

FILE TRANSMISSION METHOD AND SYSTEM

FIELD OF THE TECHNOLOGY

[0001] The present disclosure relates to the field of network transmission technologies, and in particular, to a file transmission method and a file transmission system.

BACKGROUND OF THE DISCLOSURE

[0002] With the improvement of processing capabilities of mobile terminals, the mobile terminals are used as an entrance to the mobile Internet, and are gradually widely used, and further, various files are transmitted dependent on the mobile Internet. For example, when people use the mobile Internet, people have a more intense demand for file uploading and often expect to upload, by using a mobile terminal, some files to a receive end located in the mobile internet for storage, so as to facilitate others to share the uploaded file or back up and store the file.

[0003] However, due to small network bandwidth of the mobile Internet and instability caused by the fact that a channel is easy to be interfered with, for example, transmission speeds of current 2G and 3G networks are lower than that of a fixed broadband, but packet loss rates of the 2G and 3G networks are higher than that of the fixed broadband, such networks with a low transmission speed and a high packet loss rate are referred to as weak networks. It needs to take a lot of time and waste bandwidth of a mobile network to upload a file in a weak network environment.

[0004] In an existing weak network environment, a common solution of file uploading is described as follows: Step 1) A sender establishes a TCP or HTTP connection to a receiver. 2) Sequentially acquire a data block of a file and send the data block to the receiver, and add the currently sent data block to a timeout waiting queue. 3) The receiver determines whether data is valid, and if yes, reply that the data block is received successfully; otherwise, no reply is made or reply that the data block is invalid. 4) The sender performs step 2) if the sender receives the reply of the receiver and the reply indicating that the data block is valid, until the file is completely sent; otherwise, the sender re-sends the data block if the sender receives the reply, of the receiver, indicating that the data block is invalid or waiting for the reply times out (the sender performs timing and waits). A common solution in an existing weak network is to send file blocks in series, which cannot effectively use bandwidth, and leads to a long timeout waiting time, and finally leads to an excessively long file transmission time.

SUMMARY

[0005] Based on this, it is necessary to provide a file transmission method and system for the problem in the existing technology that in a weak network environment, bandwidth cannot be effectively used for file transmission, a timeout waiting time is long, and finally an excessively long file transmission time is caused, so as to accelerate file transmission by means of multi-connection and multi-window concurrency.

[0006] A file transmission method includes steps of: establishing at least two connections to a receive end, each connection including at least one send window; dividing a to-be-sent file into at least two data blocks, filling a corresponding data block in a send window of each connection, and sending the data block to the receive end by using the send window; acquiring acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive; and clearing, according to the acknowledgement information, a send window corresponding to the data block information received this time, filling, in the cleared send window, a data block that the receive end expects to receive, and sending the data block to the receive end by using the send window.

[0007] Correspondingly, the present disclosure further provides a file transmission system, including: a connection module, configured to establish at least two connections to a receive end, each connection including at least one send window; a dividing and sending module, configured to divide a to-be-sent file into at least two data blocks, fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window; an acknowledgement acquiring module, configured to acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive; and a sending module, configured to clear, according to the acknowledgement information, a send window corresponding to the data block information received this time, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

[0008] According to the present disclosure, at least two connections are established to a receive end, each connection including at least one send window, so as to form multi-connection and multi-window transmission. A to-be-sent file is divided into multiple data blocks, and the data blocks are distributed to different connections for parallel sending. A data block that the receive end expects to receive is filled in a cleared send window according to acknowledgement information of

the receive end, and the data block is sent to the receive end by using the send window, until all the data blocks are received by the receive end. There is no fixed mapping relationship between each connection and a file block, but for a connection with a send window that is cleared first, a data block sent next is filled in the send window, that is, for a connection with a higher transmission speed, more data blocks are distributed to the connection; therefore, an overall transmission time of the to-be-sent file can be effectively reduced. Distribution of a data block depends on an idle degree and a speed of a connection, especially in a case in which an individual connection is stuck due to poor transmission quality or a low transmission speed of the connection, and according to the present disclosure, a data block can be distributed to a connection with a high transmission speed, which avoids an overall increase of a consumed time caused by a stuck individual connection. According to the present disclosure, utilization of bandwidth is improved and a file transmission time is effectively reduced in a connection manner not limited to TCP, UDP, and HTTP by means of multi-connection and multi-window concurrency and proper data block distribution and filling.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 is a flowchart of a file transmission method according to the present disclosure;

[0010] FIG. 2 is a flowchart of a first embodiment of a file transmission method according to the present disclosure;

[0011] FIG. 3 is a flowchart of a second embodiment of a file transmission method according to the present disclosure;

[0012] FIG. 4 is a flowchart of a third embodiment of a file transmission method according to the present disclosure;

[0013] FIG. 5 is a schematic diagram of a file transmission process of a fourth embodiment of a file transmission method according to the present disclosure;

[0014] FIG. 6 is a schematic diagram of a data block transmission process in a connection at a given time in FIG. 5;

[0015] FIG. 7 is a structural block diagram of a file transmission system according to the present disclosure;

[0016] FIG. 8 is a structural block diagram of an embodiment of a file transmission system according to the present disclosure; and

[0017] FIG. 9 is a diagram of modules of a computer system 1000 that can implement the

embodiments of the present invention.

DESCRIPTION OF EMBODIMENTS

[0018] To make the objective, technical solutions, and advantages of the present disclosure clearer, the following further describes the present disclosure in detail with reference to the accompanying drawings.

