



(12)发明专利

(10)授权公告号 CN 104539739 B

(45)授权公告日 2019.03.29

(21)申请号 201510038094.6

(22)申请日 2015.01.26

(65)同一申请的已公布的文献号

申请公布号 CN 104539739 A

(43)申请公布日 2015.04.22

(73)专利权人 浙江大学

地址 310012 浙江省杭州市西湖区余杭塘路866号

(72)发明人 冯冬芹 黄家辉

(74)专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 王宝筠

(51)Int.Cl.

H04L 29/08(2006.01)

H04L 29/06(2006.01)

(56)对比文件

CN 103312732 A,2013.09.18,

CN 102752283 A,2012.10.24,

CN 103312732 A,2013.09.18,

CN 101465791 A,2009.06.24,

US 7565533 B2,2009.07.21,

CN 101877620 A,2010.11.03,

CN 101645883 A,2010.02.10,

CN 103346830 A,2013.10.09,

CN 102752402 A,2012.10.24,

审查员 田雨润

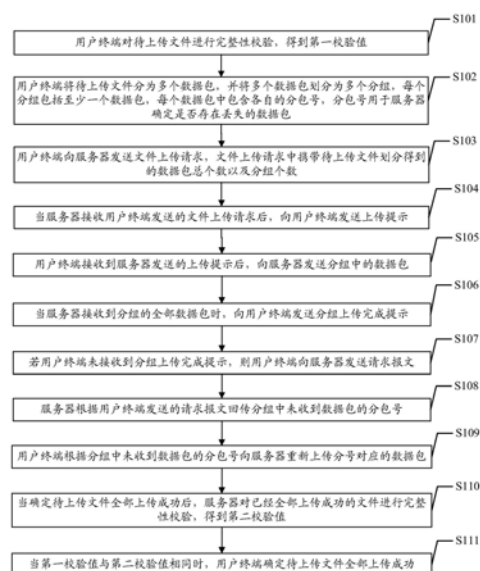
权利要求书5页 说明书19页 附图6页

(54)发明名称

一种文件上传的系统、方法及装置

(57)摘要

本发明实施例公开了一种文件上传的系统,包括:用户终端对待上传文件进行完整性校验,得到第一校验值;将该文件分为多个数据包,将多个数据包划分为多个分组;向服务器发送文件上传请求;发送分组中的数据包;当服务器接收到分组的全部数据包时发送分组上传完成提示;若用户终端未接收到分组上传完成提示则发送请求报文;服务器根据请求报文回传未收到数据包的分包号;用户终端根据该分包号向服务器重新上传对应数据包;当文件上传成功后,服务器对文件进行完整性校验,得到第二校验值;第一校验值与第二校验值相同时,确定该文件全部上传成功。本发明实施例能够提高文件的上传的成功率,实时将未收到数据包重新上传,保证文件传输的完整性。



1. 一种文件上传的系统,其特征在于,包括:用户终端与服务器;
所述用户终端用于发送文件;
所述服务器用于接收所述文件;
所述文件上传的系统用于进行完整性校验、划分数据包处理、分组发送数据包、实时丢包重传及响应;
所述用户终端对待上传文件进行所述完整性校验,得到第一校验值;
所述用户终端将所述待上传文件分为多个数据包,并将所述多个数据包划分为多个分组,每个分组包括至少一个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器确定是否存在丢失的数据包;
所述用户终端向所述服务器发送文件上传请求,所述文件上传请求中携带所述待上传文件划分得到的数据包总个数以及分组个数;
当所述服务器接收所述用户终端发送的所述文件上传请求后,向所述用户终端发送上传提示;
所述用户终端接收到所述服务器发送的上传提示后,向所述服务器发送分组中的数据包;
当所述服务器接收到所述分组的全部数据包时,向所述用户终端发送分组上传完成提示;
若所述用户终端未接收到分组上传完成提示,则所述用户终端向所述服务器发送请求报文;
所述服务器根据所述用户终端发送的请求报文回传分组中未收到数据包的分包号;
所述用户终端根据所述分组中未收到数据包的分包号向所述服务器重新上传所述分包号对应的数据包;
当确定所述待上传文件全部上传成功后,所述服务器对已经全部上传成功的文件进行完整性校验,得到第二校验值;
当所述第一校验值与所述第二校验值相同时,用户终端确定所述待上传文件全部上传成功。
2. 根据权利要求1所述的系统,其特征在于,所述当所述服务器接收所述用户终端发送的所述文件上传请求后,向所述用户终端发送上传提示之前,所述系统还包括:
若所述用户终端未接收到所述服务器发送的上传提示,则继续发送文件上传请求;
用户终端判断所述文件上传请求次数是否超过预置发送次数;
若未超过,则所述用户终端重新向服务器发送文件上传请求;
若超过,则所述用户终端终止所述待上传文件的上传。
3. 根据权利要求1所述的系统,其特征在于,所述当所述服务器接收到所述每个分组的全部数据包时,向所述用户发送分组上传完成提示之后,所述系统还包括:
当所述用户终端接收到所述分组上传完成提示时,判断剩余数据包个数是否大于每组中数据包的总数;
若剩余数据包个数大于每组中数据包的总数,则所述用户终端根据分包号向所述服务器发送数据包;
若剩余数据包个数小于等于每组中数据包的总数,则所述用户终端发送所有剩余的数

据包。

4. 一种文件上传的方法,其特征在于,包括:

对待上传文件进行完整性校验,得到第一校验值;

将所述待上传文件划分为多个数据包;

向服务器发送文件上传请求,所述文件上传请求中携带所述待上传文件划分得到的数据包总个数;

接收到所述服务器发送的上传提示后,向所述服务器发送所述多个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器根据所述数据包的总个数确定所述待上传文件是否全部上传成功;

当所述待上传文件中所有的数据包上传成功后,接收所述服务器发送的第二校验值,所述第二校验值为所述服务器确定所述待上传文件全部上传完成后,对所述待上传文件进行完整性校验得到的;

当所述第一校验值与所述第二校验值相同时,确定所述待上传文件全部上传成功。

5. 根据权利要求4所述的方法,其特征在于,所述将所述待上传文件划分为多个数据包之后,还包括:

将所述多个数据包划分为多个分组,每个分组包括至少一个数据包;

所述接收到所述服务器发送的上传提示后,向所述服务器发送所述多个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器根据所述数据包的总个数确定所述待上传文件是否全部上传成功包括:

接收到所述服务器发送的分组上传提示后,判断所述待上传文件中所有的数据包个数是否大于分组要求的分包个数;

若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包;

所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包;

若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

6. 根据权利要求4所述的方法,其特征在于,所述向服务器发送文件上传请求,所述文件上传请求中携带所述待上传文件划分得到的数据包总个数包括:

向所述服务器发送文件上传请求;

等待所述服务器发送的上传提示;

若未收到上传提示,则根据预置的最大发送文件上传请求次数,向所述服务器发送文件上传请求;

当超过预置的最大发送文件上传请求次数时,则中断文件上传。

7. 根据权利要求5所述的方法,其特征在于,所述若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包之后,还包括:

若未接收到所述服务器发送分组上传完成提示,则重新向所述服务器发送文件上传请求;

接收所述服务器发送未收到的数据包的分包号;

根据所述未收到的数据包的分包号发送所述分包号对应的数据包。

8. 一种文件上传的方法, 其特征在于, 包括:

接收用户终端发送的文件上传请求;

根据所述文件上传请求, 向所述用户终端发送文件分组上传提示;

接收所述用户终端发送的多个数据包;

根据每个数据包中包含的各自的分包号与所述文件上传请求中携带的待上传文件划分得到的数据包总个数, 确定所述待上传文件是否全部上传成功;

当确定所述待上传文件全部上传成功后, 对已经全部上传成功的文件进行完整性校验, 得到第二校验值;

向所述用户终端发送所述第二校验值, 以使所述用户终端在第一校验值和所述第二校验值相同时, 确定所述待上传文件全部上传成功, 所述第一校验值为对所述待上传文件进行完整性校验, 得到的校验值。

9. 根据权利要求8所述的方法, 其特征在于, 所述接收所述用户终端发送的所述多个数据包之后, 还包括:

根据所述多个分组, 判断是否接收到每个分组中全部的数据包;

若接收到每个分组中全部的数据包, 则向所述用户终端发送上传完成提示;

若没有接收到每个分组中全部的数据包, 则继续等待接收所述每个分组中的剩余数据包。

10. 根据权利要求8至9中任一项所述的方法, 其特征在于, 所述接收所述用户终端发送的所述多个数据包之后, 还包括:

提取所述数据包中的冗余检验码CRC;

所述根据所述多个分组, 判断是否接收到每个分组中全部的数据包包括:

当接收到多个分组中的每个数据包时, 根据CRC验证码判断所述每个数据包中的数据是否正确;

若分组中所述每个数据包的数据均正确, 则向所述用户终端发送上传正确完成提示;

若分组中有一个或多个数据包中的数据错误, 则丢弃所述数据包。

11. 一种用户终端, 其特征在于, 包括:

第一校验模块, 用于对待上传文件进行完整性校验, 得到第一校验值;

第一划分模块, 用于将所述待上传文件划分为多个数据包;

第一发送模块, 用于向服务器发送文件上传请求, 所述文件上传请求中携带所述待上传文件划分得到的数据包总个数;

第二发送模块, 用于接收到所述服务器发送的上传提示后, 向所述服务器发送所述多个数据包, 每个数据包中包含各自的分包号, 所述分包号用于所述服务器根据所述数据包的总个数确定所述待上传文件是否全部上传成功;

第一接收模块, 用于当所述待上传文件中所有的数据包上传成功后, 接收所述服务器发送的第二校验值, 所述第二校验值为所述服务器确定所述待上传文件全部上传完成后, 对所述待上传文件进行完整性校验得到的;

第一确定模块, 用于当所述第一校验值与所述第二校验值相同时, 确定所述待上传文件全部上传成功。

12. 根据权利要求11所述的用户终端,其特征在于,所述用户终端还包括:

第二划分模块,用于将所述多个数据包划分为多个分组,每个分组包括至少一个数据包;

所述第二发送模块包括:

第一判断单元,接收到所述服务器发送的分组上传提示后,判断所述待上传文件中所有的数据包个数是否大于或等于分组要求的分包个数;

第一发送单元,若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包;

第二发送单元,所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包;

第三发送单元,若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

13. 根据权利要求11所述的用户终端,其特征在于,所述第一发送模块包括:

第四发送单元,用于向所述服务器发送文件上传请求;

等待单元,用于等待所述服务器发送的上传提示;

