

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2012 年 9 月 27 日 (27.09.2012)



(10) 国际公布号
WO 2012/126424 A2

- (51) 国际专利分类号:
H04L 12/56 (2006.01)
- (21) 国际申请号: PCT/CN2012/075715
- (22) 国际申请日: 2012 年 5 月 18 日 (18.05.2012)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (71) 申请人 (对除美国外的所有指定国): 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (72) 发明人: 及
- (75) 发明人/申请人 (仅对美国): 刘怡 (LIU, Yi) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (74) 代理人: 深圳市深佳知识产权代理事务所 (普通合伙) (SHENPAT INTELLECTUAL PROPERTY AGENCY); 中国广东省深圳市国贸大厦 15 楼西座 1521 室, Guangdong 518014 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 根据申请人的请求, 在条约第 21 条(2)(a)所规定的期限届满之前进行。
- 不包括国际检索报告, 在收到该报告后将重新公布(细则 48.2(g))。

(54) Title: METHOD AND DEVICE FOR FORWARDING DATA PACKET

(54) 发明名称: 一种数据包的转发方法和设备

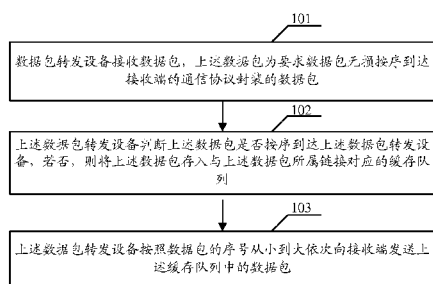


图 1 / Fig. 1

101 THE DATA PACKET FORWARDING DEVICE RECEIVES A DATA PACKET, WITH THE DATA PACKET BEING A DATA PACKET ENCAPSULATED BY A COMMUNICATION PROTOCOL WHICH REQUIRES UNDamaged AND ORDERED ARRIVAL OF DATA PACKETS AT THE RECEIVING END

102 THE DATA PACKET FORWARDING DEVICE JUDGES WHETHER OR NOT THE DATA PACKET ARRIVES AT THE DATA PACKET FORWARDING DEVICE IN ACCORDANCE WITH THE CORRECT ORDER; IF IT DOES NOT, THEN THE DATA PACKET WILL BE STORED IN A CACHE QUEUE CORRESPONDING TO THE LINK TO WHICH THE DATA PACKET BELONGS

103 THE DATA PACKET FORWARDING DEVICE SENDS THE DATA PACKETS IN THE CACHE QUEUE TO THE RECEIVING END ACCORDING TO THE SERIAL NUMBER OF THE DATA PACKET FROM SMALL TO BIG SUCCESSIVELY

(57) Abstract: Provided are a method and device for forwarding a data packet. The implementation of the method comprises: a data packet forwarding device receives a data packet, with the data packet being a data packet encapsulated by a communication protocol which requires undamaged and ordered arrival of data packets at the receiving end; the data packet forwarding device judges whether or not the data packet arrives at the data packet forwarding device in accordance with the correct order; if it does not, then the data packet will be stored in a cache queue corresponding to the link to which the data packet belongs; and the data packet forwarding device sends the data packets in the cache queue to the receiving end according to the serial number of the data packet from small to big successively. By way of adjusting the sequence of the data packets before the data packets are sent to the receiving end in the above method, the sending sequence of the data packets is closer to the original sequence thereof, which can lower the disordered sequence rate of the data packets and reduce the problem of bandwidth occupation by the repeated sending of the ACK request and the retransmission of the data packets by the receiving end, thus improving the throughput rate of the network.

