A New Heartbeat Processing Mechanism Design Based on Equipment Online Detection

Zhu Jin-Tian

Abstract: Heartbeat packets between the client and the server through one party regularly sent by the other party to make appropriate responses to judge each other connection is working. The traditional heartbeat packet processing mechanism has been widely used in all kinds of network communication, however, according to different needs, has their own shortcomings. Combined with the traditional heartbeat mechanism, a new heartbeat packet protocol based Socket is designed. This protocol aims to make the new heartbeat packet more flexible, real-time, reliable. Meanwhile, lower expenses flow and expenses also generally adapt to different network environments.

Keyword: Heartbeat packet, network communication, Socket

1 Introduction

With the deepening development of science and technology, the network has become an important part of human activities. In a network, detecting the connection state of interconnected computers accurately and implementing on-line monitoring of the server to ensure normal network communication is essential. And in fact most of these functions are implemented through the heartbeat packets.

Heartbeat packets are used to determine whether a node in the network is functioning properly. Heartbeat mechanism is achieved in this way: one sends a streamlined communication packet to the other regularly, the other reply after receipt of heartbeat packets. Heartbeat packets are sent to tell each other "I am alive" (still connected). If the other does not receive such a package at a certain time interval, the recipient believes that the sender has been disconnected, and then makes the corresponding disconnect logic processing.

Recently, heartbeat packet mechanism has been widely used in the field of network communications, and network fault detections. Improving based on the traditional heartbeat packet, new heartbeat mechanism can strength the reliability of devices online detection and reduce network traffic overhead.

2 The basis of heartbeat model design

2.1 Socket mechanism

Socket is in fact an abstract of a communication port, it provides a communication port actually which helps communicate with any computer with a socket interface. For every application program transmitting data on the network, the sending and receiving procedures are achieved through the Socket interface.

Socket supports the sending and receiving of data. Firstly, socket is created for sending data, and then methods of socket such as sendto is used for data transmission that aimed at a port ;Similarly, socket is created in receiving end firstly, and then the socket is bound to a port, all data sent to this port will be read out by the socket function such as recv. It is like to read out the data in the file.

In general, socket provided by Windows Socket can be classified into two kinds: stream sockets and data-gram sockets. Stream socket is connection-oriented, you must create a path first before data exchange between each other. Meanwhile, stream socket checks the data, includes the retransmission mechanism, and therefore provides reliable delivery. Data-gram socket also supports bidirectional data flow to establish a connection, but differently it can send data without guaranteeing to establish a connection. Data-gram socket does not check and retransmit, so the recipient may not be able to receive complete and correct data, but data-gram socket is more provincial traffic and its communication rate is higher.

2.2 Implementation and improvement of the traditional heartbeat package

There are three approaches in traditional heartbeat packet. The first one is: heartbeat packets are sent by the application itself to detect whether the connection is normal. Its approximate method is: the server sends a packet to the client regularly in the event of a Timer, and then starts a low-level thread, and tests the client's response constantly in this thread. The second one is: using a Keep-alive mechanism that TCP includes itself to detect dead connections. After turning on a Keep-alive function, one will send the heartbeat packets automatically to the other in a prescribed time, while the other will reply automatically to tell each other that they are still online after the receipt of heartbeat packets. This method corresponds to a stream socket. The third one is: heartbeat packets are implemented by the UDP protocol, its working process is similar to the above, and its method corresponds to data-gram sockets, and sends more short and pithy heartbeat message. The recipient judges that the other side has dropped if there is no packet to be received over a certain period of time. In order to safeguard two-way channels, heartbeat line detection implemented by UDP is not necessarily better than TCP in Provincial traffic. But one of its main advantages is that we can understand the network status through packet loss.

The implementation of three heartbeat mechanisms above can be said to have advantages and disadvantages, we will choose the most suitable way depending on the circumstances in an actual use. However, when the use environment changes, the original use of the heartbeat mechanism may be no longer appropriate and even cause some troubles. Therefore, it is necessary for us to design a higher flexibility heartbeat protocol which will be able to adapt to the new environment by simply reset some parameters provided by the agreement when the environment changes.

