System and Method for Maintaining a Network Connection

Field Of The Invention

[0001] The present application relates generally to computer networking and more particularly to a system and method for maintaining a network connection.

Background Of The Invention

[0002] In certain network connections, such as connections made over the Hypertext Transfer Protocol ("HTTP"), it can be desired to maintain a persistent connection between the client and the web-server in order to reduce the overhead needed to reestablish the connection. However, Network Address Translation ("NAT") gateways and other equipment that lie along the connection pathway may terminate the connection in the event that the connection goes idle beyond a predefined period of time.

[0003] In order to prevent NAT gateway from terminating the connection, it is known to periodically send "keep-alive" packets from the client to the web-server. Such keep-alive packets do not actually include any transactional information and have no effect of the state of the data between the client and the web-server, and are merely used to prevent the NAT gateway from terminating the connection. It is typical to aggressively send keep-alive packets, without any regard to the actual parameters used by the NAT gateway, and thereby implement a universal strategy to keep the connection open.

[0004] However, these prior art methods of maintaining persistent connections are ideally suited to channels where bandwidth is not constrained. Thus, in bandwidth constrained mediums, such as wireless network channels, this strategy is wasteful of precious bandwidth. This prior art method is also undesirable in battery operated devices, where aggressive delivery of keep-alive packets could quickly drain the battery.

Summary of the Invention

[0005] It is an object to provide a novel a system and method for maintaining a network connection that obviates or mitigates at least one of the above-identified disadvantages of the prior art.

[0006] An aspect of the invention provides a system comprising a first electronic device including a first network interface for participating in a network connection. The system also includes a physical link connected to the first network interface. The physical link is for carrying the network connection. The link includes equipment for terminating the connection if the connection remains idle according to a predefined time-out criteria.

[0007] The system also includes a second electronic device that includes a second network interface for participating in the network connection via the link. The second electronic device is operable to determine the predefined time-out criteria, and send keep-alive signals to the first electronic device within the parameters of the time-out criteria.

[0008] The type of time-out criteria that is determined by the second electronic device is not particularly limited. For example, commonly that the time-out criteria is simply a time period of inactivity over the physical link. Another, less common time-out criteria can be in a specially configured Network Address Translation device that located on the physical link that is configured to only time-out the connection if the NAT device requires the resources. Another time-out criteria that is somewhat more common is where there is a non-idle timeout. In this system the NAT device can choose to terminate any connection that is active for more than a predefined period. The second electronic device can be configured to determine which of these criteria (or any other criteria) is being employed along the physical link and respond with a delivery of keep-alive signals so as to reduce the likelihood of the connection being terminated.

[0009] An aspect of the invention provides an electronic device comprising a microcomputer and a network interface for establishing a network connection with a second electronic device over a physical link. The physical link includes equipment with a timeout period that terminates the connection when the connection is idle. The microcomputer is operable to send keep-alive signals to the second electronic device according to an iteratively

changing criteria to establish the timeout period. Thereafter, the microcomputer sends the keep-alive signals within the timeout period and thereby reduces the likelihood of the equipment dropping the connection due to idleness.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The invention will now be described by way of example only, and with reference to the accompanying drawings, in which:

Figure 1 is a schematic representation of a system for maintaining a network connection in accordance with an embodiment of the invention;

Figure 2 is a flow chart depicting a method of maintaining a network connection in accordance with another embodiment of the invention;

Figure 3 shows the system of Figure 1 during the performance of the method in Figure 2;

Figure 4 shows set of sub-steps for performing one of the steps in the method of Figure 2; and,

Figure 5 shows the system of Figure 1 during the performance of the method in Figure 2.