[见续页]



(57) 摘要:

本发明实施例提供一种数据包的转发方法和设备。其中方法的实现包括:数据包转发设备接收数据包,所述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包;所述数据包转发设备判断所述数据包是否按序到达所述数据包转发设备,若否,则将所述数据包存入与所述数据包所属链接对应的缓存队列;所述数据包转发设备按照数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。以上方法通过在数据包发送至接收端之前,对数据包的顺序进行调整,使数据包的发送顺序更接近于数据包原有的顺序,可以降低数据包的乱序率,减少接收端重复发送 ACK 请求以及重传数据包占用带宽的问题,从而提升网络吞吐率。

一种数据包的转发方法和设备

技术领域

本发明涉及通信技术领域，特别涉及一种数据包的转发方法和设备。

5

背景技术

目前网络通信协议种类很多，其中一些网络通信协议提供了可靠的数据传输服务，可靠的数据传输服务要求数据包无损按序到达接收端，TCP（Transmission Control Protocol）是其中使用最广泛的一种，其提供了一种面向连接的、可靠的字节流服务。TCP 数据包的传输可靠性主要体现在两个方面，一是数据不能损坏，二是数据的顺序必须保持发端和收端一致。在发生上述两方面情况异常时，TCP 数据包的发送方会主动降低发送流量。

商用网络系统的组网是十分复杂的，比如：核心网到基站的传输通常采用公共网络传输，而公共网络由数量庞大的交换设备以网状结构组合在一起。由于组网复杂，容易造成数据包在传输路径上传输以后到达接收端时的时延不唯一，从而导致接收端接收到的数据包的乱序。

按照 TCP 规定，发送端按照 TCP 数据包的序号依次发送，例如：发送端向接收端依次发送序号为 1~9 的 TCP 数据包。由于上述 1~9 的各 TCP 数据包在传输路径上时延不相同，接收端收到的 TCP 数据包，可能存在乱序的问题。例如，收到的 TCP 数据包的序号依次为 1、2、3、5、6、7、4、8、9，其中序号 4 的数据包延时的时间较长，导致接收端收到的 TCP 数据包乱序。

依据 TCP 的规定，目前接收端设备不能区分 TCP 数据包乱序和丢失，因此目前接收端处理 TCP 数据包乱序的问题的方式是：接收端在发现接收到的 TCP 数据包的序号乱序时，向 TCP 数据的发送端发送确认（Acknowledgement，ACK）请求，用于请求 TCP 数据包的发送端发送当前正确接收的最后一个序号的下一个序号的 TCP 数据包。以前述收到的 TCP 数据包的序号依次为 123567489 为例，在接收到序号为 3 的 TCP 数据包时确定下一个 TCP 数据包的序号为 4 并发送 ACK 请求 4，但是实际收到的是 5，因此再发送一次 ACK

请求并继续判断下一个接收的 TCP 数据包是否为序号为 4 的 TCP 数据包, 发现仍然不是序号为 4 的 TCP 数据包再发一次 ACK 请求并继续判断下一个接收的 TCP 数据包是否为序号为 4 的 TCP 数据包, 依此类推, 上例中接收端接收到序号依次为 123567489 的数据包后会发送 3 次重复 ACK 请求, 会导致发送端重复发送序号为 4 的 TCP 数据包。

发明人在实现本发明实施例的过程中发现: 按照 TCP 的规定接收端接收到的 TCP 数据包出现乱序后会重复发送 ACK 请求, 导致 TCP 数据包重传次数过多, 占用带宽, 因此网络吞吐率较低。

10 发明内容

本发明实施例提供一种数据包的转发方法和设备, 用于降低数据包的乱序率, 从而提升网络吞吐率。

一种传数据包的转发方法, 包括:

数据包转发设备接收数据包, 所述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包;

所述数据包转发设备判断所述数据包是否按序到达所述数据包转发设备, 若否, 则将所述数据包存入与所述数据包所属链接对应的缓存队列;

所述数据包转发设备按照数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。

20 一种数据包转发设备, 包括:

接收单元, 用于接收数据包, 所述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包;

判断单元, 用于判断所述数据包是否按序到达所述数据包转发设备;

25 存储控制单元, 用于若所述判断单元判断结果为否, 则将所述数据包存入与所述数据包所属链接对应的缓存队列;