From the communication began to develop until today, we can say that the traditional heartbeat packet is still applicable to adapt to most environments, thus the heartbeat packets research and improvement shows rare. But in recent years, wireless networks are developing rapidly. It is more flexible than the wired network, but has more frequent data transmission errors because of signal interference, limited distance and so on. For wireless networks, in order to ensure the accurate detection of the connection status, it is best to adopt connection-oriented communication that ensures data reliable.

On the other hand, a wide range of wireless network is generally provided by the communication carriers, and keeping the mutual transmission of the heartbeat packet also means additional traffic will generate continually. So it is necessary for us to improve the heartbeat packet format in order to shorten the packet length which does not reduce the original function.

3 New Heartbeat Protocol

3.1 Device state design

In a heartbeat mechanism, three parameters are used. One is called Heartbeat timer. Every time this timer is reset, the device should send a heartbeat packet. It could be regarded as the cycle of heartbeat package transmission. Another one is called Response timer. It sets the waiting delay for reply, when is reset, the device should retransmit a heartbeat packet. The rest one is called Response Retry Number, sets the attempt limit.

The following aims at C/S network structure for discussion. The communication state of each client in the network can be divided into three conditions: disconnected, connecting, and connected. When the device is started and has physically connected to the network, the connection state turns from disconnected to connecting and try establishing the connection with the server by Socket. After the success of the network connection the client switches to connected state. The client in connected state communicates with the server according to the exchange of heartbeat packets on a regular basis (Heartbeat timer). If the client waits until Response timer is reset after sending one heartbeat packet, it is considered transmission failure. The client then turns its connection state to connecting and retransmission a heartbeat packet. If the retransmission time reaches the maximum value (Response Retry Number), the client considers that the communication path is broken and returned to disconnected state. The state switching process is shown in FIGURE 1.

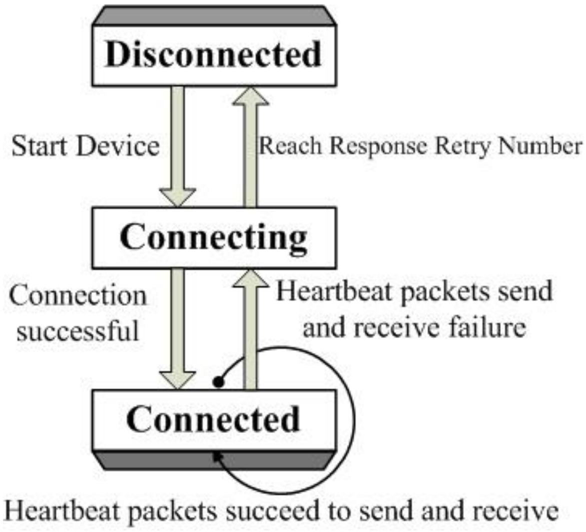

FIGURE1 Device communication states

3.2 The thinking of heartbeat mechanism design

Heartbeat packets can be sent either from the server to the client or from the client to the server, in which direction basically see which way is more convenient. For single server with multiple clients work mode, if the server is the active side, the most convenient way is unified sending of heartbeat packets to all devices at the same time. However, because each device's state has its particularity, the server should keep time for each client separately. This method is not efficient if the number of devices is high. Therefore, this protocol stipulates that clients be the active side to send heartbeat packets.

The protocol uses a brand new check mode. Basic thinking: The server and client respectively reserve a storage unit, “t” to say, which stores a value that increases gradually with time. The server and the client read each other's value in this storage unit, if the numerical increment of the other side is the same as or similar to its own, the connection is normal, and send check result (Y or N) to the other side.

3.3 Packet format design

The format of a heartbeat packet is: source port 2 bytes, destination port 2 bytes, and heartbeat packet data segment part 2 bytes. The front 15 bits of heartbeat packet data segment is used to store the current value of t when the packet is to be sent, and 0 in the last bit means N, while 1 means Y, as shown in FIGURE 2.