DETAILED DESCRIPTION OF THE INVENTION

[0011] Referring now to Figure 1, a system for maintaining a persistent network connection is indicated generally at 30. In a present embodiment, system 30 includes at least one client 34 that connects to a service provider node 38 via a wireless link 42. Node 38 includes a wireless base station 46 that interacts with client 34 via link 42 and a NAT gateway 50. In turn, gateway 50 connects to the Internet 54 via a backhaul 58. Backhaul 58 can be a T1, T3 or any other suitable link for connecting node 38 to Internet 54. Internet 54, itself, connects to a web-server 62 via a second backhaul 66.

[0012] In a present embodiment, client 34 is a battery operated device that is based on the computing environment and functionality of a wireless personal digital assistant. It is, however, to be understood that client 34 need not be battery operated and/or can include the construction and functionality of other electronic devices, such as cell phones, smart telephones, desktop computers or laptops with wireless 802.11 or bluetooth capabilities or the like.

[0013] It is also to be understood that, at least a portion of the connection between client 34 and web-server 62 is bandwidth-constrained. In system 30, since link 42 is a wireless connection that may need to serve a plurality of clients 34, then link 42 is bandwidth constrained in relation to backhaul 58, backhaul 66 and the other elements that compose the connection between client 34 and web-server 62. Such bandwidth constraints can thus interfere with the speed with which a user operating clients 34 can access Internet 54 and web-server 62. Such constraints become particularly acute when a plurality of clients wish to access link 42. Furthermore, judicious use of link 42 by client 34 is desirable due to the fact that client 34 is battery operated.

[0014] NAT gateway 50 is based on standard NAT technology and thus allows a multiple number of clients 34 connected to node 38 to connect to Internet 54 though a public Internet Protocol ("IP") address assigned to NAT gateway 50. Accordingly, client 34 (and other clients connected to node 38) will typically have a private IP address, while NAT gateway 50 will have a public IP address accessible to any party on Internet 54. Thus, as client 34 accesses Internet 54, web-server 62 will communicate with client 34 via gateway 50, with gateway 50 "translating" IP addresses during such communication. In an example unique to the present embodiment, client 34 has the private IP address "10.0.0.2", gateway has the private IP address 10.0.0.1 and the public IP address of "50.0.0.1" and web-server has the public IP address "62.0.0.1".

[0015] Like existing NAT gateways, gateway 50 is thus also configured to automatically terminate idle connections between client 34 and Internet 54 in order to free-up resources for NAT gateway 50. Client 34 is configured to maintain a connection between client 34 and web-server 62 notwithstanding the automatic termination feature of gateway 50.

More particularly, client 34 is configured to send keep-alive packets during an idle communication period to web-server 50 according to a variable criteria, such keep-alive packets being intended to prevent gateway 50 from dropping the connection between client 34 and web-server 50, but without changing the state of data in client 34 or web-server 62. Such keep-alive packets can be any suitable packet, that achieves this result, such as a "no-op" command, a command that generates a non critical error result in the server or a command designed into the application level protocol as a keep alive mechanism. In a present embodiment, the variable criteria is based on an time period that is arrived upon iteratively. The iterations are considered complete when a time period is established that is substantially close to the maximum amount of time that NAT gateway 50 will allow to lapse before terminating the connection between client 34 and web-server 50. Further understanding about client 34 and this criteria will provided below.

[0016] In order to help explain certain of these implementations and various other aspects of system 30, reference will now be made to Figure 2 which shows a method for maintaining a network connection and which is indicated generally at 400. In order to assist in the explanation of the method, it will be assumed that method 400 is operated by client 34 using system 30. However, it is to be understood that client 34, system 30 and/or method 400 can be varied, and need not work exactly as discussed herein in conjunction with each other, and that such variations are within the scope of the invention.

[0017] Before discussing method 400, it will be assumed that NAT gateway 50 is configured to drop connections where a connection is idle for greater than fifteen minutes (However, other time periods are also within the scope of the invention, according to the configuration of the particular NAT gateway. Such other time periods can be greater than twenty minutes, or greater than thirty minutes, or greater than ten minutes.) It will also be assumed that this timeout period is unknown to client 34 on invocation of method 400.

