A METHOD FOR INCREASING AVAILABLE INTERNET ADDRESSES

BACKGROUND OF THE INVENTION

The present invention relates to a method for increasing the number of available internet addresses.

The number of unique internet addresses provided by the existing IPv4 standard of IP (internet protocol) addresses is approximately 4×10^9, which has created - or will soon lead to - a worldwide shortage of IP addresses. This shortage has the potential to become particularly acute owing to the trend towards connecting an increasing number of devices to the internet, each device requiring one or more physical IP address.

The worldwide governing body of the internet, the Internet Engineering Task Force (or IETF) is presently trying to establish a new standard, IPv6, to address this problem, but IPv6 is only a theoretical solution and faces major challenges in implementation and deployment. For example, IPv6 must accommodate existing devices as well as new devices, so that older devices can communicate with new, IPv6 compliant devices. If not, it will be necessary to upgrade existing devices to become IPv6 compliant, which will itself be complex and expensive. Also, the capacity of IPv6 compliant devices is a significant issue in the deployment of IPv6, as these devices will require almost triple the current necessary capacity to support the new standard.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an internet addressing system that is compatible with the existing system and but which allows the number of available addresses to be greatly increased.

The present invention provides, therefore, a method of

directing data from a source device to a destination device over a computer network, comprising:

forming a data packet including said data, conventional address data and additional address data;

extracting from said data packet said additional address data; and

transmitting said data packet to said destination device as indicated either by said additional address data, or by said address data and said additional address data.

Preferably said additional address data is accommodated in said data packet in a field provided for other data, such as said data or said conventional address data, whereby said other data is contained with said additional address data in said field.

Thus, it is not necessary to modify the data packet structure to accommodate the additional address data; rather it is stored in a field normally used for other information (such as the data or the conventional address data) along with that other information.

Preferably said additional address data is included in a field provided for said data or said conventional address data.

Thus, this can be achieved without the need to alter existing applications or devices, as the data packet (which will be referred to below as a virtual data packet) can be created from a conventional data packet, with the additional address data in the content data or conventional address data so that the virtual data packet can be understood by existing devices.

Preferably said computer network is the internet, and said address data is internet protocol address data.

Thus, the method of the invention can be used as an extension (rather than a replacement) of the present IP domain system, as the data field of the data packets is used to include additional address information, thereby extending the number of possible unique addresses. The method will typically co-exist with the current IPv4 domain system and employ the existing technological infrastructure; the method can be used to multiply the number of available IP addresses within the existing data packet under IPv4 standards.

In some embodiments, the method includes forming said data packet over an existing, conventional data packet through existing standards.

The additional address data may indicate that the address data is adequate to uniquely identify the destination address, in which case the address of the destination device is indicated by both the address data and the additional address data.

Preferably said address data includes either source address data or destination address data, or both source address data and destination address data. More preferably, said additional address data includes the additional destination address data.

Preferably said forming said data packet to include additional address data is performed by a gateway means. Preferably said method includes recognizing and routing said data packet by means of said gateway.

The gateway means could be an actual, physical server, or could be implemented as software or a plug-in, such as in a computer operating system.

Preferably said address data corresponds to said gateway

means. More preferably said destination address data corresponds to said gateway means and said source address data corresponds to a further gateway means.

Thus, many destinations can be associated with a single IP address (i.e. that of a gateway means) and many sources can be associated with a single conventional IP address (i.e. that of the further gateway means).

The present invention further provides a method of directing data from a source device to a destination device over a computer network, comprising:

forming a data packet including said data and address data, said address data resolvable into conventional address data and additional address data;

transmitting said data packet to said destination device as indicated either by said additional address data, or by said conventional address data and said additional address data.

The present invention also provides an apparatus for directing data from a source device to a destination device over a computer network, comprising:

means for forming a data packet including said data, conventional address data and additional address data;

means for transmitting said data packet; and

means for extracting from said data packet said additional address data, and transmitting said data packet to said destination device as indicated either by said additional address data, or by said address data and said additional address data.