发送单元, 用于按照数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。

30 上述技术方案具有如下有益效果: 通过在数据包发送至接收端之前, 对数据包的顺序进行调整, 使数据包的发送顺序更接近于数据包原有的顺序, 可以降低数据包的乱序率, 减少接收端重复发送 ACK 请求以及重传数据包占用带

宽的问题，从而提升网络吞吐率。

附图说明

为了更清楚地说明本发明实施例的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

图 1 为本发明实施例方法流程示意图；

图 2A 为本发明实施例方案应用场景的网络结构示意图；

图 2B 为采用本发明实施例进行排序的实例示意图；

图 3 为本发明实施例另一方法流程示意图；

图 4 为本发明实施例转发设备结构示意图；

图 5 为本发明实施例另一转发设备结构示意图；

图 6 为本发明实施例另一转发设备结构示意图；

图 7 为本发明实施例另一转发设备结构示意图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

本发明实施例提供了一种传输控制协议数据包转发方法，如图 1 所示，包括：

101：数据包转发设备接收数据包，上述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包；

需要说明的是上述数据包可以为长期演进（Long Term Evolution，LTE）的网络中的 TCP 数据包，也可以是 TCP 以外的其他要求数据包按序到达接收端的通信协议封装的数据包，本发明实施例对此不予限定。

102：上述数据包转发设备判断上述数据包是否按序到达上述数据包转发

设备, 若否, 则将上述数据包存入与上述数据包所属链接对应的缓存队列;

上述判断上述数据包是否按序到达上述数据包转发设备, 一种实现方式可以包括: 获取上述数据包的链接识别信息以及上述数据包的序号; 通过上述链接识别信息确定上述数据包所属链接; 判断上述数据包的序号是否大于上述上述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号, 若是, 则确定上述数据包没有按序到达上述数据包转发设备。

上述链接识别信息可以是: 上述数据包的源互联网协议 IP 地址、目的 IP 地址、源端口号以及目的端口号。以 TCP 数据包为例, 这些信息可以通过读取 TCP 头中的相应字段来获取。需要说明的是, 获取链接识别信息的目的在于能够确定数据包所属的链接, 具体可选的信息的具体内容并不仅限于以上四个举例, 因此以上举例不应理解为对本发明实施例的限定。

可选地, 关于缓存队列, 可以预先建立, 也可以在确定需要对数据包进行重新排序并发送时才建立该缓存队列, 还可以在确定上述数据包没有按序到达上述数据包转发设备之后, 建立与上述数据包所属链路对应的缓存队列。采用在确定上述数据包没有按序到达上述数据包转发设备之后建立缓存队列的方式, 可以防止不必要的缓存队列占用存储资源。

进一步地, 在将上述数据包存入其所属链接的缓存队列之前还可以包括: 若上述数据包的序号小于或等于上述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号, 则确定上述数据包所属链路对应的缓存队列是否为空, 若为空, 则向接收端发送上述数据包, 若不为空, 则将上述数据包存入与上述数据包所属链接对应的缓存队列。本实施例是基于在确定上述数据包没有按序到达上述数据包转发设备之后, 建立缓存队列的基础上提出的, 对于比上述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号小的数据包, 可能并不是因时延导致到达的时间滞后, 可以将这类数据包直接发送给接收端。

优选地, 上述 102 中将上述数据包存入与上述数据包所属链接对应的缓存队列包括: 将上述数据包存入与上述数据包所属链接对应的缓存队列, 且缓存队列按照序号从小到大排序。在本步骤中缓存队列中存储的数据包的序号按照从小到大排序可以与协议原有的数据包的发送方案兼容。在本步骤进行排序作为一个优选方案, 不应理解为对本发明实施例的限定。