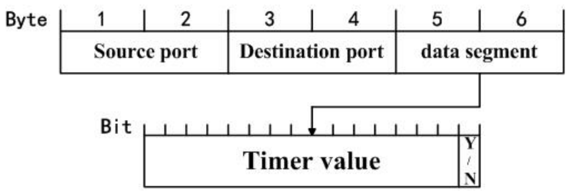

FIGURE 2 Format of a heartbeat packet

3.4 Parameters setting

3.4.1 Adjustable network parameters

e: e means the value in the data segment increases every $\frac{e}{1000}$ second;

n: Set the allowed error range, can be set according to the need of the precision of packets, minimum of 0;

others: Heartbeat timer, Response timer, Response Retry Number(all introduced above).

3.4.2 The storage space every node allocates

For a single client (Cx) in the network：

t: the storage unit client Cx reserved, and its value increases with time, as introduced above;

ts: A 2-byte space, stores t’s value when received the Y packet from the server last time, set the first bit 0;

Ts: A 2-byte space, stores the timer value in the Y packet received from the server last time, set the first bit 0.

For the server (S)：

T: the storage unit S reserved, and its value increases with time, as introduced above;

Tx: A 2-byte space, stores T’s value when received the Y packet from Cx last time, set the first bit 0;

tx: A 2-byte space, stores the timer value in the Y packet received from Cx last time, set the first bit 0.

3.5 The realization of the heartbeat model

After a client(Hereinafter referred to as Cx) has connected to the network, it initializes storage space ts and Ts (both set 0 first), and starts its timer t, then send a heartbeat packet to the server (Hereinafter referred to as S), the data segment of which is 00000000,00000001. After receiving, S allocates some storage space to store data used to communicate with Cx, they are tx and Tx (also both set 0 first), then reply a packet, the data segment of which records the current value of T and the last bit is 1. Cx receives reply and save the timer value of the packet into Ts, current value of t into ts. If Cx waits till Response timer is reset but still not yet received reply, then resets t to 0 and resend a heartbeat packet. Access to the server process is shown in FIGURE 3.

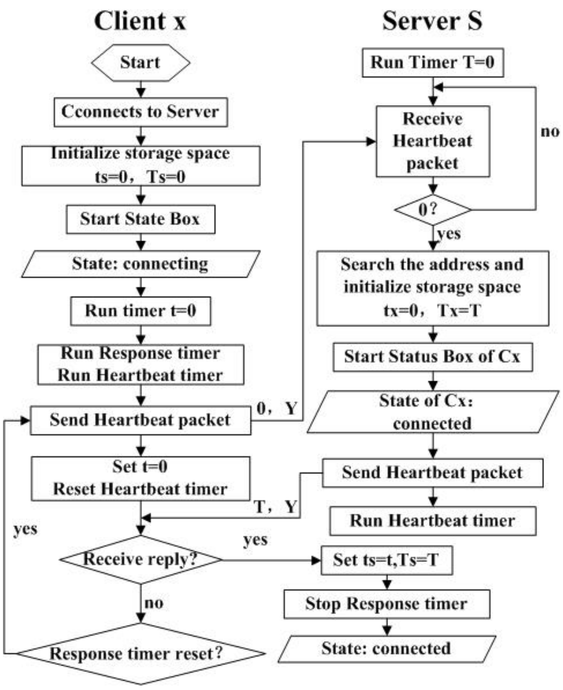

FIGURE 3 Access to the server

Under normal condition of the connection, Cx sends a heartbeat packet when Heartbeat timer is reset. The data segment of which records current value of t and the last bit is 1. S checks after receiving. The concrete method is: compare (t-tx) with (T-Tx) , t is the timer value in the packet. If Numerical growth in a reasonable range (within n), S considers the connection with Cx is fine and returns a heartbeat packet whose last bit is 1. Cx does the same check after receiving this packet. The server and the client both update their storage space each time after receiving a correct (through check) packet with last bit 1. Thus a “heartbeat” has finished. Because the timer will return to 0 after increasing to the largest value, the situation in which the difference is $32768_{(D)}$(1111111,$11111111_{(B)}$) after comparing must be taken into account when checking.

If a heartbeat packet delivery is not smooth, embodied in: the error is greater than n, or because a packet is damaged or lost in transit that one side fails to receive the packet from the other side within the time set, or one side receives a packet ends of 0, etc., then the side that found mistakes changes connection state and sends a packet with last bit 0. The opposite side receives the N packet and checks, if the result is correct, replies a packet with last bit 1, else, replies a packet with last bit 0 and repeats this process until the check result is correct. The specific process is as shown in FIGURE 4.