[0018] Beginning first at step 410, a set of default criteria is loaded. As will be discussed below, the default criteria that is loaded is used by client 34 to define an initial time period during which keep-alive packets that are sent by client 34 in order to prevent gateway 50 from dropping a connection between client 34 and an entity on Internet 54, but without

changing the state of data in client 34 or that entity. In the present example, it will be assumed that the default criteria that is loaded will be a period of five minutes. (Other example default periods can be seven minutes, ten minutes and twelve minutes.)

[0019] Next, at step 420, a connection is established. Continuing with the present example, it will be assumed that client 34 opens a connection with web-server 62. This example is represented in Figure 3, wherein a connection is represented by a dotted line indicated generally at 100. The connection is opened in the usual manner, such as by having a web-browser on client 34 open an HTTP web-page that is located on web-server 34. The establishment of connection 100 thus involves having NAT gateway 50 create a mapping of client 34's private IP address to gateway 50's own public IP address. This is represented in Figure 4 by having gateway 50 represent to web-server 62 that the public IP address of client 34 is "50.0.0.1/8", wherein "50.0.0.1" is gateway 50's own public IP address, while "/8" represents the individual port on gateway 50 that is mapped to client 34's private IP address of "10.0.0.2". Thus, traffic carried over connection 100 will be passed through gateway 50 using this mapping. Once connection 100 is opened, network traffic is sent thereover in the usual manner. In general, it is to be reemphasized that this is merely an example and the way in which a connection is established is not particularly limited.

[0020] Next, at step 430, keep-alive signals are sent according to the established criteria. Since the criteria that was established at step 410 is a period of five minutes, then at step 430 keep-alive signals will be sent from client 34 to web-server 62 every five minutes. Since these keep-alive signals pass through gateway 50, then gateway 50 will only perceive that connection 100 is idle for five minute periods. Since this five minute period is less than the fifteen minute timeout period that gateway 50 will await before terminating connection 100, then gateway 50 will not terminate connection 100 and thus connection 100 will be persistent.

[0021] Method 400 will then advance to step 440, at which point a determination is made as to whether the connection established at step 420 has been terminated. Since the five minute interval during which client 34 sends keep-alive signals to web-server 62 is less than the previously mentioned fifteen minute timeout period, connection 100 will not be

terminated and so it will be determined at step 440 that "no", connection 100 as not been terminated and method 400 will advance to step 450.

[0022] At step 450, an adjustment, if any, to the criteria used at step 430 will determined. In a present embodiment, step 450 is carried out over a number of sub-steps, indicated generally at 450 on Figure 4. At step 451, it is determined whether the connection has ever been terminated. If there has been a prior termination, then the method advances to step 452 and the last known good criteria is maintained, and thus no adjustment is made to the criteria. At this point the method returns to step 430 on Figure 2.

[0023] However, if at step 451 it is determined that there has been no prior termination of the connection, then the method advances to step 453 and an adjustment is made to increase the time between delivery of the keep-alive signals. Thus, in the example being discussed herein in relation to connection 100, it will be determined at step 451 that connection has never been terminated, and the method will advance from step 451 to step 4523. At step 453, the criteria will be adjusted to increase the amount of time between delivery of keep-alive signals. The amount and/or rate by which the increase is made at step 453 is not particularly limited. In accordance with the present example, it will be assumed that the time interval will be increased by one-minute each time method 400 advances to step 453. Accordingly, during this cycle through method 400, the time period will be increased to six minutes from five minutes.

[0024] The method then advances from step 453 back to step 430, at which point the keep-alive signals are sent according to the criteria that has been established at step 453. Since the criteria that was established at step 453 is a period of six minutes, then at step 430 keep-alive signals will be sent from client 34 to web-server 62 every six minutes. Since these keep-alive signals pass through gateway 50, then gateway 50 will only perceive that connection 100 is idle for six minute periods. Since this six minute period is less than the fifteen minute timeout period that gateway 50 will await before terminating connection 100, then gateway 50 will not terminate connection 100 and thus connection 100 will be persistent.