103: 上述数据包转发设备按照数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包。

可选地, 上述 103 中按照数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包包括: 在定时器计时达到预定值时或者上述缓存队列中的数据包达到预定个数时, 按照上述缓存队列中的数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包。需要说明的是, 以上关于定时器以及数据包达到预定个数的举例, 均属于触发转发设备发送数据包的条件, 这个触发条件可以有很多, 并不仅限于以上两种, 本发明实施例对此不予限定。具体的, 上述预定值可以设置为 2 毫秒 (ms)。

采用本发明实施例方案, 通过在数据包发送至接收端之前, 对数据包的顺序进行调整, 使数据包的发送顺序更接近于数据包原有的顺序, 可以降低数据包的乱序率, 减少接收端重复发送 ACK 请求以及重传数据包占用带宽的问题, 从而提升网络吞吐率。

另外, 按照 TCP 的协议规定, 收到同一个 TCP 数据包的三次 ACK 请求则会被 TCP 数据包的发送端确认为网络出现了可靠性异常的状况, 在发送端确认可靠性异常以后会主动将数据包的发送速率降低一半, 因此对于 TCP 数据包传输而言, TCP 数据包乱序会导致发送端主动降低发送速率, 进而降低网络吞吐率。而采用本发明实施例, 可以降低数据包的乱序率, 从而减少发送端因收到 ACK 请求主动降低发送速率的状况, 因此可以提升网络吞吐率。

以转发 TCP 数据包为例, 以上实施例的执行主体可以是 TCP 数据包的发送端到接收端之间的任意转发设备, 比如基站。如图 2A 所示, 在下行方向上, 上述 TCP 数据包的发送端在核心网, 核心网发送 TCP 数据包到达基站, 基站将 TCP 数据包发送给接收端。在图 2A 中所示的接收端为手机, 转发设备为基站, 可以理解的是接收端可以有很多种类, 可能是终端设备也可能是其他网络侧设备, 转发设备也可以有很多种类, 本发明实施例对此不予限定。

比如, 可以在长期演进 (Long Term Evolution, LTE) 系统的网元如演进型基站 (eNodeB) 中添加一个 TCP 处理模块, 以执行本发明实施例提供的方法, 该 TCP 处理模块可以对下行的数据包进行处理。该 TCP 处理模块检测每个下行的 TCP 数据包是否按序到达 eNodeB, 如果检测到前后两个 TCP 数据包的序号不连续, 则可以确定 TCP 数据包没有按序到达 eNodeB, 那么可以在

eNodeB 上缓存乱序的数据包，并缓存随后一段时间内到达 eNodeB 的数据包并对其进行排序。

举例说明：在图 2B 中上方为发生乱序时数据包到达 eNodeB 的情况，下方是经过本发明实施例方法处理后数据包到达接收端的情况，此处的接收端是 TCP 数据包的目的端，在图 2A 中举例为手机。在图 2B 中，time 为 TCP 数据包的到达时间，tcp_seq 为 TCP 数据包的序号，info 为其它信息（图 2 中显示为文件传送协议数据的长度）。从图 2 上部可以看到序号为 3284316717 的数据包到达 eNodeB 时本应该出现在序号为 3284325257 和 3284318177 的数据包之间，由于乱序而出现在 3284321097 后面，经过本方案的处理后，该数据包回到了正确的位置。因此，采用本发明实施例方法可以降低乱序率。