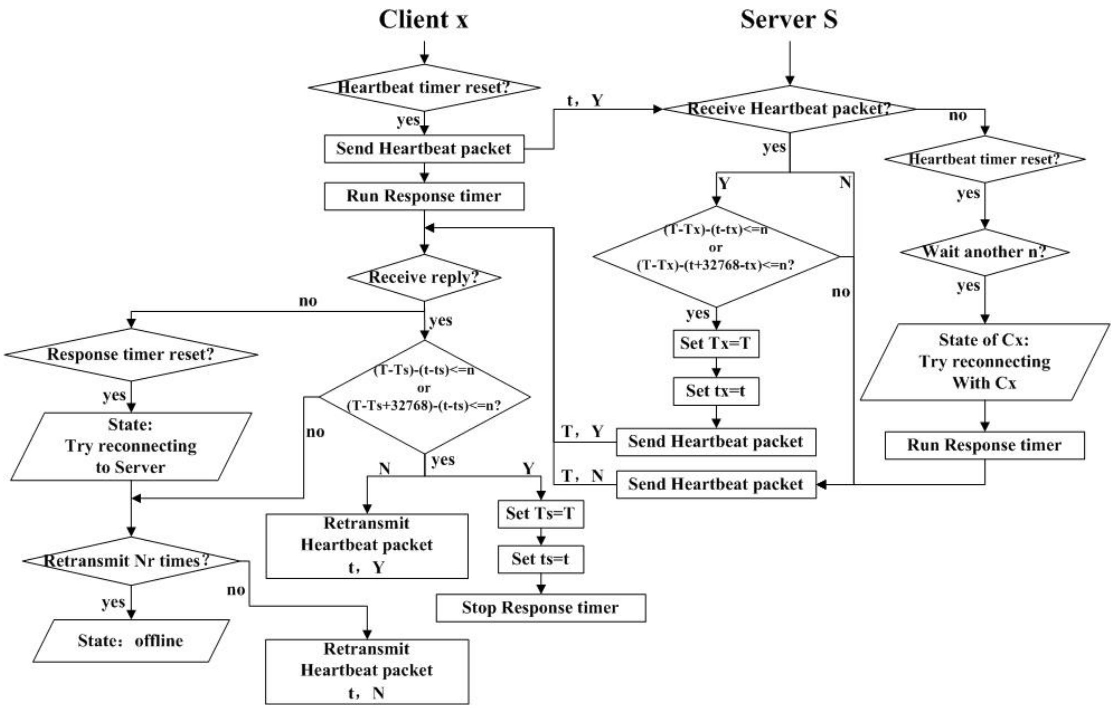

FIGURE 4 Regularly send heartbeat packets

If connection between Cx and S is broken, S can not receive a heartbeat packet after waiting until Heartbeat timer is reset and for another n, then changes the state of Cx to connecting and sends Cx a N packet. S repeats this for Response Retry Number times but still cannot receive a reply, then considers that Cx is offline, turns the state of Cx to disconnected releases the storage space used to store information of Cx. As shown in FIGURE 5.