第五发送单元,用于若未收到所述上传提示,则根据预置的最大发送文件上传请求次数,向所述服务器发送文件上传请求;

中断单元,用于当超过所述预置的最大发送文件上传请求次数时,则中断文件上传。

14. 根据权利要求12所述用户终端,其特征在于,所述用户终端还包括:

第三发送模块,若未接收到所述服务器发送分组上传完成提示,则重新向所述服务器发送文件上传请求;

第二接收模块,用于接收所述服务器发送未收到的数据包的分包号;

第四发送模块,用于根据所述未收到的数据包的分包号发送所述分包号对应的数据包。

15. 一种服务器,其特征在于,包括:

第三接收模块,用于接收用户终端发送的文件上传请求;

第五发送模块,用于根据所述文件上传请求,向所述用户终端发送文件分组上传提示;

第四接收模块,用于接收所述用户终端发送的多个数据包;

第二确定模块,用于根据每个数据包中包含的各自的分包号与所述文件上传请求中携带的待上传文件划分得到的数据包总个数,确定所述待上传文件是否全部上传成功;

第二校验模块,用于当确定所述待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值;

第六发送模块,用于向所述用户终端发送第二校验值,以使所述用户终端在第一校验值和所述第二校验值相同时,确定所述待上传文件全部上传成功,所述第一校验值为对所述待上传文件进行完整性校验,得到的校验值。

16. 根据权利要求15所述的服务器,其特征在于,所述服务器还包括:

判断模块,用于根据所述多个分组,判断是否接收到每个分组中全部的数据包;

第七发送模块,用于若接收到每个分组中全部的数据包,则向所述用户终端发送上传完成提示;

等待模块,用于若没有接收到每个分组中全部的数据包,则继续等待接收所述每个分组中的剩余数据包。

17.根据权利要求16所述的服务器,其特征在于,所述服务器还包括:

提取模块,用于提取所述数据包中的冗余检验码CRC;

所述判断模块包括:

第二判断单元,用于当接收到多个分组中的每个数据包时,根据CRC验证码判断所述每个数据包中的数据是否正确;

第六发送单元,用于若分组中所述每个数据包的数据均正确,则向所述用户终端发送上传正确完成提示;

丢弃单元,用于若分组中有一个或多个数据包中的数据错误,则丢弃所述数据包。

18.一种文件上传的系统,其特征在于,包括:用户终端和服务器

所述用户终端为上述权利要求11至14中任一项所述的用户终端;

所述服务器为上述权利要求15至17中任一项所述的服务器。

一种文件上传的系统、方法及装置

技术领域

[0001] 本发明实施例涉及通信领域,尤其涉及一种文件上传的系统、方法及装置。

背景技术

[0002] 在现代通信技术领域中,数据的采集、信息的共享和文件的传输都显得至关重要。然而由于通信网络的信号和通信信道等因素的不确定性,使得通信系统在可靠性方面存在着一定的隐患,比如数据包丢失、数据包损坏或数据包欺骗等。由于数据包的长度有限制,当要传输大数据时,如大型文件,则需要将其进行分包来传输。

[0003] 现有技术中针对文件的传输主要有几种方案:

[0004] 1、发送方将文件拆分成固定大小的数据包,并对每个数据包添加数据包号,每隔一定的时间根据数据包号由小到大的顺序传输数据包到接收方,接收方在收到最大数据包号或在规定时间到了之后向发送方发送确认报文。

[0005] 2、借鉴用户数据报协议(UDP, User Datagram Protocol)报文传输时的分包、组包与丢包重传机制,接收方每收到一个数据包就回确认报文,直到所有数据包发送完毕。

[0006] 3、在文件传输过程中采用哈希算法对文件内容进行校验,再利用哈希算法对该计算的哈希值进行二次校验。

[0007] 上述两种现有技术中,虽然第一种方案的数据包开销小,但发送方无法实时了解数据包传输的状态。而第二种方案虽然可以实时了解数据包传输的状态,但是对数据包的数量要求很大。两种方案都没有对整个文件的完整性进行校验,无法判断文件在传输过程中内容是否遭到了破坏,第三种方案能够大大提升文件传输的可靠性,但相对来说略显复杂,增加了计算哈希值的开销。

发明内容

[0008] 本发明实施例提供了一种文件上传的系统、方法及装置,该文件上传的系统可用于进行完整性校验、划分数据包处理、分组发送数据包、实时丢包重传及响应,能够有效地加强数据传输的可靠性,而且实现简单,链路开销小。

[0009] 有鉴于此,本发明第一方面提供一种文件上传的系统,包括:用户终端与服务器;

[0010] 所述用户终端用于发送文件;

[0011] 所述服务器用于接收所述文件;

[0012] 所述文件上传的系统用于进行完整性校验、划分数据包处理、分组发送数据包、实时丢包重传及响应;

[0013] 所述用户终端对待上传文件进行所述完整性校验,得到第一校验值;

[0014] 所述用户终端将所述待上传文件分为多个数据包,并将所述多个数据包划分为多个分组,每个分组包括至少一个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器确定是否存在丢失的数据包;

[0015] 所述用户终端向所述服务器发送文件上传请求,所述文件上传请求中携带所述待

上传文件划分得到的数据包总个数以及分组个数；

[0016] 当所述服务器接收所述用户终端发送的所述文件上传请求后,向所述用户终端发送上传提示；

[0017] 所述用户终端接收到所述服务器发送的上传提示后,向所述服务器发送分组中的数据包；

[0018] 当所述服务器接收到所述分组的全部数据包时,向所述用户终端发送分组上传完成提示；

[0019] 若所述用户终端未接收到分组上传完成提示,则所述用户终端向所述服务器发送请求报文；

[0020] 所述服务器根据所述用户终端发送的请求报文回传分组中未收到数据包的分包号；

[0021] 所述用户终端根据所述分组中未收到数据包的分包号向所述服务器重新上传所述分包号对应的数据包；

[0022] 当确定所述待上传文件全部上传成功后,所述服务器对已经全部上传成功的文件进行完整性校验,得到第二校验值；

[0023] 当所述第一校验值与所述第二校验值相同时,用户终端确定所述待上传文件全部上传成功。

[0024] 结合本发明实施例的第一方面,在第一种可能的实现方式中,所述当所述服务器接收所述用户终端发送的所述文件上传请求后,向所述用户终端发送上传提示之前,所述系统还包括：

[0025] 若所述用户终端未接收到所述服务器发送的上传提示,则继续发送文件上传请求；

[0026] 用户终端判断所述文件上传请求次数是否超过预置发送次数；

[0027] 若未超过,则所述用户终端重新向服务器发送文件上传请求；

[0028] 若超过,则所述用户终端终止所述待上传文件的上传。

[0029] 结合本发明实施例的第一方面,在第二种可能的实现方式中,所述当所述服务器接收到所述每个分组的全部数据包时,向所述用户发送分组上传完成提示之后,所述系统还包括：

[0030] 当所述用户终端接收到所述分组上传完成提示时,判断剩余数据包个数是否大于每组中数据包的总数；

[0031] 若剩余数据包个数大于每组中数据包的总数,则所述用户终端根据分包号向所述服务器发送数据包；

[0032] 若剩余数据包个数小于等于每组中数据包的总数,则所述用户终端发送所有剩余的数据包。

[0033] 本发明第二方面提供一种文件上传的方法,包括：

[0034] 对待上传文件进行完整性校验,得到第一校验值；

[0035] 将所述待上传文件划分为多个数据包；

[0036] 向服务器发送文件上传请求,所述文件上传请求中携带所述待上传文件划分得到的数据包总个数；

[0037] 接收到所述服务器发送的上传提示后,向所述服务器发送所述多个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器根据所述数据包的总个数确定所述待上传文件是否全部上传成功;

[0038] 当所述待上传文件中所有的数据包上传成功后,接收所述服务器发送的第二校验值,所述第二校验值为所述服务器确定所述待上传文件全部上传完成后,对所述待上传文件进行完整性校验得到的;

[0039] 当所述第一校验值与所述第二校验值相同时,确定所述待上传文件全部上传成功。

[0040] 结合本发明实施例的第二方面,在第一种可能的实现方式中,所述将待上传文件划分为多个数据包之后,还包括:

[0041] 将所述多个数据包划分为多个分组,每个分组包括至少一个数据包;

[0042] 所述接收到所述服务器发送的上传提示后,向所述服务器发送所述多个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器根据所述数据包的总个数确定所述待上传文件是否全部上传成功包括:

[0043] 接收到所述服务器发送的分组上传提示后,判断所述待上传文件中所有的数据包个数是否大于分组要求的分包个数;

[0044] 若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包;

[0045] 所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包;

[0046] 若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

[0047] 结合本发明实施例的第二方面,在第二种可能的实现方式中,所述向服务器发送文件上传请求,所述文件上传请求中携带所述待上传文件划分得到的数据包总个数包括:

[0048] 向所述服务器发送文件上传请求;

[0049] 等待所述服务器发送的上传提示;

[0050] 若未收到上传提示,则根据预置的最大发送文件上传请求次数,向所述服务器发送文件上传请求;

[0051] 当超过预置的最大发送文件上传请求次数时,则中断文件上传。

[0052] 结合本发明实施例的第二方面的第一种可能的实现方式,在第三种可能的实现方式中,所述若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包之后,还包括:

[0053] 若未接收到所述服务器发送分组上传完成提示,则重新向所述服务器发送文件上传请求;

[0054] 接收所述服务器发送未收到的数据包的分包号;

[0055] 根据所述未收到的数据包的分包号发送所述分包号对应的数据包。

[0056] 本发明第三方面提供一种文件上传的方法,包括:

[0057] 接收所述用户终端发送的所述文件上传请求;

[0058] 根据所述文件上传请求,向所述用户终端发送所述文件分组上传提示;

[0059] 接收所述用户终端发送的所述多个数据包;

[0060] 根据每个数据包中包含的各自的分包号与所述文件上传请求中携带的所述待上传文件划分得到的数据包总个数,确定所述待上传文件是否全部上传成功;

[0061] 当确定所述待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值;

[0062] 向所述用户终端发送所述第二校验值。

[0063] 结合本发明实施例的第三方面,在第一种可能的实现方式中,所述接收所述用户终端发送的所述多个数据包之后,还包括:

[0064] 根据所述多个分组,判断是否接收到每个分组中全部的数据包;

[0065] 若接收到每个分组中全部的数据包,则向所述用户终端发送上传完成提示;