[0019] Refer to FIG. 1, which is a flowchart of a file transmission method according to the present disclosure. The file transmission method includes steps of:

[0020] S101: Establish at least two connections to a receive end, each connection including at least one send window.

[0021] Connections between a transmit end and the receive end are established, the number of the connections is at least two, and each connection includes at least one send window. According to development complexity and a yield ratio, preferably, the number of the connections is not greater than 3, and the number of send windows may be as large as possible. If the receive end is a server, further, the connections may be at least two connections between the transmit end and one server, and may also be connections to at least two servers, and one connection exists between each server and the transmit end.

[0022] Data can be sent in parallel by establishing multiple connections, so that in a weak network environment, a bandwidth resource is fully used and transmission of a to-be-sent file is accelerated.

[0023] The send window is a sending tool of the connection, and is used for carrying to-be-sent data and sending a data block to the receive end.

[0024] S102: Divide a to-be-sent file into at least two data blocks, fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window.

[0025] The to-be-sent file is divided into at least two data blocks, different data blocks are distributed to each connection, a data block is filled in the send window of each connection, and one data block is filled in each send window. The data blocks are sent to the receive end by using the send windows, so that the data blocks are sent in parallel to reduce a file transmission time.

[0026] S103: Acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0027] After successfully receiving the data block, the receive end returns the acknowledgement information to the transmit end, where the acknowledgement information includes the data block information received this time and the data block information that the receive end expects to receive. The data block information received this time may be sequence number information of a data block received this time by the receive end, and the data block information that the receive end expects to receive may be sequence number information of a data block that the receive end expects to receive.

[0028] The transmit end can clearly know, from the acknowledgement information, the data block information and the data block information that the receive end expects to receive, which are received successfully, and the transmit end sends a corresponding data block to the receive end according to the information.

[0029] S104: Clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

[0030] The data block information received this time in the acknowledgement information clearly indicates, to the transmit end, which data block has been received by the receive end; therefore, the send window corresponding to the data block information received this time is cleared; and as long as there is a connection with a cleared send window, the data block that the receive end expects to receive is filled in the cleared send window, and the data block is sent to the receive end by using the send window.

[0031] According to the present disclosure, at least two connections are established to a receive end, each connection including at least one send window, so as to form multi-connection and multi-window transmission. A to-be-sent file is divided into at least two data blocks, and the data blocks are distributed to different connections for parallel sending. A data block that the receive end expects to receive is filled in a cleared send window according to acknowledgement information of the receive end, and the data block is sent to the receive end by using the send window, until all the data blocks are received by the receive end. There is no fixed mapping relationship between each connection and a file block, but for a connection with a send window that is cleared first, a data block sent next is filled in the send window, that is, for a connection with a higher transmission speed, more data blocks are distributed to the connection; therefore, an overall transmission time of the to-be-sent file can be effectively reduced. Distribution of a data block depends on an idle degree and a speed of a connection, especially in a case in which an individual

connection is stuck due to poor transmission quality or a low transmission speed of the connection, and according to the present disclosure, a data block can be distributed to a connection with a high transmission speed, which avoids an overall increase of a consumed time caused by a stuck individual connection. According to the present disclosure, utilization of bandwidth is improved and a file transmission time is effectively reduced in a connection manner not limited to TCP, UDP, and HTTP by means of multi-connection and multi-window concurrency and proper data block distribution and filling.

[0032] Refer to FIG. 2, which is a flowchart of a first embodiment of a file transmission method according to the present disclosure.

[0033] S201: Establish at least two connections to a receive end, each connection including at least one send window.

[0034] S202: Divide a to-be-sent file into at least two data units, where each data block includes several consecutive data units.

[0035] When each connection is used the first time to send a data block, each data block includes at least two data units if a size of the to-be-sent file is less than a preset value; otherwise, each data block includes one data unit.

[0036] The to-be-sent file is first divided into data units of the smallest unit, and then, several consecutive data units are combined into a data block. The number of data units included in each data block is determined by a transmission speed of each connection; therefore, sizes of data blocks sent by all the connections may be different, and sizes of data blocks sent by a same connection at different times may also be different, but a size of a data block cannot be greater than a maximum value of the size of the data block.

[0037] Preferably, when a data block is sent initially, if a size of the to-be-sent file is less than 100 K, one data block includes 4 data units, or if a size of the to-be-sent file is greater than or equal to 100 K, one data block includes one data unit combination.

[0038] Preferably, a data unit and a maximum value of a size of a data block are set as follows:

in a 2G network, a size of a data unit is 8 KB, and a maximum value of a size of a data block is 64 KB;

in a 3G network, a size of a data unit is 32 KB, and a maximum value of a size of a data block is 512 KB; and

in 4G and wifi networks, a size of a data unit is 64 KB, and a maximum value of a size of a data block is 1024 KB.

[0039] S203: Fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window.

[0040] S204: Acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0041] S205: Clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

[0042] The to-be-sent file is first divided into data units of the smallest unit, and then, several consecutive data units are combined into a data block. The number of data units included in each data block is determined by a transmission speed of each connection; therefore, sizes of data blocks sent by all the connections may be different, and sizes of data blocks sent by a same connection at different times may also be different; therefore, a data block sent by using a connection with a higher transmission speed is larger, so that a connection with a higher transmission speed can transmit a larger amount of data, thereby reducing a transmission time.

[0043] Refer to FIG. 3, which is a flowchart of a second embodiment of a file transmission method according to the present disclosure.

