point in a Wi-Fi Network. The feature of determining the fastest route by calculating a route determination value based on a combination of ping rates and traceroute distances is specific to this group of claims.

- Claims 41-51 are directed to a method/server for providing additional HTML data associated with an internet resource request for transmission over a Wi-Fi network. The feature of determining whether to provide additional HTML data alongside retrieved HTML data based upon the error rate is specific to this group of claims.

PCT Rule 13.2, first sentence, states that unity of invention is only fulfilled when there is a technical relationship among the claimed inventions involving one or more of the same or corresponding special technical features. PCT Rule 13.2, second sentence, defines a special technical feature as a feature which makes a contribution over the prior art.

When there is no special technical feature common to all the claimed inventions there is no unity of invention.

In the above groups of claims, the identified features may have the potential to make a contribution over the prior art but are not common to all the claimed inventions and therefore cannot provide the required technical relationship. Therefore there is no special technical feature common to all the claimed inventions and the requirements for unity of invention are consequently not satisfied a priori.