[0066] 若没有接收到每个分组中全部的数据包,则继续等待接收所述每个分组中的剩余数据包。

[0067] 结合本发明实施例的第三方面或第三方面第一种可能的实现方式,在第二种可能的实现方式中,所述接收所述用户终端发送的所述多个数据包之后,还包括:

[0068] 提取所述数据包中的冗余检验码CRC;

[0069] 所述根据所述多个分组,判断是否接收到每个分组中全部的数据包包括:

[0070] 当接收到多个分组中的每个数据包时,根据CRC验证码判断所述每个数据包中的数据是否正确;

[0071] 若分组中所述每个数据包的数据均正确,则向所述用户终端发送上传正确完成提示;

[0072] 若分组中有一个或多个数据包中的数据错误,则丢弃所述数据包。

[0073] 本发明第四方面提供一种用户终端,包括:

[0074] 第一校验模块,用于对待上传文件进行完整性校验,得到第一校验值;

[0075] 第一划分模块,用于将所述待上传文件划分为多个数据包;

[0076] 第一发送模块,用于向服务器发送文件上传请求,所述文件上传请求中携带所述待上传文件划分得到的数据包总个数;

[0077] 第二发送模块,用于接收到所述服务器发送的上传提示后,向所述服务器发送所述多个数据包,每个数据包中包含各自的分包号,所述分包号用于所述服务器根据所述数据包的总个数确定所述待上传文件是否全部上传成功;

[0078] 第一接收模块,用于当所述待上传文件中所有的数据包上传成功后,接收所述服务器发送的第二校验值,所述第二校验值为所述服务器确定所述待上传文件全部上传完成后,对所述待上传文件进行完整性校验得到的;

[0079] 第一确定模块,用于当所述第一校验值与所述第二校验值相同时,确定所述待上传文件全部上传成功。

[0080] 结合本发明实施例的第四方面,在第一种可能的实现方式中,所述用户终端还包括:

[0081] 第二划分模块,用于将所述多个数据包划分为多个分组,每个分组包括至少一个数据包;

[0082] 所述第二发送模块包括:

[0083] 第一判断单元,接收到所述服务器发送的分组上传提示后,判断所述待上传文件

中所有的数据包个数是否大于或等于分组要求的分包个数；

[0084] 第一发送单元,若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包；

[0085] 第二发送单元,所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包；

[0086] 第三发送单元,若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

[0087] 结合本发明实施例的第四方面,在第二种可能的实现方式中,所述第一发送模块包括：

[0088] 第四发送单元,用于向所述服务器发送文件上传请求；

[0089] 等待单元,用于等待所述服务器发送的上传提示；

[0090] 第五发送单元,用于若未收到所述上传提示,则根据预置的最大发送文件上传请求次数,向所述服务器发送文件上传请求；

[0091] 中断单元,用于当超过所述预置的最大发送文件上传请求次数时,则中断文件上传。

[0092] 结合本发明实施例的第四方面第一种可能的实现方式,在第三种可能的实现方式中,所述用户终端还包括：

[0093] 第三发送模块,若未接收到所述服务器发送分组上传完成提示,则重新向所述服务器发送文件上传请求；

[0094] 第二接收模块,用于接收所述服务器发送未收到的数据包的分包号；

[0095] 第四发送模块,用于根据所述未收到的数据包的分包号发送所述分包号对应的数据包。

[0096] 本发明第五方面提供一种服务器,包括：

[0097] 第三接收模块,用于接收所述用户终端发送的所述文件上传请求；

[0098] 第五发送模块,用于根据所述文件上传请求,向所述用户终端发送所述文件分组上传提示；

[0099] 第四接收模块,用于接收所述用户终端发送的所述多个数据包；

[0100] 第二确定模块,用于根据每个数据包中包含的各自的分包号与所述文件上传请求中携带的所述待上传文件划分得到的数据包总个数,确定所述待上传文件是否全部上传成功；

[0101] 第二校验模块,用于当确定所述待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值；

[0102] 第六发送模块,用于向所述用户终端发送第二校验值。

[0103] 结合本发明实施例的第五方面,在第一种可能的实现方式中,所述服务器还包括：

[0104] 判断模块,用于根据所述多个分组,判断是否接收到每个分组中全部的数据包；

[0105] 第七发送模块,用于若接收到每个分组中全部的数据包,则向所述用户终端发送上传完成提示；

[0106] 等待模块,用于若没有接收到每个分组中全部的数据包,则继续等待接收所述每个分组中的剩余数据包。

[0107] 结合本发明实施例的第五方面或第五方面第一种可能的实现方式,在第二种可能的实现方式中,所述服务器还包括:

[0108] 提取模块,用于提取所述数据包中的冗余检验码CRC;

[0109] 所述判断模块包括:

[0110] 第二判断单元,用于当接收到多个分组中的每个数据包时,根据CRC验证码判断所述每个数据包中的数据是否正确;

[0111] 第六发送单元,用于若分组中所述每个数据包的数据均正确,则向所述用户终端发送上传正确完成提示;

[0112] 丢弃单元,用于若分组中有一个或多个数据包中的数据错误,则丢弃所述数据包。

[0113] 从以上技术方案可以看出,本发明实施例具有以下优点:

[0114] 本发明实施例中,提出一种文件上传的系统,该系统包括用户终端与服务器,在上传文件前后对待上传文件进行完整性校验,可以保证文件传输的完整性,而在文件上传时,还可以将待上传文件划分为多个数据包和多个分组,按组上传文件,收到分组上传完成提示后发出响应,若没有收到该提示,则实时将未收到数据包重新上传,这样进一步的保证了文件传输的完整性。

附图说明

[0115] 图1A为本发明实施例中文件上传系统一个实施例示意图;

[0116] 图1为本发明实施例中文件上传的方法一个实施例示意图;

[0117] 图2为本发明实施例中文件上传的方法另一个实施例示意图;

[0118] 图3为本发明实施例中用户终端一个实施例示意图;

[0119] 图4为本发明实施例中用户终端另一个实施例示意图;

[0120] 图5为本发明实施例中用户终端另一个实施例示意图;

[0121] 图6为本发明实施例中用户终端另一个实施例示意图;

[0122] 图7为本发明实施例中服务器一个实施例示意图;

[0123] 图8为本发明实施例中服务器另一个实施例示意图;

[0124] 图9为本发明实施例中服务器另一个实施例示意图;

[0125] 图10为本发明实施例中文件上传系统一个实施例示意图。

具体实施方式

[0126] 本发明实施例提供了一种文件上传的系统、方法及装置,用于提高文件的上传的成功率和加大上传文件内容的准确性。

[0127] 请参阅图1A,本发明实施例中文件上传系统一个实施例包括:

[0128] S101、用户终端对待上传文件进行完整性校验,得到第一校验值;

[0129] 本实施例中,用户终端对待上传的文件进行完整性校验,并在检验后获得第一校验值。

[0130] S102、用户终端将待上传文件分为多个数据包,并将多个数据包划分为多个分组,每个分组包括至少一个数据包,每个数据包中包含各自的分包号,分包号用于服务器确定是否存在丢失的数据包;

[0131] 本实施例中,在完整性校验结束后,用户终端可以将待上传文件划分为多个数据包,再将多个数据包划分为多个分组,且每个分组包括至少一个数据包,每个数据包中包含了各自的分包号,数据包分包号初值可以设为1,分包号用于让服务器确定是否存在丢失的一个或多个数据包。

[0132] S103、用户终端向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数以及分组个数;

[0133] 本实施例中,用户终端在完成分包和分组后,可以向服务器发送文件上传请求,同时可以在发送请求后记录请求的次数,并等待服务器返回是否准备好接收数据包的响应,而文件上传请求中包含了数据包总个数以及分组个数,也可以包括待上传文件的大小。

[0134] S104、当服务器接收用户终端发送的文件上传请求后,向用户终端发送上传提示;

[0135] 本实施例中,当服务器接收到用户终端发送的文件上传请求后,可以根据自身的工作状态决定是否向用户终端发送准备好接收数据包的响应,服务器工作状态良好的情况下,将向用户终端发送上传提示。

[0136] S105、用户终端接收到服务器发送的上传提示后,向服务器发送分组中的数据包;

[0137] 本实施例中,当用户终端接收到服务器发送的上传提示后,向服务器发送其中一个分组中的数据包,且可以根据分包号从小到大的顺序向服务器依次发送数据包,直到发送了该组中所有的数据包,每次发送一个数据包后,分包号也应该加1。

[0138] S106、当服务器接收到分组的全部数据包时,向用户终端发送分组上传完成提示;

[0139] 本实施例中,当服务器收到该分组中所有的数据包后,则返回该组数据包接收成功的响应,即向用户终端发送分组上传完成提示,如果服务器没有接收到该分组中所有的数据包时,则会继续等待接收。

[0140] S107、若用户终端未接收到分组上传完成提示,则用户终端向服务器发送请求报文;

[0141] 本实施例中,如果用户终端在一段时间内都未收到由服务器发送的分组上传完成提示,则会向服务器发送请求报文,并且可以重新记录请求次数。

[0142] S108、服务器根据用户终端发送的请求报文回传分组中未收到数据包的分包号;

[0143] 本实施例中,当服务器接收到用户终端发送的请求报文后,将遍历分包号,并回传该分组中没有收到的分包号。

[0144] S109、用户终端根据分组中未收到数据包的分包号向服务器重新上传分包号对应的数据包;

[0145] 本实施例中,用户终端根据该分组中没有收到的数据包的分包号,重新上传分包号对应的数据包,直至该分组中所有数据包都成功上传至服务器。

[0146] S110、当确定待上传文件全部上传成功后,服务器对已经全部上传成功的文件进行完整性校验,得到第二校验值;

[0147] 本实施例中,当确定待上传文件划分得到的各个分组的所有数据包都成功上传后,服务器将会对已经收到的上传文件进行完整性校验,并且得到第二校验值。

[0148] S111、当第一校验值与第二校验值相同时,用户终端确定待上传文件全部上传成功。

[0149] 本实施例中,服务器将第二校验值返回给用户终端,当第一校验值与第二校验值

一致时,则可以确定待上传文件全部上传成功。

[0150] 本发明实施例中,提出一种文件上传的系统,该系统包括用户终端与服务器,在上传文件前后对待上传文件进行完整性校验,可以保证文件传输的完整性,而在文件上传时,还可以将待上传文件划分为多个数据包和多个分组,按组上传文件,收到分组上传完成提示后发出响应,若没有收到该提示,则实时将未收到数据包重新上传,这样进一步的保证了文件传输的完整性。

[0151] 可选地,在上述图1A对应的实施例的基础上,本发明实施例提供的文件上传系统的第一个可选实施例中,当服务器接收用户终端发送的文件上传请求后,向用户终端发送上传提示之前,系统还可以包括:

[0152] 若用户终端未接收到服务器发送的上传提示,则继续发送文件上传请求;

[0153] 用户终端判断文件上传请求次数是否超过预置发送次数;

[0154] 若未超过,则用户终端重新向服务器发送文件上传请求;

[0155] 若超过,则所述用户终端终止所述待上传文件的上传。

[0156] 本实施例中,当服务器接收到用户终端发送的文件上传请求后,由于自身的工作状态不佳,则不向用户终端发送上传提示,而用户终端在没有接收的上传提示的情况下,会继续向服务器发送文件上传请求,当判断得到请求的次数已经超过预置发送请求的次数时,则用户终端会终止对该文件的上传,如果未超过预置次数,则会继续发送文件上传请求。

[0157] 其次,本发明实施例中,用户终端在向服务器上传文件的时候,可以根据服务器的工作状态来决定是否适合上传该文件,这样可以有效地减轻服务器工作负荷,同时避免了上传过程中出现堵塞的情况,使得方案在使用的时候更具有实践性。

[0158] 可选地,在上述图1A对应的实施例的基础上,本发明实施例提供的文件上传系统的第二个可选实施例中,当服务器接收到每个分组的全部数据包时,向用户发送分组上传完成提示之后,系统还可以包括:

[0159] 当用户终端接收到分组上传完成提示时,判断剩余数据包个数是否大于每组中数据包的总数;

[0160] 若剩余数据包个数大于每组中数据包的总数,则用户终端根据分包号向服务器发送数据包;

[0161] 若剩余数据包个数小于等于每组中数据包的总数,则用户终端发送所有剩余的数据包。

[0162] 本实施例中,当服务器收到一个分组中所有的数据包之后,将向用户终端发送分组上传完成提示,用户终端收到该提示后,将会判断剩余数据包个数是否大于每组中的数据包个数,如果余下的数据包个数大于每个分组中的数据包个数,则用户终端根据分包号由小至大的顺序向服务器发送下一分组中所有的数据包,如果余下数据包个数小于或等于每组中数据包的总数时,则用户终端可以一次性发送所有剩余的数据包。

[0163] 需要说明的是,当余下数据包个数等于每个分组中所有数据包的个数时,也可以选择把余下的数据包划分为另一分组,并上传该组中的数据包,故此处不作限定。

[0164] 再次,本发明实施例中,用户终端根据余下数据包个数来决定是否需要重新进入下一轮的分组上传,如果余下的数据包个数小于或等于分组中所有数据包个数时,可以直

接进行一次上传,这样可以不需要浪费多余的时间去完成分组上传的过程,而是直接根据实际情况进行一次剩余数据包的上传,既可以节约上传时间,又可以节省计算资源。

[0165] 上述实施例是从本发明的系统角度进行描述,下面将对本发明中的文件上传方法进行详细描述,请参阅图1,本发明实施例中文件上传的方法一个实施例包括:

[0166] 101、对待上传文件进行完整性校验,得到第一校验值;

[0167] 本实施例中,用户终端对待上传的文件先进行完整性校验,进行完整性校验为了判断文件传输是否成功,校验后可以获得第一校验值。

[0168] 需要说明的是,本实施例中,待上传文件的格式可以是任意类型的文件,故此处不对上传文件的格式做限定。

[0169] 需要说明的是,本实施例中可以采用哈希函数中的安全散列算法(SHA, Secure Hash Algorithm),例如SHA-0、SHA-1或SHA-2,也可以采用哈希函数中的信息摘要算法(MD, Message-Digest Algorithm),例如MD4或MD5,故此处不做限定。

[0170] 102、将待上传文件划分为多个数据包;

[0171] 本实施例中,用户终端将待上传文件划分为多个数据包,每个数据包以一定的字节数为单位进行分包。

[0172] 103、向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数;

[0173] 本实施例中,用户终端向服务器提出上传文件的请求,并记录请求的次数,向服务器发送的文件上传请求中携带了待上传文件划分后得到的数据包总个数。

[0174] 104、接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功;

[0175] 本实施例中,当用户终端接收到服务器发送的上传提示后,再向服务器发送多个数据包,其中每个数据包中包含了各自的分包号,分包号初值设定为1,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功。

[0176] 105、当待上传文件中所有的数据包上传成功后,接收服务器发送的第二校验值,第二校验值为服务器确定待上传文件全部上传成功后,对待上传文件进行完整性校验得到的;

[0177] 本实施例中,用户终端接收服务器发送的第二校验值,第二校验值是服务器确定待上传文件都上传后,再对待上传文件进行完整性检验,并将检验得到的第二校验值发送至用户终端。

[0178] 106、当第一校验值与第二校验值相同时,确定待上传文件全部上传成功。

[0179] 本实施例中,用户终端将获取到的第一校验值与获取到的第二校验值进行比对,当两者相同时,可以确定待上传文件已全部上传成功。

[0180] 哈希算法将任意长度的二进制值映射为固定长度的较小二进制值,而哈希值则是哈希算法中较小的二进制值,哈希值是一段数据中唯一的,并且极其紧凑的数值表示形式,如果散列一段明文且只更改该段落的一个字母,随后的哈希都将产生不同的值,因此要找到散列为同一个值的两个不同的输入,在计算上来说基本上是不可能的。

[0181] 本发明实施例中,用户终端对待上传文件先进行完整性校验,得到第一校验值,等

接收分组上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功,全部文件上传后,接收服务器发送的第二校验值,当第一校验值与第二校验值相同时,确定待上传文件全部上传成功,这样可以提高文件的上传的成功率以及加大上传文件内容的准确性。

[0182] 可选地,在上述图1对应的实施例的基础上,本发明实施例提供的文件上传的方法的第一个可选实施例中,将待上传文件划分为多个数据包之后,还可以包括:

[0183] 将多个数据包划分为多个分组,每个分组包括至少一个数据包;

[0184] 接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功包括:

[0185] 接收到所述服务器发送的分组上传提示后,判断所述待上传文件中所有的数据包个数是否大于分组要求的分包个数;

[0186] 若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包;

[0187] 所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包;

[0188] 若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

[0189] 本发明实施例中,用户终端将待上传文件以N个字节为单位进行分包,若是文件大小不足N个字节,则将文件内容封装到一个数据包中,只需要发送一个数据即可实现文件的上传。

[0190] 本发明实施例中,用户终端将多个数据包划分为多个组,并以M个数据包为单位进行划分,分组个数是由总的数据包个数除以M后再加1所得到的,加1是因为可能存在最后所剩余的数据包总数小于每组的数据包个数,用户终端完成分包和分组之后,会向服务器发送文件上传的请求,该请求中携带了待上传文件划分得到的数据包总个数和分组的个数。

[0191] 其次,本实施例中,将数据包分组进行传输,通过分包以及分组上传可以及时对传输过程中出现的数据包丢失做出响应,对丢失的数据包重新上传,这样可以有效的提高文件上传的成功率,同时提升方案的灵活性。

[0192] 可选地,在上述图1对应的实施例的基础上,本发明实施例提供的文件上传的方法的第二个可选实施例中,向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数可以包括:

[0193] 向服务器发送文件上传请求;

[0194] 等待服务器发送的上传提示;

[0195] 若未收到上传提示,则根据预置的最大发送文件上传请求次数,向服务器发送文件上传请求;

[0196] 当超过预置的最大发送文件上传请求次数时,则中断文件上传。

[0197] 本发明实施例中,用户终端向服务器提出上传文件的请求,并记录请求的次数,向服务器发送的文件上传请求中携带了待上传文件划分后得到的数据包总个数。

[0198] 本发明实施例中,用户终端在向服务器发送文件上传请求后,等待服务器发送可以上传文件的提示,只有与服务器成功建立连接后才能进行文件的上传,当用户终端判断

一段时间之内没有收到服务器发送的上传提示,则会根据预置的最大发送文件上传请求次数,向该服务器再次发送文件上传请求。

[0199] 需要说明的是,预置的最大发送文件上传请求次数可以为一次,或一次以上,预设最大发送文件上传请求次数的方式也有多种,可以通过接口编译,也可以是其他途径,此处不做限定。当用户终端向服务器发送的文件上传请求次数已经超过预置的最大发送文件上传请求次数时,则停止上传该文件。

[0200] 再次,本实施例中,用户终端可以实时了解文件存储的状态,若服务器接收缓存区已满,或者出现接收异常的情况,则停止用户终端继续上传文件,这样可以很好的解决接收缓存区溢出的问题,或者其他安全问题,大大提升了方案的实用性。

[0201] 可选地,在本发明实施例提供的文件上传的方法第一个可选实施例的基础上,本发明实施例提供的文件上传的方法的第三个可选实施例中,若待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包之后,还可以包括:

[0202] 若未接收到服务器发送分组上传完成提示,则重新向服务器发送文件上传请求;

[0203] 接收服务器发送未收到的数据包的分包号;

[0204] 根据未收到的数据包的分包号发送分包号对应的数据包。

[0205] 本发明实施例中,当用户终端判断得到还未接收服务器发送分组上传完成提示时,则会重新向服务器发送文件上传请求,如果服务器没有收到用户终端的请求,不会主动去询问用户终端是否需要上传文件。

[0206] 需要说明的是,如果通过移动网络进行数据上传,服务器与用户终端之间需要外接短信猫,令服务器与用户终端具备发送和接收短消息的能力,还可以通过其他手段使服务器与用户终端拥有发送和接收信息的功能,此处不对具体手段做限定。

[0207] 本发明实施例中,服务器只有在收到用户终端发送的文件上传请求之后,才会遍历分包号数组,将未收到的数据包的分包号返回至用户终端,使得用户终端接收这些分包号,用户终端根据未收到的数据包的分包号,再将这些分包号对应的数据包重新发送至服务器。

[0208] 更进一步地,本实施例中,用户终端通过判断是否收到服务器发送的分组上传完成提示,来决定是否需要再次传输未上传成功的数据包,服务器在收到用户终端请求后遍历分包号,将未收到的分包号再反馈给用户终端,这样的实时丢包重传机制可以使文件传输的完整性得到保证,极大的提升了用户的体验。

[0209] 上面的实例中都是从用户终端的角度对本发明中文件上传的方法进行描述,下面将从服务器的角度对本发明中数据处理的方法进行描述。请参阅图2,本发明文件上传的方法的另一个实施例包括:

[0210] 201、接收用户终端发送的文件上传请求;