BRIEF DESCRIPTION OF THE DRAWING

In order that the present invention can be more clearly ascertained, preferred embodiments will now be described, by way of example, with reference to the accompanying

drawing, in which:

Figure 1 is a schematic view of an internet protocol data packet according to the prior art;

Figure 2a is a schematic view of an example of a TCP/IP data packet, depicting implicit representation of additional address data according to a preferred embodiment of the present invention;

Figure 2b is a schematic view of a virtual internet protocol data packet according to a preferred embodiment of the present invention;

Figure 3a is a schematic representation of an extended domain space, in relationship to the internet, according to a preferred embodiment of the present invention;

Figure 3b is a schematic representation of data flow over the extended domain space of figure 3a invention;

Figure 4 is a flow-chart of the sequence depicted in figure 3b;

Figure 5 is a schematic representation of the routing of a data packet according to the present invention, for the case where all nodes are within the existing IP domain space;

Figure 6 is a schematic representation of the routing of a data packet according to the present invention, for the case where all nodes except the destination are within the existing IP domain space; and

Figure 7a is a schematic representation of the routing of a data packet according to the present invention, for the case where all nodes except the source and destination are within the existing IP domain space;

Figure 7b is a schematic representation of an example of a scenario in the environment shown in figure 7a, utilizing the Request-URI field as the implicit destination address for virtual address packet switching.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 is a schematic representation of a conventional

IPv4 data packet 10, including header 11, packet size 12, source address 13, destination address 14, data 15 and trailer 16. In a method of data packet transmission according to a preferred embodiment of the present invention, it is the data 15 that is used to identify an implicit additional address relating to destination. Such an implicit representation of additional address data is shown in figure 2a, which depicts an example of a TCP/IP data packet 17. Data packet 17 includes TCP/IP header 18, which comprises Packet Header Information (e.g. Packet Size, TCP Port info, UDP indicator, etc.) 19, source address 13 and destination address 14. Data packet 17 also includes fields 20 for specifying data source (request/response): the HTTP header 21a, and the HTTP message body 21b (i.e. the content of data field). The HTTP header 21a comprises the Request-URI ("Uniform Resource Identifier") field, which identifies the resource (server or service) as the implicit virtual destination address according to RFC2616 (HTTP Protocol), Section 5.1.2 & 5.2. In more general terms, the implicit virtual destination address field takes the position of the unique assigned field of its corresponding Point-to-Point communications layer, e.g. TCP Port Number, HTTP Request-URI, etc.

Thus, the additional destination address can be implicit because according to the method of the present invention it is unnecessary to physically append an additional address field within the data packet. Rather, the additional address field is incorporated within the data packet as a unique identifier from the specific application itself. In web browsing, such as with the example shown in figure 2a (using the HTTP protocol as a higher level communications protocol), the implicit nature of the additional destination address is implied (or incorporated) as the Request-URI field when the requester (or web browser) sends a request to a web server.

Thus, referring to figure 2b, a "virtual address" (VA) data packet 22 according to the invention includes the same fields as in data packet 10 of figure 1, but in addition the data field includes a destination virtual internet protocol address 23. By this method existing 32-bit IP addresses are, in effect, multiplied by having implicit additional address data assigned as VAs to a fixed set of IP addresses.

Each VA is a unique identifier based on the type of application protocol services (HTTP, NNTP, FTP, TELNET, POP3, SMTP, HTTPS, etc) used. This enables each VA to be implicitly appended to the conventional internet protocol address field. All VA packets are represented with an additional unique destination address. Thus, a VA packet represented with both 'legal' (or public) source/destination public internet protocol addresses is equivalent to a conventional IP packet. A VA packet represented with either 'illegal' (or private) source/destination internet protocol address activates the VA numbering system of the present invention.