发明人对本发明是实施例的效果进行了实测，测试场景如图 2A 所示：LTE 网络，测试对象为核心网传输的 TCP 数据包达到基站，基站作为数据包转发设备执行本发明实施例方案，并将 TCP 数据包转发给手机。该场景下本发明实施例方案能够提高 LTE 网络的平均吞吐率。具体的增益效果与有线传输侧（核心网到基站）的乱序率、乱序范围、乱序时延以及传输往返时延（Round-Trip Time, RTT）等有关。乱序率越大、RTT 越大则增益越明显。需要说明的是：要求数据包无损按序到达接收端的通信协议封装的数据包可以应用于很多种类的网络，LTE 网络的举例不应理解为对本发明实施例的限定。发明人对以上实施例的效果进行实测，测试的场景为：手机所在小区带宽 20MHz，LTE 网络的无乱序峰值速率 148Mbps，制造乱序率为 1/10000、乱序平均分布、乱序范围为 2~5 的数据流，手机从核心网采用文件传输协议（File Transfer Protocol, FTP）单线程下载 5 次 1G 大小的文件。测试结果为：统计得到不采用本发明实施例方法的 LTE 网络平均吞吐率为 40.367Mbps，采用本发明实施例方法后 LTE 网络平均吞吐率最小 126.912Mbps，最大为 137.568Mbps，相比于不采用本发明实施例方法增益为 214.3%~241.3%。因此实验证明采用本发明实施例方案可以有效提升 LTE 网络吞吐率。

以下实施例将给出一个具体的举例来对本发明实施例的实现进行说明，以 TCP 数据包的转发为例进行说明，具体如图 3 所示，包括如下步骤：

301：数据包转发设备接收数据包，并检查其是否为 TCP 数据包，若否，进入 302，若是 TCP 数据包，则进入 303；

302: 直接透传上述 TCP 数据包, 然后返回 301;

303: 获取 TCP 数据包的链接识别信息并依此确定其所属的链接 (即 TCP 链接), 检查其所属的 TCP 链接是否已存在排序信息;

上述排序信息包括的信息有: 源、目的 IP 地址, 源、目的端口号, 还可
5 包括: 定时器计时的值、当前已发送的最大序号 CurrSeq。如果存在排序信息, 则进入 305, 如果不存在排序信息, 则进入 304;

304: 为上述数据包建立排序信息, 然后进入 305;

305: 将 TCP 数据包进行排序预处理;

预处理可以包括: 将 CurrSeq 初始化为收到的第一个 TCP 数据包的序号;
10 需要说明的是, 如果 CurrSeq 本身存在初值 (即曾经被初始化过) 则不用执行对 CurrSeq 的初始化, 直接进入 306;

需要说明的是: 上述 301~305 采用的是接收到第一个 TCP 数据包就直接建立了排序信息; 若排序信息在确定 TCP 数据包出现了乱序后才建立, 那么本步骤中的 CurrSeq 初始化为建立排序信息当前的 TCP 数据包的序号。可以
15 理解的是, 如果 TCP 数据包没有出现乱序的状况, TCP 数据包会被直接发送。

306: 计算按序发送的下一个 TCP 数据包的序号 NxtSn; 解析得到上述 TCP 数据包的序号, 将解析得到的序号与 NxtSn 做比较: 若上述 TCP 数据包的序号小于等于 NxtSn, 进入 307; 若上述 TCP 数据包的序号大于 NxtSn, 进入 308;

307: 检查与上述 TCP 数据包所属链接对应的缓存队列是否为空, 若缓存
20 队列为空, 进入 302, 并将 CurrSeq 更新为当前已转发的 TCP 数据的最大序号, 计算并更新 NxtSn 为新的 CurrSeq 的下一个 TCP 数据包的序号; 若缓存队列不为空, 进入 308;

在上述 307 中检查缓存队列为空的情况有两种: 一种是不存在与上述 TCP 数据包所属链接对应的缓存队列, 另一种是存在与上述 TCP 数据包所属链接
25 对应的缓存队列, 并且该缓存队列中没有存数据包。上述“不存在与上述 TCP 数据包所属链接对应的缓存队列”的原因如下: 若在步骤 306 之前采用在确定 TCP 数据包出现了乱序后才建立的方案, 那么在没有出现乱序的情况下, 上述 TCP 数据包所属链接对应的缓存队列是不存在的; 也即: 上述 307 中在检查缓存队列是否为空时, 并不存在与上述 TCP 数据包所属链接对应的缓存队列。

30 308: 检查定时器是否启动, 若定时器未启动, 则启动定时器, 并将上述

TCP 数据包存入缓存队列;