[0025] Method 400 will thus continue cycle through steps 430, 440 and 450 (i.e. sub-steps 451 and 453) as previously mentioned until the criteria established at step 453 finally

adjusts the time interval beyond the timeout period of gateway 50. More specifically, once at step 453 a time period of sixteen minutes is established, then during the next cycle through step 430 the keep-alive signal will be sent outside the fifteen minute time-out period, and thus connection 100 will be terminated.

[0026] This time, when method 400 reaches step 440, it will be determined that connection 100 has been terminated, and thus method 400 will advances from step 440 to step 460, at which point the last-known good criteria will be loaded. In the present example, the last-known good criteria that was established previously at step 453 will be the time interval of fifteen minutes, and thus at step 460, in this example, client 34 will load the time period of fifteen minutes as the criteria.

[0027] Method 400 then advances from step 460 to step 420 at which point the connection is established (i.e. re-established). Continuing with the present example, it will be assumed that client 34 reopens a connection with web-server 62. This example is represented in Figure 5, wherein a new connection is represented by a dotted line indicated generally at 104. The connection is opened in the usual manner, such as by having a web-browser on client 34 open an HTTP web-page that is located on web-server 34. The establishment of connection 100 thus involves having NAT gateway 50 create a mapping of client 34's private IP address to gateway 50's own public IP address. This is represented in Figure 4 by having gateway 50 represent to web-server 62 that the public IP address of client 34 is "50.0.0.1/9", wherein "50.0.0.1" is gateway 50's own public IP address, while "/9" represents the individual port on gateway 50 that is mapped to client 34's private IP address of "10.0.0.2". Thus, traffic carried over connection 104 will be passed through gateway 50 using this mapping. Once connection 104 is opened, network traffic is sent thereover in the usual manner.

[0028] Method 400 then advances to step 430, at which point the keep-alive signals are sent according to the criteria that has been established at step 460. Since the criteria that was established at step 460 is a period of fifteen minutes, then at step 430 keep-alive signals will be sent from client 34 to web-server 62 every fifteen minutes. Since these keep-alive signals pass through gateway 50, then gateway 50 will only perceive that connection 104 is

idle for fifteen minute periods. Since this fifteen minute period is acceptable according to the fifteen minute timeout period that gateway 50 will await before terminating connection 104, then gateway 50 will not terminate connection 100 and thus connection 104 will be persistent.

[0029] Method 400 will then advance from step 430 to step 440, at which point a determination is made as to whether the connection established at step 420 has been terminated. Since the fifteen minute interval during which client 34 sends keep-alive signals to web-server 62 is acceptable in relation to the fifteen minute timeout period, connection 104 will not be terminated and so it will be determined at step 440 that "no", connection 104 as not been terminated and method 400 will advance to step 450.

[0030] At step 450, an adjustment, if any, to the criteria used at step 430 will determined. Recall that in a present embodiment step 450 is carried out over a number of sub-steps, indicated generally at 450 on Figure 4. At step 451, it is determined whether the connection has ever been terminated. Since the connection between client 34 and web-server 62 has been terminated once already (i.e. since connection 100 was terminated), then the method advances to step 452 and the last known good criteria is maintained, and thus no adjustment is made to the criteria. More specifically, since it is known that the fifteen minute time interval is an acceptable criteria, this criteria is maintained and at this point the method returns to step 430 on Figure 2.

[0031] Back at step 430, at which point the keep-alive signals are thus sent according to the criteria preserved at step 452. Since the criteria that was established at step 452 is a period of fifteen minutes, then at step 430 keep-alive signals will be sent from client 34 to web-server 62 every fifteen minutes. Since these keep-alive signals pass through gateway 50, then gateway 50 will only perceive that connection 104 is idle for fifteen minute periods, within the accepted fifteen minute timeout period that gateway 50 will await before terminating connection 100. Thus gateway 50 will not terminate connection 104 and thus connection 104 will be persistent. Method 400 will thus continue cycle as long as needed to maintain a connection between client 34 and web-server 62 during idle period.