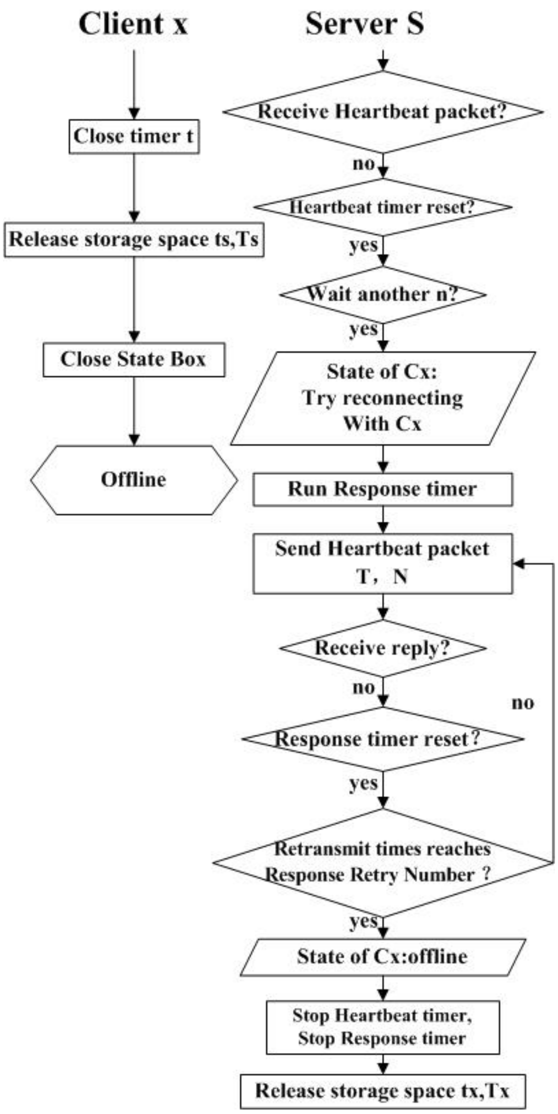

FIGURE 5 Disconnect from the server

3.6 The practical applications of the heartbeat protocol

Heartbeat protocol can be used in mobile phone card IOT attendance. At a certain time interval, the device will send heartbeat packets and the heartbeat packets can guarantee the device reconnects the PC which breaks abnormally. It has the power data protection that attendance data will not cause power loss. The data will be securely recorded into the system after successful attendance. Each takes intelligent terminals such as Wifi, GPRS, 3G, GPS module which is similar to a netbook. We can realize staff attendance in real-time with these terminals. After testing, the new heartbeat protocol can detect the online status of each attendance successfully.

4 Conclusion and discussion

Since general functions of the heartbeat packets are relatively simple, most of the heartbeat mechanism is achieved by UDP or TCP protocol. Thus works about improving heartbeat protocol are rare. However, with the development and popularization of the network, online detection of network equipment in a network is more and more important and traditional heartbeat mechanism are facing challenging.

The heartbeat protocol above not only guarantees the heart function in theory, but also ensures the reliability of the equipment line detection and shortens the length of the packet significantly. Meanwhile, packets provided by protocol adapt to the current protocol better in structure. Some adjustable parameters of the protocol has defined ,we can modify these parameters to suit most needs for the heartbeat protocol and enhance its flexibility.

REFERENCES

[1] Hailong Wang. A Heartbeat Protocol Based on the Real-Time Ethernet. [D].Dalian. Dalian University of Technology.

[2] Mohamed G. Gouda, and Tommy M. McGuire. Accelerated Heartbeat Protocols

[3] Junkai Yi, Li Chen, and Jianwei Sun. Data Flow Clustering Detection Approach of Network Heartbeat Packet Sequence. [J]. Computer Engineering. 2011.37(24):61-63.

[4] Yuanmou Chen, and Changqi Wu. Based on the Socket of Remote Monitoring

Communication System Implementation. [J]. Computer Security. 2010.(3):60-62.

[5] Fengshi Zhou. The Principle and Implementation of Heartbeat Packet based on Windows Socket Communication. [J].Journal of Shazhou Professional Institute of Technology. 2009.12(3):17-21.

[6] Zhikun Hu, et al. Remote monitoring system of rectifier based on improved heartbeat mechanism. [J]. Computer Applications. 2008.28（2）：363-366.

[7] Jin Tian. Performance Analysis on GPRS Data Packetization Size and Heartbeat- packets Interval. [J]. Telecommunication Engineering. Vol. 52, No. 4, Apr. 2012.

[8] Changsheng Xie, Qingping Hu, and Zhihu Tan. Heartbeat-gear：A New Real-Time Heartbeat Inspecting Technique. [J]. COMPUTER ENGINEERING & SCIENCE. Voi.26, No.5, 2004.

[9] Zonghao Hou, et al. Heartbeat Mechanism Supporting Load Balancing in Multi-Machine Environment. [J]. MINI-MICRO SYSTEMS. Voi.26, No.1, Jan.2005.

[10] Yutong Lu, Min Wang, and Nong Xiao. A New Heartbeat Mechanism for Large-Scale Cluster.