本步骤中, 若定时器已启动, 则直接将该 TCP 数据包存入缓存队列; 需要说明的是, 在本步骤中优选地进行按照 TCP 数据包的序号从小到大依次排序缓存队列中的 TCP 数据包。

- 5 309: 定时器超时后, 无论是否收到导致乱序的数据包, 都将缓存队列中的 TCP 数据包全部按序 (即: 按 TCP 数据包的序号从小到大) 依次发送给接收端, 并将 CurrSeq 更新为当前已转发的 TCP 数据的最大序号, 计算并更新 NxtSn 为新的 CurrSeq 的下一个 TCP 数据包的序号, 然后进入 301。在本步骤中, 将缓存队列中的 TCP 数据包发送给接收端后, 可以将已经发送的 TCP 数据包从缓存队列中删除。
- 10

需要说明的是: 设置定时器时长的原则: 定时器时长越长, 能纠正的乱序比例越高, 排序效果越好, 但会增加数据的 RTT 时延, 对于丢包场景, 可能会降低峰值速率, 还可能增加超文本传送协议 (hypertext transport protocol, http) 业务的时延; 定时器时长越短, 能纠正的乱序比例越低, 排序效果越差, 但对数据的 RTT 时延影响小, 在丢包场景对峰值速率影响小并且对 http 业务时延影响小。定时器的默认时间根据场景设置不同的定时器取值来应对不同的乱序时延。具体的时间本发明实施例不予限定。在本步骤中, 定时器可以设置为 2ms。

15

另外需要说明的是, 在上面图 3 步骤的任意一个步骤执行过程中, 若确定了链接的数据业务完毕, 那么该流程将结束。另外, 可以理解的是在缓存队列中的 TCP 数据包全部发送后缓存队列可以清空。

20

本发明实施例给出了以上图 3 对应实施例执行的一个举例, 假设如下:

TCP 数据包最大长度: 100;

TCP 数据包序号为: 3012、3112、3212、3312、3362、3462、3562、3662、3762、3862、3962;

25

数据包转发设备接收到 TCP 数据包的顺序为: 3012、3112、3212、3312、3362、3562、3662、3762、3862、3462、3962;

观察可以发现: 导致乱序的数据包为序号为 3462 的数据包。

那么执行图 3 的方案, 具体为:

30 数据包 3012 在步骤 301 到达数据包转发设备并确定为 TCP 数据包, 然后

在步骤 304 建立排序信息, CurrSeq 在步骤 305 被初始化为 3012。在步骤 306 会计算得到 NxtSn=当前 CurrSeq + 100。由于当前缓存队列为空并且收到的是 3112, 那么在执行步骤 306 中的比较时会确定 3112 等于 NxtSn, 因此可以确定不乱序, 该 TCP 数据包会被直接发送给接收端, 更新 CurrSeq 和 NxtSn。同理, 对于数据包 3212, CurrSeq 更新后为 3312, 并计算 NxtSn 得到的是 3312。

由于有时会出现服务器不按最大包长来发包的情况, 例如 3312 这个包, 下一个包序号是 3362, 由于 3312 这个数据包长度只有 50 字节, 收到 3362 时, 此时缓存队列仍然为空, 此时 $\text{CurrSeq} < 3362 < 3412$, 不会被认为是乱序, 3362 依然会被转发, 并更新 CurrSeq 为 3362, 计算 NxtSn 得到 3462。

在收到 3562 数据包时, 由于其大于 NxtSn, 会被认为乱序, 该数据包将会被存入缓存队列, 并且启动排序定时器。对于后面序号为: 3662、3762、3862, 3462 的数据包, 如果在定时器超时之前到达, 将会被存入缓存队列并在缓存队列中被重新排序, 定时器超时后按序发出, 并更新 CurrSeq 为 3862。假如收到序号为 3762 的数据包后, 未收到序号为 3462 的数据包定时器就超时了, 则将缓存队列中的 3562、3662、3762 按序发出, 并更新 CurrSeq 为 3762 (当前发送的数据包的最大序号); 后续收到 3862, 判断不乱序, 直接转发, 更新 CurrSeq; 再收到 3462, 小于当前 CurrSeq, 直接转发。收到序号为 3962 的数据包时, 会确定其等于 NxtSn, 直接转发, 并更新 CurrSeq 和 NxtSn。