Figure 3a depicts schematically an extended domain space according to a preferred embodiment of the present invention, and its relationship with the conventional domain space. The conventional domain space 24 includes the internet 25, internet service providers 26a,b,c,d and various devices (such as browser 27a, internet enabled telephone 27b and mobile office 27c).

The extended domain space 28 is visible to conventional domain space 24 via gateway 29, which has a single, conventional IP address. Gateway 29 provides full connectivity 30 (including local and remote, via LAN, WAN, Dial-up, etc) to an essentially unlimited number of devices, represented in this example by Web Server 32a,

Apps Server 32b, Realaudio Server 32c, Multi-Media Server 32d, WAP Server 32e, Phone Server 32f, Database Server 32g, Ftp Server 32h, Newsgroup Server 32i, Email Server 32j and Telnet Server 32k. The devices 32a to 32k have VAs, respectively, of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and further such VAs may be assigned essentially without limit.

Conventional IP data packets are identified as VA data packets in a system domain gateway in the following manner.

Figure 3b is a schematic representation of data flow over an extended domain space according to a preferred embodiment of the present invention.

Standard web browser 38 issues a HTTP (hypertext transfer protocol) request when a user of browser 38 wishes to download information from a web server 40. The HTTP request is received by first system gateway 42, which determines if the request is for a node found on the existing Internet, or within the extended domain. This is determined on the basis of the destination IP address 18 (see Figure 1), which will correspond to an entry on a conventional DNS, or the DNS of the first extended domain gateway 42. If the corresponding IP address is a public IP address, the request will be effected within the existing IP domain, otherwise, the request is effected within the extended domain. Upon retrieval of the source route (data packet from the source, browser 38), the first gateway will put this information into its source route table in order to trace the return path of the request.

Upon receipt of the corresponding IP address associated with the destination requested, web browser 38 converts the request into a VA data packet (see figure 2b), in which the full address information is contained in the data field. The VA data packet is then directed by first router 44a over the existing IP network 46 via second router 44b to

second extended domain gateway 48. Second gateway 48 translates the request and forwards it to web server 40, which responds by creating a VA data packet containing the requested data. The web server 40 then forwards the requested information to second gateway 48 to be transported to the requester (browser 38). The VA packet is transmitted via second router 44b, the Internet 46 and first router 44a to first gateway 42, which translates the response back as a conventional IP data packet, which is transmitted to browser 38 based on the source route table at the first gateway 42. This sequence is also summarized in figure 4. A request, such as a HTTP request 50, is forwarded to the first gateway 52 which determines 54 if the request is for a node within the extended domain or outside the extended domain. If it determines that, YES, the request is for a node within the extended domain, the request is submitted 56 to the extended domain DNS. If not, the request is submitted 58 to a standard DNS (though, in practice, the extended DNS and the standard DNS might well be part of the same physical server). The respective DNS returns 60 the value of the corresponding IP to the requester 50, which encapsulates 62 the IP and VA data into a VA data packet. The first gateway 52 then sends 64 the VA data packet to a router and from there to the Internet 66. The routed VA data packet 68 is received by second extended domain gateway 70, which strips the VA data packet and checks 72 whether the data packet is indeed an extended domain data packet.

If YES, the second gateway 70 communicates 74 with the web server (40 in figure 3b) located using the destination VA contained in the VA data packet and the corresponding private IP address associated with it. If not, the second gateway 70 communicates with the server using its original IP destination address value to locate the server.

The requested data is retrieved 78 from the server, which

sends the IP and VA information to the second gateway 70. The server encapsulates 80 the IP and VA information into a VA data packet, and sends 82 the VA data packet to the router and from there the internet 66.