[0211] 本实施例中,当用户需要进行文件上传时,通过用户终端向服务器发送文件上传请求,服务器接收这个由用户终端发送的请求,服务器在进入接收状态之后,不会主动询问用户终端是否需要上传文件,而是等待用户终端发送文件上传的请求。

[0212] 202、根据文件上传请求,向用户终端发送文件分组上传提示;

[0213] 本实施例中,服务器根据用户终端发送的文件上传请求,向用户终端发送文件分

组上传提示,该提示用于指示用户终端一侧将已分好组的数据包,按一定顺序向服务器发送。

[0214] 203、接收用户终端发送的多个数据包;

[0215] 本实施例中,当用户终端根据文件分组上传提示向服务器发送数据包时,服务器会接收由用户终端发送的一组中多个数据包。

[0216] 204、根据每个数据包中包含的各自的分包号与文件上传请求中携带的待上传文件划分得到的数据包总个数,确定待上传文件是否全部上传成功;

[0217] 本实施例中,服务器与用户终端建立连接后开始接收用户终端发送的数据包,根据每个数据包中包含的各自分包号,将这些分包号与文件上传请求中携带的待上传文件被划分得到的总包数做比较,当每个分包号对应的数据包个数总和等于总包数时,则确定待上传文件已经全部上传成功,如果不相等时,则继续等待接收剩余的数据包。

[0218] 205、当确定待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值;

[0219] 本实施例中,当确定待上传文件全部上传成功完毕后,将数据包中的内容根据分包号先进行还原,得到还原后的文件,再对该文件进行完整性校验,采用哈希算法,得到的哈希值即为第二校验值。

[0220] 206、向用户终端发送第二校验值。

[0221] 本实施例中,服务器将得到的第二校验值通过结果报文返回给用户终端,使用户终端根据第一校验值与第二校验值的对比,判断待上传文件的完整性。

[0222] 本实施例中,服务器接收用户终端发起的请求,并根据用户终端对待上传文件的分组进行接收,同时确定待上传文件是否完整接收,最后发送第二校验值至用户终端,如此保证文件传输的完整性,与此同时,服务器只需要等待用户终端发出的相应请求,而不需要主动询问,这样可以减小服务器的压力,加快传输效率,提升用户体验。

[0223] 可选地,在上述图2对应的实施例的基础上,本发明实施例提供的文件上传的方法的第一个可选实施例中,接收用户终端发送的多个数据包之后,还可以包括:

[0224] 根据多个分组,判断是否接收到每个分组中全部的数据包;

[0225] 若接收到每个分组中全部的数据包,则向用户终端发送上传完成提示;

[0226] 若没有接收到每个分组中全部的数据包,则继续等待接收每个分组中的剩余数据包。

[0227] 本发明实施例中,服务器在接收了一个组内的多个数据包后,根据分包号来判断一组中的数据包是否完整,该组数据包接收成功的响应可以通过发送响应报文来告知用户终端,当判断得出没有接收到每个分组中的全部数据包时,服务器将继续等待用户终端发送每个分组中的剩余数据包,当判断得到已接收了每个分组中全部的数据包后,则向用户终端发送上传完成提示。

[0228] 其次,本实施例中,服务器可以根据多个分组判断是否接收到每个分组中全部的数据包,如果没有接收全部数据包,则继续等待接收剩余数据包,若是已接收全部数据包,就向用户终端发送上传完成提示,通过这种方法可以有效的保证待上传文件的完整性。

[0229] 可选地,在上述图2对应的实施例的基础上,本发明实施例提供的文件上传的方法的第二个可选实施例中,接收用户终端发送的多个数据包之后,还可以包括:

[0230] 提取数据包中的冗余检验码CRC;

[0231] 根据多个分组,判断是否接收到每个分组中全部的数据包包括:

[0232] 当接收到多个分组中的每个数据包时,根据CRC验证码判断所述每个数据包中的数据是否正确;

[0233] 若分组中所述每个数据包的数据均正确,则向所述用户终端发送上传正确完成提示;

[0234] 若分组中有一个或多个数据包中的数据错误,则丢弃所述数据包。

[0235] 本发明实施例中,服务器在接收了数据包后,提取数据包中的CRC校验码,采用CRC冗余校验对接收的数据包中一个或者多个数据包进行内容的检测。

[0236] 需要说明的是,在传输过程中采用CRC冗余校验来进行检测可以使用的生成多项式有CRC-12或CRC-16,也可以是CRC-CCITT或CRC-32,还可以是其他的多项式,故此处不做限定。

[0237] 本发明实施例中,当服务器接收到一个数据包时,服务器即根据CRC验证码来判断这个数据包中的数据是否正确,同时,用户终端也可以使用CRC验证码来判断每个数据包的正确性,当判断得到所有的数据包中的数据均正确时,则向用户终端发送上传完成的提示,当判断得到某一个,或者有多个数据包的数据错误时,则直接丢弃这些数据包。

[0238] 本实施例中,采用CRC冗余校验来验证数据包的内容是否正确,若不正确则直接丢弃该数据包,极大的增加了文件传输的可靠性,而且实现起来简单,解决了操作复杂的问题。

[0239] 为便于理解,下面以一个具体应用场景对本发明中一种文件上传的方法进行详细描述,具体为:

[0240] 用户甲需要将A文件上传至某网站服务器,首先用户甲将一个32字节大小的A文件读入到程序中,采用哈希算法对整个文件内容进行完整性校验,采用哈希函数SHA-256计算得到A文件的哈希值大小为X。之后将A文件以1个字节为单位分包,再以4个数据包为一组,平均分为了8个组,刚好没有剩余的字节内容。

[0241] 用户甲触发“上传”模块,向某网站服务器传递了A文件中数据包的报文类型、总包数、分包号、数据和CRC校验码,服务器根据这些信息通过对用户甲上传的请求,于是反馈“请上传”的提示给用户甲使用的计算机,使得计算机将数据包按8个小组从小到大依次向网站服务器传输。

[0242] 当网站服务器接收到每个数据包后,即通过CRC校验码对该数据包进行检验,确定这些数据包中的数据都是正确的,于是接收下一组的上述数据包,同样对剩下7个小组的所有数据包做相应的CRC校验码验证。

[0243] 当确定32个数据包全部收到,于是向用户甲使用的计算机发送上传完成提示,同时接收到服务器发来的哈希值,而哈希值大小为X,与一开始检测得到的哈希值完全一致,说明本次文件的上传时完整的,用户甲收到一个“已成功上传”提示。

[0244] 下面对本发明中的用户终端进行详细描述,请参阅图3,本发明实施例中的用户终端包括:

[0245] 第一校验模块301,用于对待上传文件进行完整性校验,得到第一校验值;

[0246] 第一划分模块302,用于将待上传文件划分为多个数据包;

[0247] 第一发送模块303,用于向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数;

[0248] 第二发送模块304,用于接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功;

[0249] 第一接收模块305,用于当所述待上传文件中所有的数据包上传成功后,接收服务器发送的第二校验值,第二校验值为服务器确定待上传文件全部上传完成后,对待上传文件进行完整性校验得到的;

[0250] 第一确定模块306,用于当第一校验值与第二校验值相同时,确定待上传文件全部上传成功。

[0251] 本实施例中,第一校验模块301对待上传文件进行完整性校验,得到第一校验值,第一划分模块302将待上传文件划分为多个数据包,第一发送模块303向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数,第二发送模块304接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功,当待上传文件中所有的数据包上传成功后,第一接收模块305接收服务器发送的第二校验值,第二校验值为服务器确定待上传文件全部上传完成后,对待上传文件进行完整性校验得到的,当第一校验值与第二校验值相同时,第一确定模块306确定待上传文件全部上传成功。

[0252] 本发明实施例中,用户终端对待上传文件先进行完整性校验,得到第一校验值,等接收分组上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功,全部文件上传后,接收服务器发送的第二校验值,当第一校验值与第二校验值相同时,确定待上传文件全部上传成功,这样可以提高文件的上传的成功率以及加大上传文件内容的准确性。

[0253] 请参阅图4,本发明用户终端的另一个实施例包括:

[0254] 第一校验模块401,用于对待上传文件进行完整性校验,得到第一校验值;

[0255] 第一划分模块402,用于将待上传文件划分为多个数据包;

[0256] 第二划分模块403,用于将第一划分模块402划分的多个数据包划分为多个分组,每个分组包括至少一个数据包;

[0257] 第一发送模块404,用于向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数与分组个数;

[0258] 第二发送模块405,用于接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功;

[0259] 第一接收模块406,用于当所述待上传文件中所有的数据包上传成功后,接收服务器发送的第二校验值,第二校验值为服务器确定待上传文件全部上传完成后,对待上传文件进行完整性校验得到的;

[0260] 第一确定模块407,用于当第一校验模块401校验的第一校验值与第一接收模块406接收的第二校验值相同时,确定待上传文件全部上传成功。

[0261] 本实施例中的第二发送模块405还可以进一步包括:

[0262] 第一判断单元4051,接收到所述服务器发送的分组上传提示后,判断所述待上传文件中所有的数据包个数是否大于分组要求的分包个数;

[0263] 第一发送单元4052,若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包;

[0264] 第二发送单元4053,所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包;

[0265] 第三发送单元4054,若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

[0266] 其次,本实施例中,将数据包分组进行传输,通过分包以及分组上传可以及时对传输过程中出现的数据包丢失做出响应,对丢失的数据包重新上传,这样可以有效的提高文件上传的成功率,同时提升方案的灵活性。

[0267] 请参阅图5,本发明用户终端的另一个实施例包括:

[0268] 第一校验模块501,用于对待上传文件进行完整性校验,得到第一校验值;

[0269] 第一划分模块502,用于将待上传文件划分为多个数据包;

[0270] 第一发送模块503,用于向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数;

[0271] 第二发送模块504,用于接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功;

[0272] 第一接收模块505,用于当所述待上传文件中所有的数据包上传成功后,接收服务器发送的第二校验值,第二校验值为服务器确定待上传文件全部上传完成后,对待上传文件进行完整性校验得到的;

[0273] 第一确定模块506,用于当第一校验模块501校验的第一校验值与第一接收模块505接收的第二校验值相同时,确定待上传文件全部上传成功。

[0274] 本实施例中的第一发送模块503还可以进一步包括:

[0275] 第四发送单元5031,用于向服务器发送文件上传请求;

[0276] 等待单元5032,用于等待服务器发送的上传提示;

[0277] 第五发送单元5033,用于若未收到上传提示,则根据预置的最大发送文件上传请求次数,向服务器发送文件上传请求;