[0044] S301: Establish at least two connections to a receive end, each connection including at least one send window.

[0045] S302: Divide a to-be-sent file into data units, where each data block includes several consecutive data units.

[0046] S303: Separately acquire a current transmission speed and an average transmission speed of each connection, and increase, if the current transmission speed is not less than the average transmission speed, a size of a data block that is currently sent by using the connection.

[0047] A size of a data block to be sent by each connection is adjusted according to a different transmission speed of each connection. First, the current transmission speed and the average transmission speed of each connection are separately acquired. Preferably, the average transmission speed is an average speed of sending previous two data blocks by using the connection.

[0048] Separately for each connection, the current transmission speed of each connection is compared with the average speed of sending previous two data blocks by using the connection, if the current transmission speed is not less than the average speed of previous two data blocks, the size of the data block that is currently sent by using the connection is increased; and preferably, the size of the data block that is currently sent is increased to twice the original size of the data block, but the size of the data block cannot be greater than a maximum value of the size of the data block.

[0049] Preferably, a data unit and a maximum value of a size of a data block are set as follows:

in a 2G network, a maximum value of a size of a data block is 64 KB; in a 3G network, a maximum value of a size of a data block is 512 KB; and

in 4G and wifi networks, a maximum value of a size of a data block is 1024 KB.

[0050] S304: Decrease, if the current transmission speed is less than the average transmission speed and a decreasing proportion is greater than a fluctuation coefficient preset by a system, the size of the data block that is currently sent by using the connection.

[0051] For example, if the current transmission speed is less than the average speed of previous two data blocks, the decreasing proportion is continued to be compared with the fluctuation coefficient preset by the system to determine whether the decreasing proportion is greater than the fluctuation coefficient preset by the system, and if yes, the size of the data block that is currently sent by using the connection is decreased, and preferably, the size of the data block that is currently sent by using the connection is decreased to a half.

[0052] If the current transmission speed is less than the average speed of previous two data blocks, and the decreasing proportion is less than or equal to the fluctuation coefficient preset by the system, the size of the data block remains unchanged.

[0053] Preferably, the fluctuation coefficient is set as follows:

in a 2G network, the fluctuation coefficient is 0.3; in a 3G network, the fluctuation coefficient is 0.2; and in 4G and wifi networks, the fluctuation coefficient is 0.2.

[0054] S305: Fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window.

[0055] S306: Acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0056] S307: Clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

[0057] By means of the foregoing mechanism of adjusting a size of a data block, in a weak network environment, a connection with a higher transmission speed is used for sending a larger data block, and transmission efficiency thereof is also higher. In addition, a size of a data block is adjusted according to a transmission speed, and when communications quality of a connection is poor, a case in which a large data block needs to be discarded and retransmitted as a whole due to a bit error in some data units in a transmission process can be avoided by decreasing a size of a data block. For example, one data block includes 4 data units; if a severe bit error occurs in the third data unit, the entire data block is discarded and retransmitted; however, by decreasing a size of a data block, 4 data units are grouped into two data blocks, and if a severe bit error occurs in the third data unit, a data block formed by the first and second data units can still be received normally, thereby reducing an amount of retransmitted data.

[0058] Refer to FIG. 4, which is a flowchart of a third embodiment of a file transmission method according to the present disclosure.

[0059] S401: Establish at least two connections to a receive end, each connection including at least one send window.

[0060] S402: Divide a to-be-sent file into at least two data blocks, fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window.

[0061] When a data block is sent by using a send window, sequence number information of the data block is sent to the receive end at the same time, so that the receive end receives and recombines the to-be-sent file. After the data block is sent, the data block is added to a timing queue.

[0062] S403: Acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0063] S404: Clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection.

[0064] A transmit end can know, according to the data block information received this time in the acknowledgement information returned by the receive end, which data block in which

connection is successfully received by the receive end. A window in which a data block is successfully sent is cleared, so that the window can be used for sending a next data block.

[0065] For a send window for which acknowledgement information is not received, a data block in the send window is not processed, unless the data block in the window is marked as a timeout data block.

[0066] S405: Determine whether the data block times out, if yes, mark a data block that is sent previously as a timeout data block, further fill the timeout data block in the cleared send window, and send the timeout data block to the receive end by using the send window.

[0067] If the acknowledgement information is not received within a specified time, that is, for the data block added to the timing queue, acknowledgement information returned by the receive end is not received before a timing time expires, the data block is marked as a timeout data block, and the data block is added to a timeout queue.

[0068] In addition, if acknowledgement information for a data block that is sent previously is not received and acknowledgement information for a preset number of data blocks that are sent subsequently is received, the data block that is sent previously is marked as a timeout data block, and the data block is added to the timeout queue. Preferably, if a value preset by a system is 2, that is, acknowledgement information for two data blocks that are sent subsequently is received while acknowledgement information for a data block that is sent previously is not received, the data block that is sent previously is marked as a timeout data block, and the data block is added to the timeout queue. For example, if acknowledgement information for a data block that is sent the third time and a data block that is sent the fourth time by using a connection is received, but acknowledgement information for a data block that is sent the second time is not received, even though a timing time of the data block that is sent the second time does not expire, the data block is still marked as a timeout data block, and the data block is added to the timeout queue.

[0069] Finally, the timeout data block is extracted from the timeout queue, the timeout data block is filled in the cleared send window, and the timeout data block is sent to the receive end by using the send window.

[0070] When a data block is lost or a reply is lost, there is no need to mark the data block as a timeout data block after the timing time expires; in this way, a time of retransmitting the data block is reduced, and finally, a time of an entire transmission process is reduced.