The routing of a VA data packet, as it is manipulated by the gateways, is shown schematically in figure 5 (demonstrated by means of a HTTP service), for the simple case where the system operates with the existing IP domain space but with all nodes within that existing IP domain space. Further, this example assumes that there are only six legal (or public) IP addresses to accommodate up to six devices or nodes in the world, such that browser 38 has an IP address of 1, first gateway 42 an IP address of 2, first router 44a an IP address of 3, second router 38 an IP address of 4, second gateway an IP address of 5 and web server 40 an IP address of 6. The web server 40 is also identified with a server hostname of "castle". As all the nodes are in the existing IP domain space, the VA data packet 84 assembled by browser 38 has public source address of 1, and public destination address of 6 (i.e. web server 40). The destination VA is assigned to be "castle", implicitly indicating that the destination has a public IP address and that no VA is required. The packet 84 also includes data 86 (the VA embedded within the data 86).

This VA data packet 84 is routed to second gateway 48, and is received as VA data packet 88. The second gateway 48 is not required to amend the VA data, as the destination (i.e. web server 40) - as noted above - is within the IP domain space (i.e. a public IP address is identified). Thus, the second gateway 48 forwards VA data packet 90 (essentially identical to VA data packet 84) to web server 40. Web server 40 responds by sending the requested information in an IP data packet to the second gateway 48 in the form of VA data packet 92 containing the requested data 94, a source IP address of 6 (i.e. the web server 40), and a

destination IP address of 1 (i.e. the browser 38). As in VA data packet 84, the destination VA is "castle". This VA data packet 92 is received by first gateways 42 as VA data packet 96, converted to conventional IP data packet and forwarded to browser 38 on the basis of having destination public IP address data of IP=1.

A more complex example of this protocol is shown schematically in figure 6, in which - as in figure 5 - there are only six IP address in existence (five public IP addresses within the IP domain and one private IP address within the extended domain), but where the web server 98 from which information is requested is within the extended domain rather than the existing IP domain space and consequently addressable only by means of a VA. In this example, web server 98 has a VA of "castle", and is the first VA in a world having only five IP addresses. This VA of "castle" may be represented according to the method of this invention.

Thus, web browser 38 encapsulates a HTTP request as VA data packet 100, in which the source IP address is 1 (i.e. that of browser 38) and the destination IP address 101a as 5 (i.e. that of second gateway 48). The destination VA data 101b is "castle", indicating that the ultimate destination is web server 98. VA data packet 100 also includes data 102.

The VA data packet 100 is received by second gateway 48 as VA data packet 104 and forwarded to the web server 98 as VA data packet 106, in which the source IP address is now that of the second gateway 48. The destination IP address, however, is translated by gateway 48 by comparison with the VA of the web server 98 (i.e. "castle"). In this example, the destination private IP address is 7 in the extended domain. VA data packet 106 is received by the web server 98, and a new VA data packet 108 is transmitted back to the

second gateway 48 with the destination VA data unchanged but disregarded (cf. VA data packet 106) and with the requested data 110. In response to receiving VA data packet 108, second gateway 48 sends VA data packet 112 to first gateway 42. VA data packet 112 is essentially identical to VA data packet 108, except that the source IP address is now that of the second gateway 48 (i.e. IP=5) and the destination IP address is that of the ultimate destination (browser 38), viz. 1. This packet is received by first gateway 42 and directed to the browser 38 as VA data packet 114.

Figure 7a is a schematic representation of a further example, similar to figure 6, in which there are only four public domain IP addresses in the world, but in which the web server have VA along with private IP addresses. The browser does not require a VA since only destinations require VA, not sources. The browser is identified by its private IP address of 9 within the extended domain. Being in the extended domain, the HTTP requests generated by browser 116 will contain relevant implicit VA information, so browser 116 generates a HTTP request in the form of VA data packet 118, containing source private IP address 9 (i.e. the IP address in the extended domain), destination public IP address 2 (i.e. that of the first gateway 42), destination VA "castle" (i.e. the VA of web browser 98) and data 120. This VA data packet is received by first gateway 42 and transmitted as VA data packet 122, essentially identical to VA data packet 118, but with the source and destination IP addresses changed to correspond to the IP address of the immediate source (first gateway 42), i.e. 2 and the immediate destination (second gateway 48), i.e. 5.