[0278] 中断单元5034,用于当超过预置的最大发送文件上传请求次数时,则中断文件上传。

[0279] 再次,本实施例中,用户终端可以实时了解文件存储的状态,若服务器接收缓存区已满,或者出现接收异常的情况,则停止用户终端继续上传文件,这样可以很好的解决接收缓存区溢出的问题,或者其他安全问题,大大提升了方案的实用性。

[0280] 请参阅图6,本发明用户终端的另一个实施例包括:

[0281] 第一校验模块601,用于对待上传文件进行完整性校验,得到第一校验值;

[0282] 第一划分模块602,用于将待上传文件划分为多个数据包;

[0283] 第二划分模块603,用于将第一划分模块602划分的多个数据包划分为多个分组,每个分组包括至少一个数据包;

[0284] 第一发送模块604,用于向服务器发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数与分组个数;

[0285] 第二发送模块605,用于接收到服务器发送的上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功;

[0286] 第一接收模块606,用于当所述待上传文件中所有的数据包上传成功后,接收服务器发送的第二校验值,第二校验值为服务器确定待上传文件全部上传完成后,对待上传文件进行完整性校验得到的;

[0287] 第一确定模块607,用于当第一校验模块601校验的第一校验值与第一接收模块606接收的第二校验值相同时,确定待上传文件全部上传成功;

[0288] 第三发送模块608,若未接收到服务器发送分组上传完成提示,则重新向服务器发送文件上传请求;

[0289] 第二接收模块609,用于接收服务器发送未收到的数据包的分包号;

[0290] 第四发送模块610,用于根据未收到的数据包的分包号发送分包号对应的数据包。

[0291] 本实施例中的第二发送模块605还可以进一步包括:

[0292] 第一判断单元6051,接收到所述服务器发送的分组上传提示后,判断所述待上传文件中所有的数据包个数是否大于分组要求的分包个数;

[0293] 第一发送单元6052,若所述待上传文件中所有的数据包个数大于分组要求的分包个数时,则根据分包号由小至大向服务器依次发送分组中所有的数据包;

[0294] 第二发送单元6053,所述分组中所有的数据包发送完成后,依次发送下一组中所有的数据包;

[0295] 第三发送单元6054,若所述待上传文件中所有的数据包个数小于等于分组要求的分包个数时,则向所述服务器一次性发送所述所有的数据包。

[0296] 更进一步地,本实施例中,用户终端通过判断是否收到服务器发送的分组上传完成提示,来决定是否需要再次传输未上传成功的数据包,服务器在收到用户终端请求后遍历分包号,将未收到的分包号再反馈给用户终端,这样的实时丢包重传机制可以使文件传输的完整性得到保证,极大的提升了用户的体验。

[0297] 请参阅图7,本发明服务器的一个实施例包括:

[0298] 第三接收模块701,用于接收用户终端发送的文件上传请求;

[0299] 第五发送模块702,用于根据文件上传请求,向用户终端发送文件分组上传提示;

[0300] 第四接收模块703,用于接收用户终端发送的多个数据包;

[0301] 第二确定模块704,用于根据每个数据包中包含的各自的分包号与文件上传请求中携带的待上传文件划分得到的数据包总个数,确定待上传文件是否全部上传成功;

[0302] 第二校验模块705,用于当确定待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值;

[0303] 第六发送模块706,用于向用户终端发送第二校验值。

[0304] 本实施例中,第三接收模块701接收用户终端发送的文件上传请求,第五发送模块702根据文件上传请求,向用户终端发送文件分组上传提示,第四接收模块703接收用户终端发送的多个数据包,第二确定模块704根据每个数据包中包含的各自的分包号与文件上

传请求中携带的待上传文件划分得到的数据包总个数,确定待上传文件是否全部上传成功,当确定待上传文件全部上传成功后,第二校验模块705对已经全部上传成功的文件进行完整性校验,得到第二校验值,第六发送模块706向用户终端发送第二校验值。

[0305] 本实施例中,服务器接收用户终端发起的请求,并根据用户终端对待上传文件的分组进行接收,同时确定待上传文件是否完整接收,最后发送第二校验值至用户终端,如此保证文件传输的完整性,与此同时,服务器只需要等待用户终端发出的相应请求,而不需要主动询问,这样可以减小服务器的压力,加快传输效率,提升用户体验。

[0306] 请参阅图8,本发明服务器的另一个实施例包括:

[0307] 第三接收模块801,用于接收用户终端发送的文件上传请求;

[0308] 第五发送模块802,用于根据第三接收模块801接收的文件上传请求,向用户终端发送文件分组上传提示;

[0309] 第四接收模块803,用于接收用户终端发送的多个数据包;

[0310] 判断模块804,用于根据多个分组,判断是否接收到每个分组中全部的数据包;

[0311] 第七发送模块805,用于若接收到每个分组中全部的数据包,则向用户终端发送上传完成提示;

[0312] 等待模块806,用于若没有接收到每个分组中全部的数据包,则继续等待接收每个分组中的剩余数据包;

[0313] 第二校验模块807,用于当确定待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值;

[0314] 第六发送模块808,用于向用户终端发送第二校验值。

[0315] 其次,本实施例中,服务器可以根据多个分组判断是否接收到每个分组中全部的数据包,如果没有接收全部数据包,则继续等待接收剩余数据包,若是已接收全部数据包,就向用户终端发送上传完成提示,通过这种方法可以有效的保证待上传文件的完整性。

[0316] 请参阅图9,本发明服务器的另一个实施例包括:

[0317] 第三接收模块901,用于接收用户终端发送的文件上传请求;

[0318] 第五发送模块902,用于根据第三接收模块901接收的文件上传请求,向用户终端发送文件分组上传提示;

[0319] 第四接收模块903,用于接收用户终端发送的多个数据包;

[0320] 提取模块904,用于提取数据包中的冗余检验码CRC;

[0321] 判断模块905,用于根据多个分组,判断是否接收到每个分组中全部的数据包;

[0322] 第二校验模块906,用于当确定待上传文件全部上传成功后,对已经全部上传成功的文件进行完整性校验,得到第二校验值;

[0323] 第六发送模块907,用于向用户终端发送第二校验值。

[0324] 本实施例中的判断模块905还可以进一步包括:

[0325] 第二判断单元9051,用于当接收到的数据包个数等于数据包总个数且每个分组中的数据包个数为每个分组中的全部数据包个数时,根据CRC验证码判断数据包中的数据是否正确;

[0326] 第六发送单元9052,用于若分组中所述每个数据包的数据均正确,则向所述用户终端发送上传正确完成提示;

[0327] 丢弃单元9053,用于若分组中有一个或多个数据包中的数据错误,则丢弃所述数据包。

[0328] 本实施例中,采用CRC冗余校验来验证数据包的内容是否正确,若不正确则直接丢弃该数据包,极大的增加了文件传输的可靠性,而且实现起来简单,解决了操作复杂的问题。

[0329] 下面对本发明实施例中的文件上传系统进行描述,请参阅图10,本发明实施例中的文件上传系统包括:

[0330] 用户终端1001以及服务器1002;

[0331] 用户终端1001对待上传文件进行完整性校验,得到第一校验值,用户终端1001将待上传文件划分为多个数据包,用户终端1001向服务器1002发送文件上传请求,文件上传请求中携带待上传文件划分得到的数据包总个数,用户终端1001接收到服务器1002发送的上传提示后,向服务器1002发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器1002根据数据包的总个数确定待上传文件是否全部上传成功,当所述待上传文件中所有的数据包上传成功后,用户终端1001接收服务器1002发送的第二校验值,第二校验值为服务器1002确定待上传文件全部上传成功后,对待上传文件进行完整性校验得到的,当第一校验值与第二校验值相同时,用户终端1001确定待上传文件全部上传成功。

[0332] 接收用户终端发送的文件上传请求;

[0333] 服务器1002根据文件上传请求,向用户终端1001发送文件分组上传提示,服务器1002接收用户终端1001发送的多个数据包,服务器1002根据每个数据包中包含的各自的分包号与文件上传请求中携带的待上传文件划分得到的数据包总个数,确定待上传文件是否全部上传成功,当确定待上传文件全部上传成功后,服务器1002对已经全部上传成功的文件进行完整性校验,得到第二校验值,服务器1002向用户终端1001发送第二校验值。

[0334] 本发明实施例中,用户终端对待上传文件先进行完整性校验,得到第一校验值,等接收分组上传提示后,向服务器发送多个数据包,每个数据包中包含各自的分包号,分包号用于服务器根据数据包的总个数确定待上传文件是否全部上传成功,全部文件上传后,接收服务器发送的第二校验值,当第一校验值与第二校验值相同时,确定待上传文件全部上传成功,这样可以提高文件的上传的成功率以及加大上传文件内容的准确性。

[0335] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0336] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0337] 作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0338] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以

是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0339] 集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0340] 以上,以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