[0071] S406: Fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

[0072] When a data block is lost or a reply is lost, there is no need to mark the data block as a timeout data block after the timing time expires; in this way, a time of retransmitting the data block is reduced, and finally, a time of an entire transmission process is reduced.

[0073] Refer to FIG. 5, which is a schematic diagram of a file transmission process of a fourth embodiment of a file transmission method according to the present disclosure.

[0074] S501: Establish at least two connections to a receive end, each connection including at least one send window.

[0075] In this embodiment, a transmit end is an instant messaging client, and the receive end is a server; the number of connections established between the transmit end and the receive end is N ; and each connection includes M send windows.

[0076] S502: Divide a to-be-sent file into at least two data blocks, and fill a corresponding data block in a send window of each connection.

[0077] In this embodiment, the to-be-sent file is divided into MM data blocks, where MM is greater than 2. A data block is filled in the send window of each connection until the send window is filled up. Because each connection includes M send windows, the number of data blocks filled each time is M at most. In FIG. 5, $FSended$ indicates a data block that is sequentially sent by the transmit end and for which a reply of acknowledgement is received, and $Inflight$ indicates a maximal position, in which a data block of a file to be sent by the transmit end is filled, in a send window. That is, a state indicated in FIG. 5 is that data blocks 1 to $2M$ have been filled in the send window, while data blocks 1 to M are sequentially sent and acknowledgement information of the server is received.

[0078] S503: Send the data block to the receive end by using the send window.

[0079] When a data block is sent by using a send window, sequence number information of the data block is sent to the receive end at the same time, so that the receive end receives and recombines the to-be-sent file. The sequence number information of the data block includes an offset $Offset$ of the data block relative to a start position of the to-be-sent file and a file size $Length$ in this embodiment; in addition, a global increasing Seq number $Seq X$ is set to indicate a sending sequence of a data block. For example, the data block 1 is sent first, so that a global increasing Seq number of the data block is $Seq 1$; a data block 2 is sent second, so that a global increasing Seq number of the data block is $Seq 2$; and so on. After the data block is sent, the data block is added to a timing queue.

[0080] S504: Acquire acknowledgement information returned, after the receive end receives

the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0081] In this embodiment, AckSeq X indicates a reply of acknowledgement for a received data block. AckNext SeqX indicates a next data block that a server end expects to receive according to a data block sequence. For example, AckSeq M indicates that the data block M is successfully received by the server, and AckNextSeq M+1 indicates that the a sequence number of a next data block that the server end expects to receive according to a data block sequence is M+1.

[0082] S505: Clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection.

[0083] The transmit end can know, according to the data block information received this time in the acknowledgement information received by and returned by the receive end, which data block in which connection is successfully received by the receive end. A window in which a data block is successfully sent is cleared, so that the window can be used for sending a next data block.

[0084] In this embodiment, as shown in FIG. 5, the data block M is successfully received by the server; therefore, a send window carrying the data block M is cleared, so that a next data block can be filled in and sent to the server.

[0085] S506: Determine whether the data block times out, if yes, mark a data block that is sent previously as a timeout data block, further fill the timeout data block in the cleared send window, and send the timeout data block to the receive end by using the send window.

[0086] If the acknowledgement information is not received within a specified time, that is, for the data block added to the timing queue, acknowledgement information returned by the receive end is not received before a timing time expires, the data block is marked as a timeout data block, and the data block is added to a timeout queue.

[0087] In this embodiment, as shown in FIG. 5, in a connection 2, the data block M+1 is lost in a transmission process; as a result, the server does not receive the data block, or the server has received the data block but the transmit end does not receive the acknowledgement information within the specified time, that is, ACK timeout, so that the data block M+1 is added to the timeout queue. The transmit end receives acknowledgement information for a data block 2M+1 in a connection 3; therefore, the send window is cleared, the data block M+1 in the timeout queue is filled in the send window in the connection 3, and the data block M+1 is sent to the server again by using the send window.

[0088] S507: Fill, in the cleared send window, a data block that the receive end expects to

receive, and send the data block to the receive end by using the send window.

[0089] If no data block times out, or there is no data block in the timeout queue, the data block that the receive end expects to receive is read from the to-be-sent file, is filled in the cleared send window, and is sent to the receive end by using the send window.

[0090] In order to further describe working processes of the transmit end and the server, a representative time slice is selected for description, as shown in FIG. 6.

[0091] FIG. 6 describes a process of sending a data block 4 at a moment in FIG. 5, the server acknowledges that the data blocks 1 and 2 are received, the data block 3 is not received by the server yet or is lost, and a data block 5 has been sent in another connection or is acknowledged by the receive end; therefore, the data block 5 is a received non-sequential data block. In this case, the server receives the data block 4, and data block information of the data block 4 includes Offset 4, Length Sx, and Seq4, indicating that an offset of the data block 4 relative to the start position of the to-be-sent file is 4, a data length is Sx, and a sending sequence is 4. Because the data block 3 is not received by the server yet or is lost, acknowledgement information AckSeq 4, AckNextSeq 3, and Seq 4 is replied, indicating that the receive end has received the data block 4 of which a sending sequence at the transmit end is 4, but the data block 3 is absent, and the receive end expects the transmit end to send the data block 3. In this case, because it is acknowledged that the data block 4 is received, a send window for sending the data block 4 is cleared. When the transmit end sends the data block 3 again, the receive end replies AckSeq 3, AckNextSeq 6, and a global Seq X of the data block 3; and because the receive end has received the data block 4 and the data block 5, a data block 6 is expected to be received next sequentially.