The data packet is received by second gateway 48 as VA data packet 124; the second gateway 48 generates VA data packet 126 by amending the source IP address to be that of the second gateway, i.e. 5, and the destination IP address to

be the IP equivalent (i.e. 1) of the VA "castle" of the web server 98. Web server 98 receives VA data packet 126 and transmits a new VA data packet 128 back to the second gateway 48 with destination VA data unchanged but disregarded (cf. VA data packet 126) and with the requested data 130. In response to receiving VA data packet 128, second gateway 48 sends VA data packet 132 to first gateway 42. VA data packet 132 is essentially identical to VA data packet 128, except that the source IP address is now that of the second gateway 48 (i.e. IP=5) and the destination IP address is now that of the immediate destination, second gateway 48, viz. 2. This packet is received by first gateway 42 as VA data packet 134; the first gateway 42 generates VA data packet 136 (which mirrors original data packet 118 but with new data 130), which is sent to browser 116 at private IP address of 9. The final point-to-point routing from first gateway 42 to browser 116 is accomplished through a look-up table at the first gateway 42, holding the information of the source route of the browser 116.

Figure 7b depicts an example of a scenario that could arise in the environment shown in figure 7a, utilizing the Request-URI field as the implicit destination address for virtual address packet switching according to the present method (as defined in RFC2616 (HTTP Protocol), Sections 5 & 6). The browser 116 is again identified by its private IP address of 9 within the extended domain, and generates in this example a HTTP request (in the form of VA data packet 138) with request-URI=http://www.castle.com/index.html. Thus, VA data packet 138 (like data packet 118) contains source private IP address 9, destination public IP address 2, destination VA http://www.castle.com/index.html and data 140. This VA data packet 138 is received by first virtual domain gateway 42 and transmitted as VA data packet 142, with the source and destination IP addresses changed to correspond to the IP address of the immediate source (first

gateway 42), i.e. 2 and the immediate destination (second gateway 48), i.e. 5.

The data packet is received by second gateway 48 as VA data packet 144; the second virtual domain gateway 48 generates VA data packet 146 by amending the source IP address to be that of the second gateway, i.e. 5, and the destination IP address to be the IP equivalent (i.e. 1) of the VA of the web server 98 (viz. http://www.castle.com/index.html). Web server 98 receives VA data packet 146 and transmits a new VA data packet 148 back to the second gateway 48 with destination VA data unchanged but disregarded and with the requested data 150. In response to receiving VA data packet 148, second gateway 48 sends VA data packet 152 to first gateway 42. VA data packet 152 is essentially identical to VA data packet 138, except that the source IP address is now that of the second gateway 48 (i.e. IP=5) and the destination IP address is now that of the immediate destination, second gateway 48, viz. 2. This packet is received by first gateway 42 as VA data packet 154; the first gateway 42 generates VA data packet 156 (which mirrors original data packet 138 but with new data 150), which is sent to browser 116 at private IP address of 9.

Thus, the gateways of the invention accomplish the following functions :

- providing the DNS name-to-address resolution service;
- Resolving data packets of existing IP domains and extended domains;
- Maintaining a source route table;
- Routing data packets from point to point.

In these examples, all devices have either private IP addresses or public IP addresses, as the examples pertain to an implementation of the invention over the IP network(s). However, the technology may also be used between devices that are based on other network addressing

systems, so that the only devices required to have IP addresses are the gateways 42 and 48, and the routers 44a and 44b. It will also be understood that the gateways can exist in several forms, including as physical servers, software plug-ins or firmware implementations within operating systems of electronic devices (e.g. personal computers, mobile devices, home appliances, vendor machines, information kiosks, etc.).