[0032] It should now be apparent that a change in routing of connection 104 (or other change in the physical link between client 34 and web-server 62) could cause the timeout

period to change -- i.e. decrease over time from the time period that had been previously established through earlier cycles through method 400. For example, if another router in Internet 54 is introduced into the pathway that carries connection 104, and where that router drops idle connections after ten minutes, then method 400 may at various times cycle through step 460 and thereby the connection between client 34 and web-server 62 may be torn down and reestablished several times until at step 460 the criteria is decreased back to a ten minute interval. It is thus contemplated that step 460 can include sub-steps that will continue to decrease the criteria to shorter and shorter time periods until the shortest timeout period for any equipment along the physical link between client 34 and web-server 62 is established, at which point that shortest timeout period will be used at step 430. In this manner, it is contemplated that the criteria used at step 430 may at various times decrease or increase according to the timeout behaviours of the equipment that forms the physical link between client 34 and web-server 62.

[0033] Also, it is to be understood that, in other embodiments of the invention, normal spurious connection timeouts can be handled by a suitably modified version of method 400. Such a modified version of method 400 can be configured to respond to such spurious connection timeouts. For example a form of weighting or hysteresis can be utilized in a suitably modified version of method 400 that favours time-intervals for delivering keep-alive signals that client 34 has previously found effective in reducing the likelihood connection 104 being terminated.

[0034] It should also be understood that the rates by which the criteria is adjusted at step 450 and step 460 is not particularly limited. Further, the type of criteria that is used need not be particularly limited. For example the changes in criteria at step 450 and 460 need not be in a linear fashion, and need not be based on simple minute-by-minute increments or decrements. For example, a logarithmic convergence, based on splitting the various time intervals in half, using Newton's Method can be used. As an additional example, it can be desired at steps 450 and step 460 to consider the remaining battery life of client 34, and thus where the battery of client 34 has a long period of remaining life, the criteria adjustment made at step 450 need not be as aggressive. However, where the battery of client 34 has a short period of remaining life, the criteria adjustment made at step 450 may aggressively attempt to

have the criteria reach the idle timeout period as fast as possible in order to preserve the battery life of client 34.

[0035] While only specific combinations of the various features and components of the invention have been discussed herein, it will be apparent to those of skill in the art that desired subsets of the disclosed features and components and/or alternative combinations of these features and components can be utilized, as desired. For example, while not necessary, it is typically contemplated that steps 430-460 are only invoked during time periods that client 34 is aware that connection 100 (or connection 104) is idle, and so method 400 can be modified to cause steps 430-460 to be performed only during those time periods when connection 100 (or connection 104) is idle.

[0036] Furthermore, it is also to be understood that the origin of the keep-alive packets need not be restricted to client 34. For example, where base station 46 is aware of the need to maintain connection 100 as persistent, then it can be desired to have base station 46 perform steps 430-460 on behalf of client 34 and thereby free up resources on client 34 and link 42. By the same token, it is contemplated that steps 430-460 could also be conducted by web-server 62 on behalf of client 34.

[0037] In another variation of the invention, it is contemplated that steps 430-460 can be performed by client 34 prior to the establishment of a connection, and thereby determine the appropriate criteria for sending keep-alive signals within the timeout period prior to establishment of the connection and thereby reduce likelihood of termination of the connection. Further, once this timeout period is established, it is contemplated that the period can be reported to other clients attached to node 38, thereby obviating the need for those clients to perform steps 430-460 themselves.

[0038] While system 30 is directed to a specific type of network, it should be understood that other types of clients, servers, and networks can be used. For example, the invention can be applied to peer-to-peer connections and need not be limited to client/server type relationships. Furthermore the type of physical connections that carry the connection are not limited, and can be based on Ethernet, Intranets, 802.11, bluetooth etc. Additionally, while the embodiments herein are discussed in relation to connections over which at least a

portion are bandwidth constrained, it should be understood that the invention is also applicable to connections that are not bandwidth constrained.