[0092] Refer to FIG. 7, which is a structural block diagram of a file transmission system according to the present disclosure.

[0093] A connection module 110 is configured to establish at least two connections to a receive end, each connection including at least one send window.

[0094] The connection module 110 establishes connections between a transmit end and the receive end, the number of the connections is at least two, and each connection includes at least one send window. According to development complexity and a yield ratio, preferably, the number of the connections is not greater than 3, and the number of send windows may be as large as possible. If the receive end is a server, further, the connections may be at least two connections between the transmit end and one server, and may also be connections to at least two servers, and one connection exists between each server and the transmit end.

[0095] Data can be sent in parallel by establishing multiple connections by the connection module 110, so that in a weak network environment, a bandwidth resource is fully used and transmission of a to-be-sent file is accelerated.

[0096] The send window is a sending tool of the connection, and is used for carrying to-be-sent data and sending a data block to the receive end.

[0097] A dividing and sending module 120 is configured to divide a to-be-sent file into at least two data blocks, fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window.

[0098] The dividing and sending module 120 divides the to-be-sent file into at least two data blocks, distributes different data blocks to each connection, fills a data block in the send window of each connection, and fills one data block in each send window. The data blocks are sent to the receive end by using the send windows, so that the data blocks are sent in parallel to reduce a file transmission time.

[0099] An acknowledgement acquiring module 130 is configured to acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0100] After successfully receiving the data block, the receive end returns the acknowledgement information to the transmit end, where the acknowledgement information includes the data block information received this time and the data block information that the receive end expects to receive. The data block information received this time may be sequence number information of a data block received this time by the receive end, and the data block information that the receive end expects to receive may be sequence number information of a data block that the receive end expects to receive.

[0101] The transmit end can clearly know, from the acknowledgement information, the data block information and the data block information that the receive end expects to receive, which are received successfully, and the transmit end sends a corresponding data block to the receive end according to the information.

[0102] A sending module 140 is configured to clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

[0103] The data block information received this time in the acknowledgement information clearly indicates, to the transmit end, which data block has been received by the receive end; therefore, the sending module 140 clears the send window corresponding to the data block information received this time; and as long as there is a connection with a cleared send window, the sending module 140 fills, in the cleared send window, the data block that the receive end expects to receive, and sends the data block to the receive end by using the send window.

[0104] According to the present disclosure, a connection module 110 establishes at least two connections to a receive end, each connection including at least one send window. A dividing and sending module 120 divides a to-be-sent file into multiple data blocks, and distributes the data blocks to different connections for parallel sending. An acknowledgement acquiring module 130 acquires acknowledgement information returned, after the receive end receives the data block, by the receive end, and finally, a sending module 140 fills, in a cleared send window according to the acknowledgement information of the receive end, a data block that the receive end expects to receive, and sends the data block to the receive end by using the send window, until all the data blocks are received by the receive end. In the file transmission system in the present disclosure, there is no fixed mapping relationship between each connection and a file block, but for a connection with a send window that is cleared first, the dividing and sending module 120 fills, in the send window, a data block sent next, that is, for a connection with a higher transmission speed, the dividing and sending module 120 distributes more data blocks to the connection. Correspondingly, a transmission time of the to-be-sent file can be effectively reduced. Distribution of a data block by the dividing and sending module 120 depends on an idle degree and a speed of a connection, especially in a case in which an individual connection is stuck due to poor transmission quality or a low transmission speed of the connection, and according to the present disclosure, a data block can be distributed to a connection with a high transmission speed, which avoids an overall increase of a consumed time caused by a stuck individual connection. According to the present disclosure, utilization of bandwidth is improved and a file transmission time is effectively reduced in a connection manner not limited to TCP, UDP, and HTTP by means of multi-connection and multi-window concurrency and proper data block distribution and filling.

[0105] Refer to FIG. 8, which is a structural block diagram of an embodiment of a file transmission system according to the present disclosure.

[0106] A connection module 210 is configured to establish at least two connections to a receive end, each connection including at least one send window.

[0107] A combination module 220 is configured to divide a to-be-sent file into at least two

data units, where each data block includes several consecutive data units.

[0108] When each connection is used the first time to send a data block, each data block includes at least two data units if a size of the to-be-sent file is less than a preset value; otherwise, each data block includes one data unit.

[0109] The combination module 220 first divides the to-be-sent file divided into data units of the smallest unit, and then, several consecutive data units are combined into a data block. The number of data units included in each data block is determined by a transmission speed of each connection; therefore, sizes of data blocks sent by all the connections may be different, and sizes of data blocks sent by a same connection at different times may also be different, but a size of a data block cannot be greater than a maximum value of the size of the data block.

[0110] Preferably, when a data block is sent initially, if a size of the to-be-sent file is less than 100 K, one data block includes 4 data units, or if a size of the to-be-sent file is greater than or equal to 100 K, one data block includes one data unit combination.

[0111] Preferably, a data unit and a maximum value of a size of a data block are set as follows:

in a 2G network, a size of a data unit is 8 KB, and a maximum value of a size of a data block is 64 KB;

in a 3G network, a size of a data unit is 32 KB, and a maximum value of a size of a data block is 512 KB; and

in 4G and wifi networks, a size of a data unit is 64 KB, and a maximum value of a size of a data block is 1024 KB.