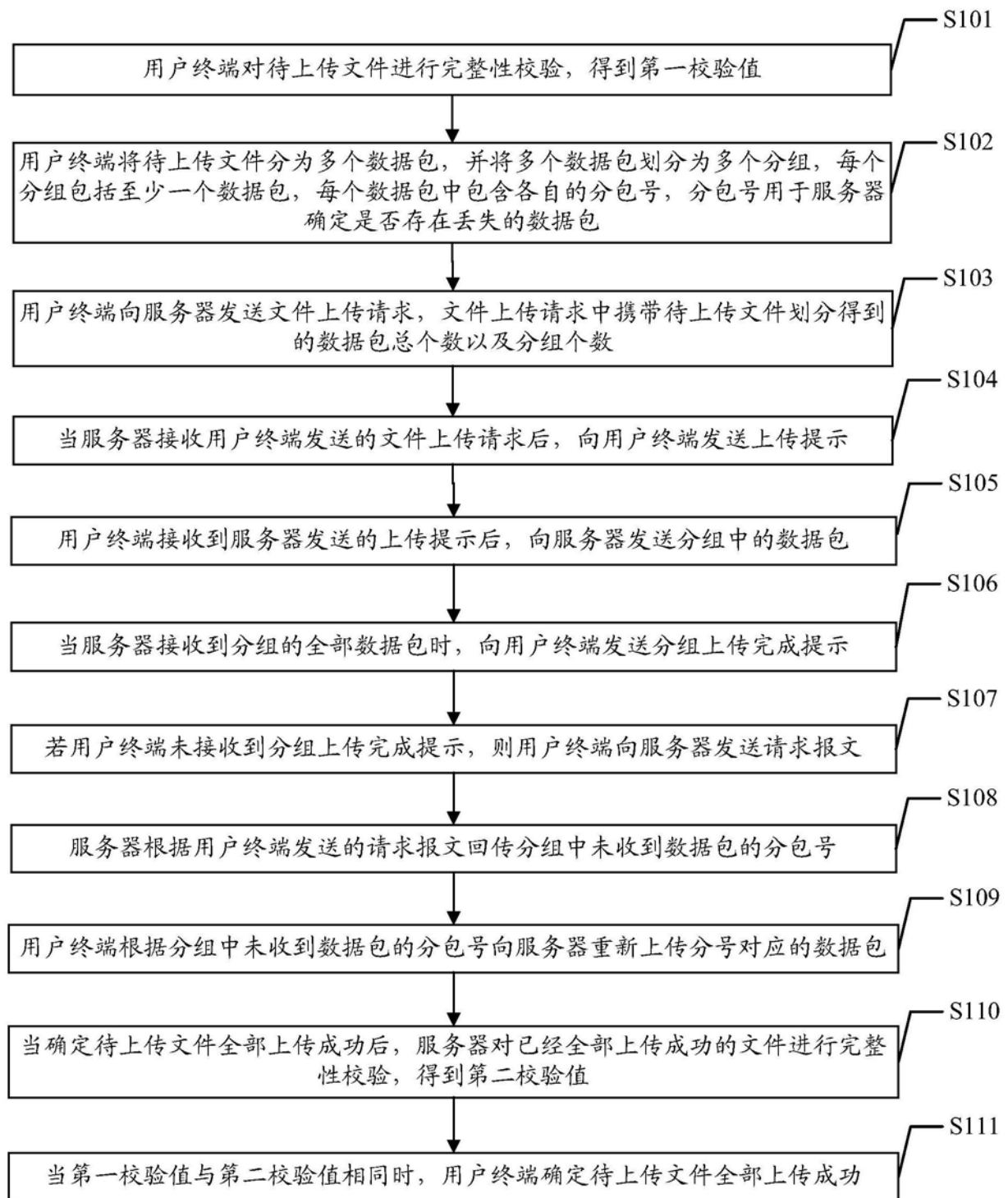


图1A

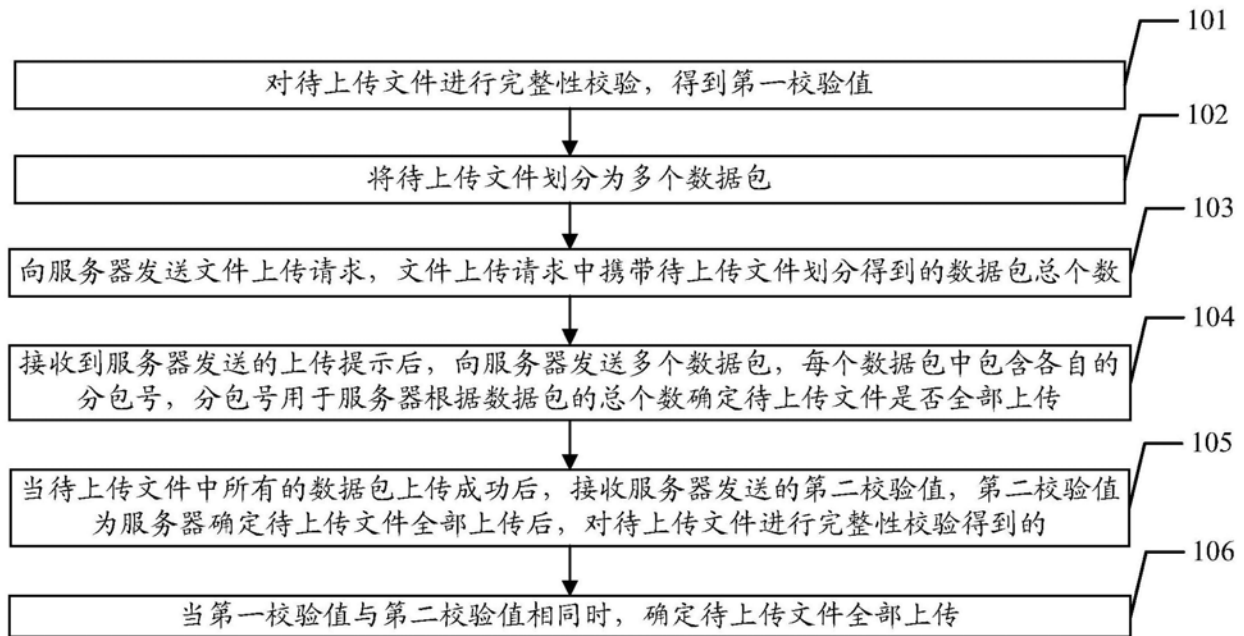


图1

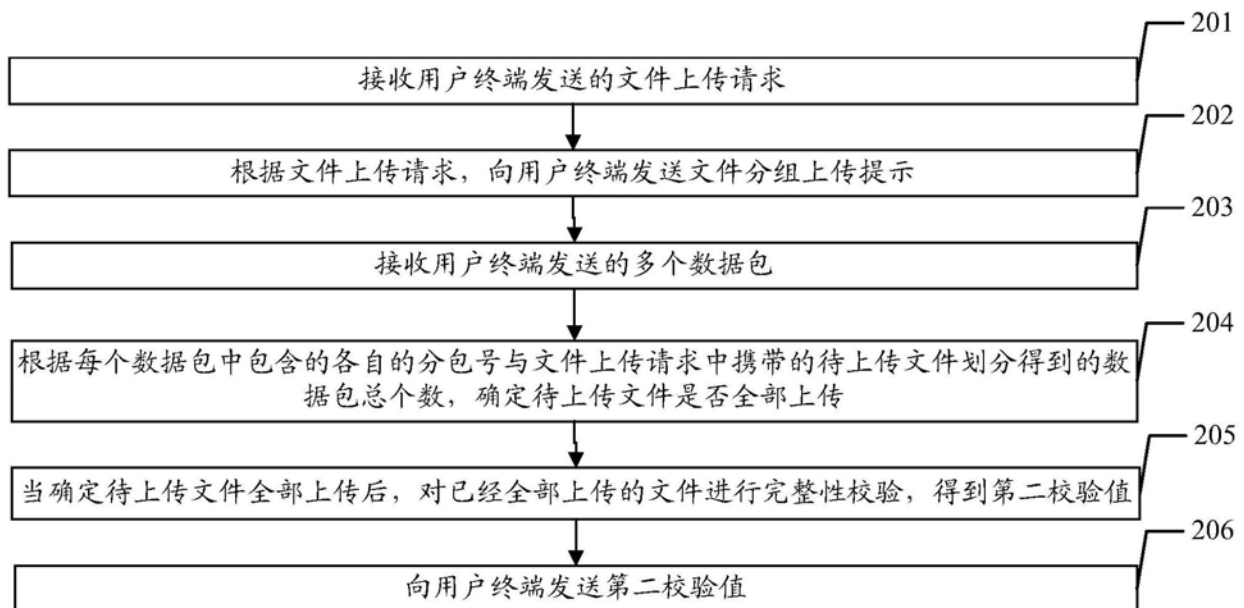


图2

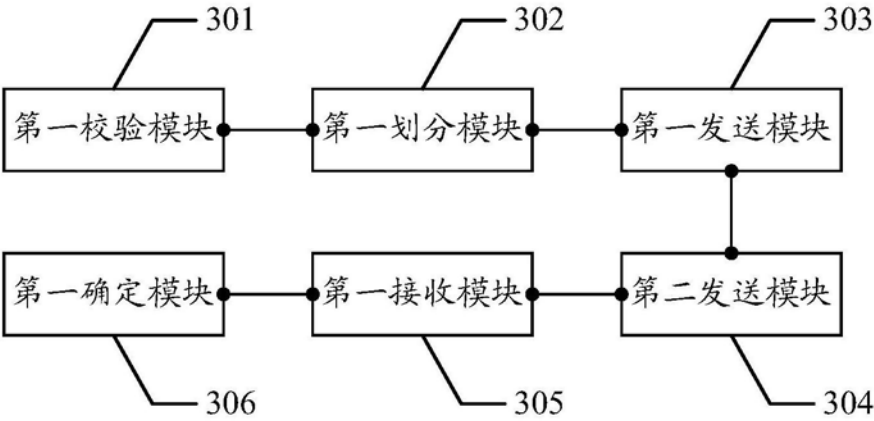


图3

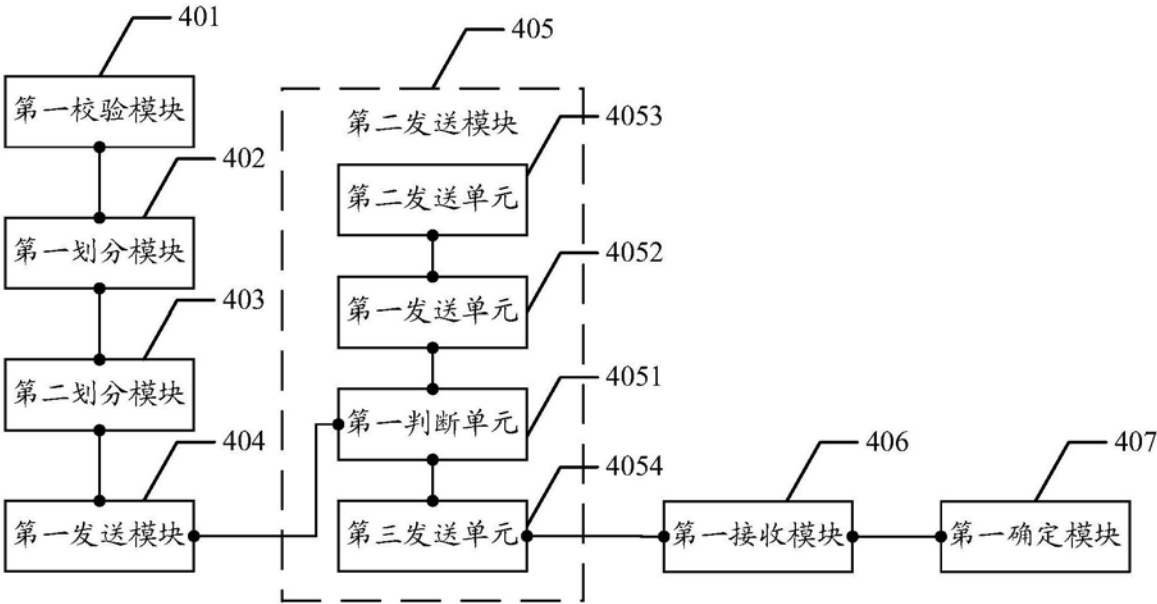


图4

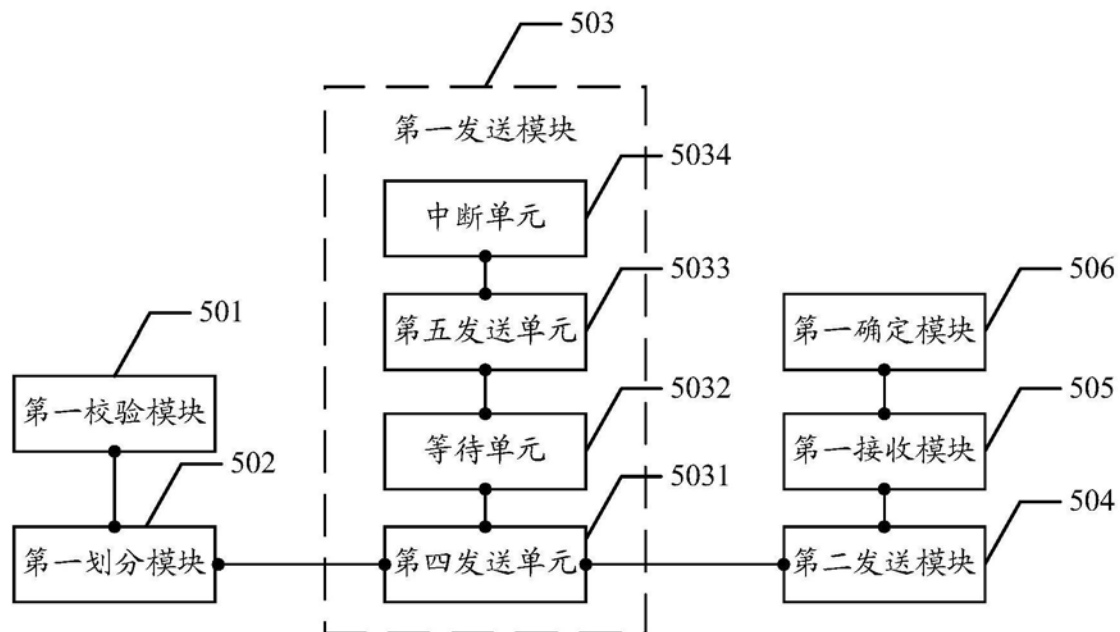