The invention could be implemented in several ways, including at the internet service provider level, so that all subscribers to the ISPs are VA enabled and hence will be able to communicate with extended domain, or installed as a plug-in into the operating systems of devices which may or may not be linked to the extended domain infrastructure.

Applications requested by a user that are within the extended domain may require users to install or have already installed a plug-in in their internet device, though this is not essential. If a plug-in is not employed, all users will generally communicate with the VA gateways that are deployed over the internet using standard applications and devices, e.g. Web Browsers, Web Servers, FTP Servers, Email Servers, etc.

If a plug-in is employed, the method can be implemented in two ways: 1) the plug-in can translate the application request and direct the packets to the nearest gateway where packet is transformed to a VA data packet and routed; 2) the plug-in can translate the application request, transform the IP packet to VA packet and then direct the packets to the nearest gateway where the packet is routed. Both implementations use the plug-in to append the additional destination address to the existing data packet as an AUXILIARY HEADER FIELD containing the destination address information. This implementation is non-transparent

and requires the co-operation of internet applications and devices to incorporate the plug-in in order to retrieve the additional destination address. For example, RFC2616 (HTTP Protocol), Sections 7 to 7.1 describe the use of ENTITY HEADER FIELDS. New Entity Header fields can be defined as additional destination address field for Vast Domain Technology in the event that a plug-in system is deployed instead of the system which detects implicit additional destination addresses.

The method is not application-specific, as it creates a sub-protocol that allows devices to be connected and identified within the internet without having a default IP address.

Modifications within the spirit and scope of the invention may be readily effected by those skilled in the art. It is to be understood therefore that this invention is not limited to the particular embodiments described by way of example hereinabove.



THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A method of directing data from a source device to a destination device over a computer network, comprising:

forming a data packet including said data, conventional address data and additional address data;

extracting from said data packet said additional address data; and

transmitting said data packet to said destination device as indicated either by said additional address data, or by said conventional address data and said additional address data.

2. A method as claimed in claim 1, wherein said additional address data is accommodated in said data packet in a field provided for other data, whereby said other data is contained with said additional address data in said field.

3. A method as claimed in either claim 1 or 2, wherein said additional address data is included in a field provided for said data or said conventional address data.

4. A method as claimed in any one of the preceding claims 2, wherein said computer network is the internet, and said address data is internet protocol address data.

5. A method as claimed in any one of the preceding claims, including forming said data packet over an existing, conventional data packet through existing standards.

6. A method as claimed in any one of the preceding claims, wherein said additional address data is indicative of whether said address data is adequate to uniquely identify the destination address and, if so, said address of the destination device is indicated by both the address data and the additional address data.

7. A method as claimed in any one of the preceding claims, wherein said address data includes either source address data or destination address data, or both source address data and destination address data.

8. A method as claimed in claim 7, wherein said additional address data includes the additional destination address data.

9. A method as claimed in any one of the preceding claims, wherein said forming said data packet to include additional address data is performed by a gateway means.

10. A method as claimed in claim 9, wherein said method includes recognizing and routing said data packet by means of said gateway.

11. A method as claimed in either claim 9 or 10, wherein said address data corresponds to said gateway means.

12. A method as claimed in claim 11, wherein said destination address data corresponds to said gateway means and said source address data corresponds to a further gateway means.

13. A method of directing data from a source device to a destination device over a computer network, comprising:

forming a data packet including said data and address data, said address data resolvable into conventional address data and additional address data;

transmitting said data packet to said destination device as indicated either by said additional address data, or by said conventional address data and said additional address data.

14. An apparatus for directing data from a source device to a destination device over a computer network,

comprising:

means for forming a data packet including said data, conventional address data and additional address data;

means for transmitting said data packet; and

means for extracting from said data packet said additional address data, and transmitting said data packet to said destination device as indicated either by said additional address data, or by said address data and said additional address data.