[0112] A first dividing module 230 is configured to separately acquire a current transmission speed and an average transmission speed of each connection, and increase, if the current transmission speed is not less than the average transmission speed, a size of a data block that is currently sent by using the connection.

[0113] A second dividing module 240 is configured to decrease, if the current transmission speed is less than an average speed of previous two data blocks and a decreasing proportion is greater than a fluctuation coefficient preset by a system, the size of the data block that is currently sent by using the connection.

[0114] A size of a data block in each connection is adjusted according to a different transmission speed of each connection. First, the first dividing module 230 separately acquires the

current transmission speed and the average transmission speed of each connection. Preferably, the average transmission speed is an average speed of sending previous two data blocks by using the connection.

[0115] Separately for each connection, the first dividing module 230 compares the current transmission speed of each connection with the average speed of sending previous two data blocks by using the connection, if the current transmission speed is not less than the average speed of previous two data blocks, the first dividing module 230 increases the size of the data block that is currently sent by using the connection; and preferably, the size of the data block that is currently sent is increased to twice the original size of the data block, but the size of the data block cannot be greater than a maximum value of the size of the data block.

[0116] Preferably, a data unit and a maximum value of a size of a data block are set as follows:

in a 2G network, a maximum value of a size of a data block is 64 KB; in a 3G network, a maximum value of a size of a data block is 512 KB; and

in 4G and wifi networks, a maximum value of a size of a data block is 1024 KB.

[0117] For example, if the current transmission speed is less than the average speed of previous two data blocks, the second dividing module 240 continues to compare the decreasing proportion with the fluctuation coefficient preset by the system to determine whether the decreasing proportion is greater than the fluctuation coefficient preset by the system, and if yes, the second dividing module 240 decreases the size of the data block that is currently sent by using the connection, and preferably, decreases the size of the data block that is currently sent by using the connection to a half.

[0118] If the current transmission speed is less than the average speed of previous two data blocks, and the decreasing proportion is less than or equal to the fluctuation coefficient preset by the system, the second dividing module 240 remains the size of the data block unchanged.

[0119] Preferably, the fluctuation coefficient is set as follows:

in a 2G network, the fluctuation coefficient is 0.3; in a 3G network, the fluctuation coefficient is 0.2; and in 4G and wifi networks, the fluctuation coefficient is 0.2.

[0120] An acknowledgement acquiring module 250 is configured to acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information including data block information received this time and data block information that the receive end expects to receive.

[0121] After successfully receiving the data block, the receive end returns the acknowledgement information to the transmit end, where the acknowledgement information includes the data block information received this time and the data block information that the receive end expects to receive. The acknowledgement acquiring module 240 acquires the acknowledgement information, and the transmit end can clearly know, from the acknowledgement information, the data block information that the receive end expects to receive, and the transmit end sends a corresponding data block to the receive end according to the information.

[0122] If a bit error occurs in a transmission process of the data and it is impossible to recover, the data block is discarded; because the receive end does not successfully receive the data block, the receive end does not return the acknowledgement information to the transmit end.

[0123] A timeout sending module 260 is configured to mark a data block that is sent this time as a timeout data block if the acknowledgement information is not received within a specified time or acknowledgement information for the data block that is sent previously is not received and acknowledgement information for a preset number of data blocks that are sent subsequently is received, fill the timeout data block in the cleared send window, and send the timeout data block to the receive end by using the send window.

[0124] A sending module 270 is configured to clear, according to the acknowledgement information, a send window corresponding to the data block information that is received this time by using the connection, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window, until all the data blocks are received by the receive end.

[0125] First, the transmit end can know, according to the data block information received this time in the acknowledgement information received by and returned by the receive end, which data block in which connection is successfully received by the receive end. The sending module 270 clears a window in which a data block is successfully sent, so that the window can be used for sending a next data block.

[0126] If the acknowledgement information is not received within the specified time, that is, for the data block added to the timing queue, acknowledgement information returned by the receive end is not received before a timing time expires, the timeout sending module 260 marks the data block as a timeout data block, and adds the data block to a timeout queue.

[0127] In addition, if the acknowledgement information for the data block that is sent previously is not received and the acknowledgement information for the preset number of data

blocks that are sent subsequently is received, the data block that is sent previously is marked as a timeout data block, and the data block is added to the timeout queue. Preferably, if a value preset by a system is 2, that is, acknowledgement information for two data blocks that are sent subsequently is received while acknowledgement information for a data block that is sent previously is not received, the data block that is sent previously is marked as a timeout data block, and the data block is added to the timeout queue. For example, if acknowledgement information for a data block that is sent the third time and a data block that is sent the fourth time by using a connection is received, but acknowledgement information for a data block that is sent the second time is not received, even though a timing time of the data block that is sent the second time does not expire, the timeout sending module 260 still marks the data block as a timeout data block, and adds the data block to the timeout queue.

[0128] Then, the timeout sending module 260 extracts the timeout data block from the timeout queue, fills the timeout data block in the cleared send window, and sends the timeout data block to the receive end by using the send window.

[0129] When a data block is lost or a reply is lost, there is no need for the timeout sending module 260 to mark the data block as a timeout data block after the timing time expires; in this way, a time of retransmitting the data block is reduced, and finally, a time of an entire transmission process is reduced.

[0130] The sending module 270 fills, in the cleared send window, the data block that the receive end expects to receive, and sends the data block to the receive end by using the send window, until all the data blocks are received by the receive end, so as to send the to-be-sent file to the receive end completely.