[0039] The above-described embodiments of the invention are intended to be examples and alterations and modifications may be effected thereto, by those of skill in the art, without departing from the scope of the invention which is defined solely by the claims appended hereto.

CLAIMS

1. An electronic device including a network interface for participating in a network connection with a second device via a network connection carried over physical link that includes equipment for terminating said connection if said connection remains idle according a predefined time-out criteria, said device operable to determine said predefined time-out criteria.

2. The electronic device of claim 1 wherein said electronic device is further operable to send keep-alive signals according to said determined criteria in order to reduce dropped connections by said equipment and reducing overall traffic carried over said link.

3. The device according to claim 2 wherein said connection is an HTTP web-page being requested by said first electronic device of said second electronic device and said keep-alive signal is a no-op signal.

4. The electronic device of claim 1 wherein said equipment is a NAT router.

5. The electronic device of claim 1 wherein said criteria is a predefined time period.

6. The electronic device of claim 5 wherein said device determines said predefined time period by:

establishing said connection with an initial default time period;

sending a keep-alive signal to said second device once during said time period;

increasing said time period if said time period does not cause said connection to be dropped then repeating said sending step; and

maintaining a last-known good time period if said time period does cause said connection to be dropped and then reestablishing said connection and returning to said sending step;

during which said device sends keep alive signals to said second electronic device, and varying said time period sending keep-alive signals over said connection during said time period for each iteration until said time period causes said equipment to terminate said connection.

7. The device according to claim 6 wherein said device is a client, said second device is a web-server and at least a portion of said link includes the Internet.

8. The device according to claim 7 wherein said client is battery operated and said time periods are increased more quickly as said battery life is depleted to thereby reduce battery consumption while determining said predefined time period.

9. The device according to claim 8 wherein said client is a wireless device and at least a portion of said link includes a wireless connection from said wireless device to the Internet.

10. A method of maintaining a network connection comprising the steps of:

loading a timeout criteria into a first electronic device of an initial default value;

establishing a connection from said first electronic device to a second electronic device via a physical link that includes equipment for terminating said connection if said connection remains idle for a predefined timeout period;

sending keep-alive signals from one said electronic device to the other said electronic device via said equipment according to said timeout criteria;

increasing said timeout criteria and repeating said sending step; and,

repeating said increasing step until said connection is terminated by said equipment and thereafter performing said sending step using a known good timeout criteria.

11. The method according to claim 10 wherein said at least one least known timeout criteria is a last-known good timeout criteria.

12. The method according to claim 10 wherein said at least one least known timeout criteria is determined by iteratively decreasing said timeout criteria until said connection is no longer terminated.

13. The method according to claim 10 wherein said connection is an HTTP web-page being requested by said first electronic device of said second electronic device and said keep-alive signal is a no-op signal.

14. The method according to claim 10 wherein said equipment is a NAT router.

15. The method according to claim 10 wherein said first device is a client, said second device is a web-server and at least a portion of said link includes the Internet.

16. The method according to claim 10 wherein said client is battery operated and said increasing step is based on larger intervals when said battery life is approaching depletion.

17. The method according to claim 16 wherein said client is a wireless device and at least a portion of said link includes a wireless connection from said wireless device to the Internet.

18. A computer-readable storage medium containing a set of instructions for an electronic device the set of instructions comprising the steps of:

loading a timeout criteria into said electronic device of an initial default value;

establishing a connection from said electronic device to a second electronic device via a physical link that includes equipment for terminating said connection if said connection remains idle for a predefined timeout period;

sending keep-alive signals from said electronic device to said second electronic device according to said timeout criteria;

increasing said timeout criteria and repeating said sending step; and,

repeating said increasing step until said connection is terminated by said equipment and thereafter performing said sending step using a known good timeout criteria.