图5

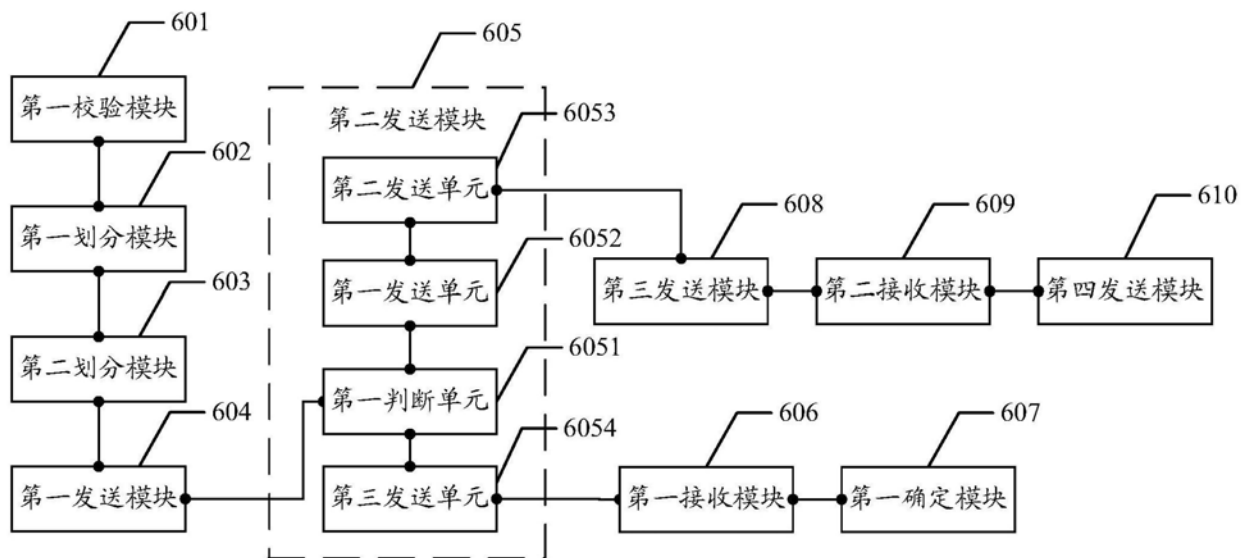


图6

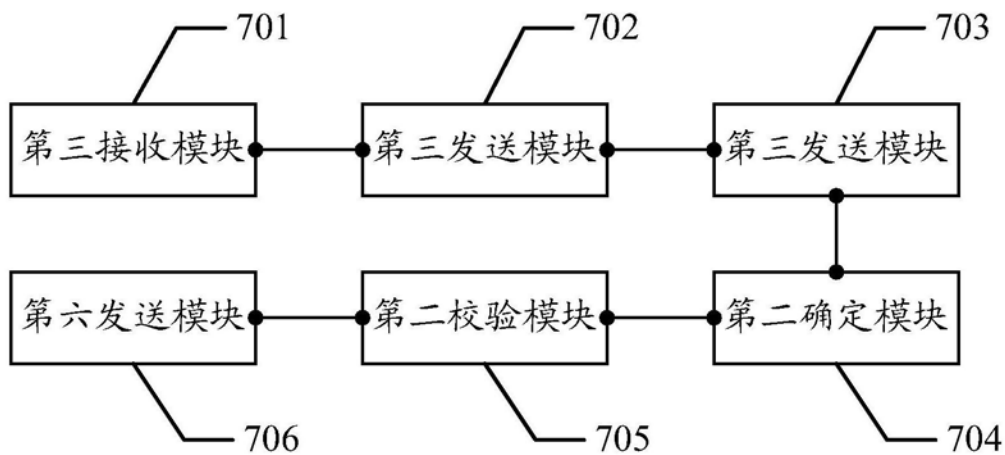


图7

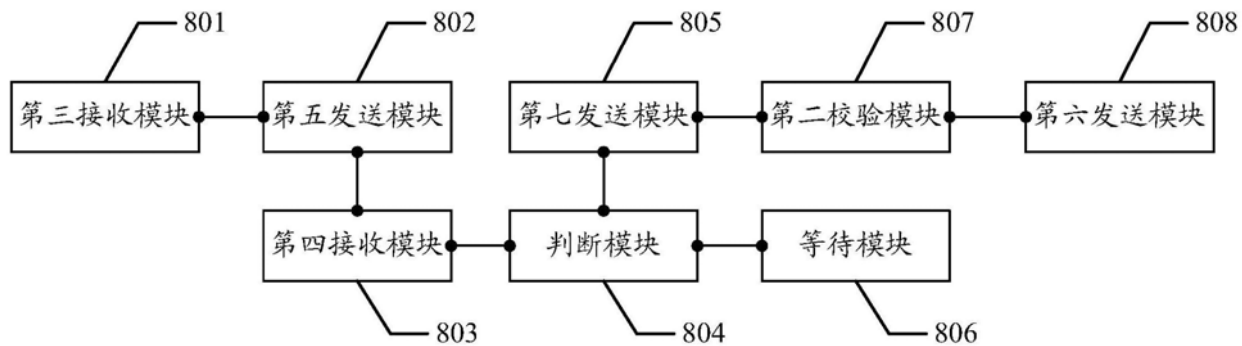


图8

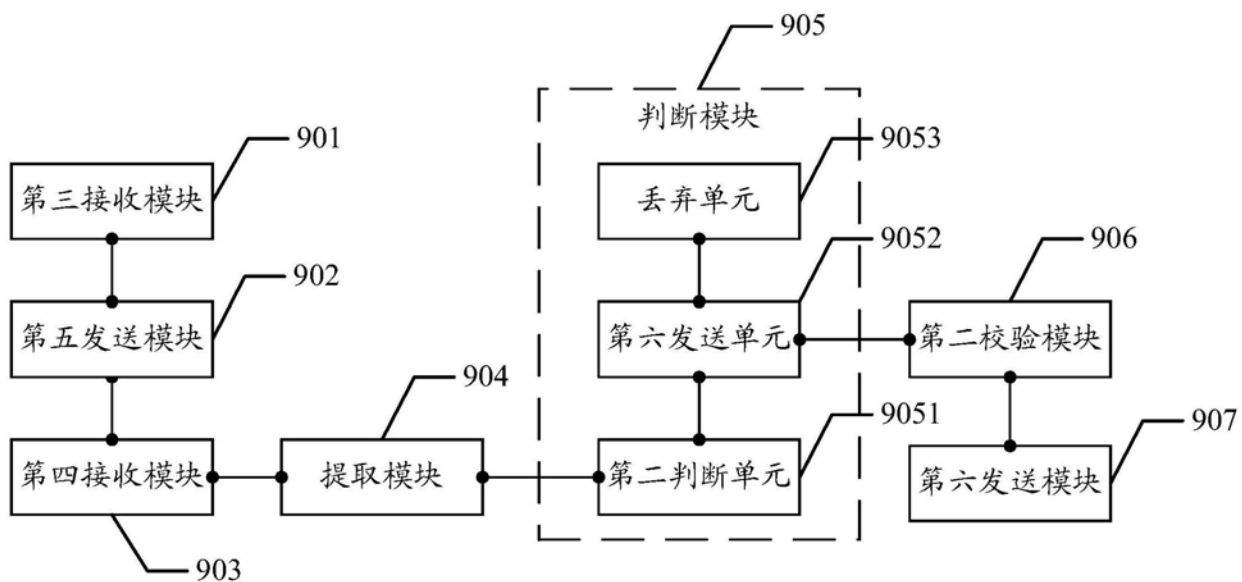


图9

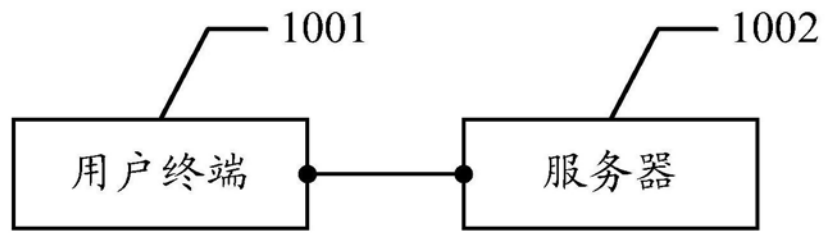


图10