[0131] According to the present disclosure, a connection module 210 establishes at least two connections to a receive end, each connection including at least one send window. A combination module 220 divides a to-be-sent file into multiple data blocks. A first dividing module 230 and a second dividing module 240 adjust sizes of data blocks and distribute the data blocks to different connections for parallel sending. Finally, a timeout sending module 260 sends a timeout data block to the receive end, and a sending module 270 fills, in a cleared send window according to acknowledgement information of the receive end, a data block that the receive end expects to receive, and sends the data block to the receive end by using the send window, until all the data blocks are received by the receive end. In the file transmission system in the present disclosure, there is no fixed mapping relationship between each connection and a file block, but for a connection with a send window that is cleared first, a data block sent next is filled in the send

window, that is, for a connection with a higher transmission speed, more data blocks are distributed to the connection. Correspondingly, a transmission time of the to-be-sent file can be effectively reduced. By means of the foregoing adjusting mechanism of adjusting sizes of data blocks by the first dividing module 230 and the second dividing module 240, in a weak network environment, a connection with a higher transmission speed is used for sending a larger data block, and transmission efficiency thereof is also higher. In addition, the first dividing module 230 and the second dividing module 240 adjust sizes of data blocks according to a transmission speed, and when communications quality of a connection is poor, a case in which a large data block needs to be discarded and retransmitted as a whole due to a bit error in some data units in a transmission process can be avoided by decreasing a size of a data block by the second dividing module 240. Further, when a data block is lost or a reply is lost, there is no need for the timeout sending module 260 to mark the data block as a timeout data block after the timing time expires; in this way, a time of retransmitting the data block is reduced, and finally, a time of an entire transmission process is reduced.

[0132] Unless the context clearly indicates otherwise, singular elements or components in the present disclosure may be in the plural and vice versa, which is not limited in the present disclosure. Although steps in the present disclosure are labeled with numbers, such numbers are not intended to limit the order of these steps. Unless the order of steps is explicitly stated or it is explicitly stated that a step needs to be performed based on another step, the relative order of the steps can be adjusted. It may be understood that as used herein, the term "and/or" involves and includes any and all combinations of one or more of the associated listed items.

[0133] FIG. 9 is a diagram of modules of a computer system 1000 that can implement the embodiments of the present invention. The computer system 1000 is merely an example applicable to a computer environment of the present disclosure, but it cannot be construed as any limitation to the use scope of the present disclosure. The computer system 1000 cannot be interpreted as needing to rely on or having one or a combination of parts of the exemplary computer system 1000 shown in the figure.

[0134] The computer system 1000 shown in FIG. 9 is an example of computer systems applicable to the present disclosure. Other architectures having different subsystem configurations may also be used. For example, a desktop computer, a notebook computer, a personal digital assistant, a smart phone, a tablet computer, a portable media player, a set-top box and other similar devices are applicable to some embodiments of the present disclosure, but the present disclosure is not limited thereto.

[0135] As shown in FIG. 9, the computer system 1000 includes a processor 1010, a memory 1020, and a system bus 1022. Various system components including the memory 1020 and the processor 1010 are connected to the system bus 1022. The processor 1010 is hardware for executing computer program instructions by means of basic arithmetic and logic operations in the computer system. The memory 1020 is a physical device for temporarily or permanently storing computer programs or data (for example, program state information). The system bus 1022 may be any one of the following types of bus structures: a memory bus or memory controller, a peripheral bus and a local bus. The processor 1010 and the memory 1020 can perform data communication through the system bus 1022. The memory 1020 includes a read-only memory (ROM) or a flash memory (neither is shown in the figure), and a random access memory (RAM), where the RAM generally refers to a main memory loaded with an operating system and application programs.

[0136] The computer system 1000 further includes a display interface 1030 (for example, a graphics processing unit), a display device 1040 (for example, a liquid crystal display), an audio interface 1050 (for example, a sound card), and an audio device 1060 (for example, a loudspeaker). The display device 1040 and the audio device 1060 are media devices for presenting multimedia content.

[0137] The computer system 1000 generally includes one storage device 1070. The storage device 1070 may be selected from multiple types of computer readable media. The computer readable media refer to any available media that can be accessed by the computer system 1000, and include removable media and non-removable media. For example, the computer readable media include, but are not limited to, a flash memory (micro SD card), a CD-ROM, a digital versatile disc (DVD) or other optical storage, a cassette, a magnetic tape, a disk storage or other magnetic storage devices, or any other media that can be used to store required information and can be accessed by the computer system 1000.

[0138] The computer system 1000 further includes an input apparatus 1080 and an input interface 1090 (for example, an IO controller). A user may input an instruction and information into the computer system 1000 by using the input apparatus 1080, such as a keyboard, a mouse or a touch panel device on the display device 1040. The input apparatus 1080 is generally connected to the system bus 1022 through the input interface 1090, but may also be connected through other interfaces or bus structures, such as a universal serial bus (USB).

[0139] The computer system 1000 may be logically connected to one or more network devices in a network environment. The network device may be a personal computer, a server, a router, a smart phone, a tablet computer or other public network nodes. The computer system 1000

is connected to the network device through a local area network (LAN) interface 1100 or a mobile communications unit 1110. An LAN refers to an interconnected computer network in a limited area such as a house, a school, a computer laboratory, or an office building using network media. WiFi and Ethernet over twisted pair are the most commonly used two technologies for building an LAN. WiFi is a technology that enables the computer systems 1000 to exchange data or to be connected to a wireless network through radio waves. The mobile communications unit 1110 is capable of making and receiving calls through radio communications lines while moving in a broad geographic area. In addition to calling, the mobile communications unit 1110 also supports Internet access in a 2G, 3G or 4G cellular communications system that provides mobile data services.

[0140] It should be noted that other computer systems including more or fewer subsystems than those of the computer system 1000 are also applicable to the present disclosure. For example, the computer system 1000 may include a Bluetooth unit capable of exchanging data in a short distance, an image sensor for capturing images, and an accelerometer for measuring the acceleration.

[0141] As described above in detail, the computer system 1000 applicable to the present disclosure can perform specified operations in the file transmission method. The computer system 1000 performs these operations in the form of running software instructions in the computer readable media by the processor 1010. These software instructions may be read into the memory 1020 from the storage device 1070 or from another device through the LAN interface 1100. The software instructions stored in the memory 1020 enable the processor 1010 to perform the file transmission method. Moreover, the present disclosure may also be implemented by using a hardware circuit or by using a combination of a hardware circuit and software instructions. Therefore, the implementation of the present disclosure is not limited to any particular combination of a hardware circuit and software.

[0142] The foregoing embodiments only show several implementation manners of the present disclosure and are described in detail, but they should not be construed as a limit to the patent scope of the present disclosure. It should be noted that, a person of ordinary skill in the art may make various changes and improvements without departing from the ideas of the present disclosure, which shall fall within the protection scope of the present disclosure. Therefore, the protection scope of the patent of the present disclosure shall be subject to the claims.

CLAIMS

What is claimed is:

1. A file transmission method, comprising steps of:

establishing at least two connections to a receive end, each connection comprising at least one
5 send window;

dividing a to-be-sent file into at least two data blocks, filling a corresponding data block in a
send window of each connection, and sending the data block to the receive end by using the send
window;

acquiring acknowledgement information returned, after the receive end receives the data block,
10 by the receive end, the acknowledgement information comprising data block information of the data
block received this time and data block information of a data block that the receive end expects to
receive; and

clearing, according to the acknowledgement information, a send window corresponding to the
data block information received this time, filling, in the cleared send window, a data block that the
15 receive end expects to receive, and sending the data block to the receive end by using the send
window.

2. The file transmission method according to claim 1, after the step of dividing a to-be-sent file
into at least two data blocks, further comprising the following step:

separately acquiring a current transmission speed and an average transmission speed of each
20 connection, and increasing, if the current transmission speed is not less than the average
transmission speed, a size of a data block that is currently sent by using the connection.

3. The file transmission method according to claim 2, wherein

decreasing, if the current transmission speed is less than the average transmission speed and a
decreasing proportion is greater than a fluctuation coefficient preset by a system, the size of the data
25 block that is currently sent by using the connection.

4. The file transmission method according to claim 1, wherein the step of dividing a to-be-sent
file into at least two data blocks comprises:

dividing the to-be-sent file into at least two data units, wherein each data block comprises
several consecutive data units, wherein

30 when each connection is used the first time to send a data block, each data block comprises at

least two data units if a size of the to-be-sent file is less than a preset value; otherwise, each data block comprises one data unit.

5 5. The file transmission method according to claim 1, after the step of clearing a send window corresponding to the data block information received this time, further comprising the following step:

marking a data block that is sent previously as a timeout data block if the acknowledgement information is not received within a specified time or acknowledgement information for the data block that is sent previously is not received and acknowledgement information for a preset number of data blocks that are sent subsequently is received, filling the timeout data block in the cleared
10 send window, and sending the timeout data block to the receive end by using the send window.

6. A file transmission system, comprising:

a connection module, configured to establish at least two connections to a receive end, each connection comprising at least one send window;

a dividing and sending module, configured to divide a to-be-sent file into at least two data
15 blocks, fill a corresponding data block in a send window of each connection, and send the data block to the receive end by using the send window;

an acknowledgement acquiring module, configured to acquire acknowledgement information returned, after the receive end receives the data block, by the receive end, the acknowledgement information comprising data block information of the data block received this time and data block
20 information of a data block that the receive end expects to receive; and

a sending module, configured to clear, according to the acknowledgement information, a send window corresponding to the data block information received this time, fill, in the cleared send window, a data block that the receive end expects to receive, and send the data block to the receive end by using the send window.

25 7. The file transmission system according to claim 6, further comprising:

a first dividing module, configured to separately acquire a current transmission speed and an average transmission speed of each connection, and increase, if the current transmission speed is not less than the average transmission speed, a size of a data block that is currently sent by using the connection.

30 8. The file transmission system according to claim 7, further comprising:

a second dividing module, configured to decrease, if the current transmission speed is less than

an average speed of previous two data blocks and a decreasing proportion is greater than a fluctuation coefficient preset by a system, the size of the data block that is currently sent by using the connection.

9. The file transmission system according to claim 6, further comprising:

- 5 a combination module, configured to divide the to-be-sent file into at least two data units, wherein each data block comprises several consecutive data units, wherein when each connection is used the first time to send a data block, each data block comprises at least two data units if a size of the to-be-sent file is less than a preset value; otherwise, each data block comprises one data unit.

10. The file transmission system according to claim 6, further comprising:

- 10 a timeout sending module, configured to mark a data block that is sent previously as a timeout data block if the acknowledgement information is not received within a specified time or acknowledgement information for the data block that is sent previously is not received and acknowledgement information for a preset number of data blocks that are sent subsequently is received, fill the timeout data block in the cleared send window, and send the timeout data block to
15 the receive end by using the send window.