如果排序信息在确定 TCP 数据包出现了乱序后才建立, 那么前面的序号为 3012、3112、3212、3312、3362 这些数据包没有出现乱序, 因此会被直接转发, 在序号为 3362 的数据包发出后, CurrSeq 为 3362, NxtSn 为 3462。在序号为 3562 的数据包到达的时候由于比 NxtSn 大, 所以确定乱序了, 此时会建立排序信息; 序号为 3562、3662、3762、3862 的数据包到达时, 序号比 NxtSn 大, 这些数据包会被存入缓存队列; 序号为 3462 的数据包的序号等于 NxtSn, 由于此时缓存队列不为空, 序号为 3462 的数据包仍然会被缓存, 并被排序。如果收到 3462 后收到 3962 之前, 定时器超时那么缓存队列中的数据包将会全部被按序转发, 并更新 CurrSeq 为 3862, NxtSn 计算得到的值是 3962。

本发明实施例还提供了一种数据包转发设备, 如图 4 所示, 包括:

接收单元 401, 用于接收数据包, 上述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包;

需要说明的是上述数据包可以为LTE网络中的TCP数据包,也可以是TCP以外的其他要求数据包按序到达接收端的通信协议封装的数据包,本发明实施例对此不予限定。

判断单元402,用于判断上述数据包是否按序到达上述数据包转发设备;

5 存储控制单元403,用于若上述判断单元402判断结果为否,则将上述数据包存入与上述数据包所属链接对应的缓存队列;

发送单元404,用于按照数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包。

10 可选地,上述发送单元404,用于按照数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包,包括:用于在定时器计时达到预定值时或者上述缓存队列中的数据包达到预定个数时,按照上述缓存队列中的数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包。需要说明的是,以上关于定时器以及数据包达到预定个数的的举例,均属于触发转发设备发送数据包的条件,这个触发条件可以有很多,并不仅限于以上两种,本发明实施例对此不予限定。具体的,上述预定值可以设置为2毫秒(ms)。

采用本发明实施例方案,通过在数据包发送至接收端之前,对数据包的顺序进行调整,使数据包的发送顺序更接近于数据包原有的顺序,可以降低数据包的乱序率,减少接收端重复发送ACK请求以及重传数据包占用带宽的问题,从而提升网络吞吐率。

20 另外,按照TCP的协议规定,收到同一个TCP数据包的三次ACK请求则会被TCP数据包的发送端确认为网络出现了可靠性异常的状况,在发送端确认可靠性异常以后会主动将数据包的发送速率降低一半,因此对于TCP数据包传输而言,TCP数据包乱序会导致发送端主动降低发送速率,进而降低网络吞吐率。而采用本发明实施例,可以降低数据包的乱序率,从而减少发送端
25 因收到ACK请求主动降低发送速率的状况,因此可以提升网络吞吐率。

可选的,如图5所示,上述判断单元402,包括:获取子单元501、链接确定子单元502以及判断子单元503;

上述获取子单元501,用于获取上述数据包的链接识别信息以及上述数据包的序号;

30 上述链接确定子单元502,用于通过上述链接识别信息确定上述数据包所

属链接;

上述判断子单元 503, 用于判断上述数据包的序号是否大于上述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号, 若是, 则确定上述数据包没有按序到达上述数据包转发设备。

- 5 上述链接识别信息可以是: 上述数据包的源互联网协议 IP 地址、目的 IP 地址、源端口号以及目的端口号。以 TCP 数据包为例, 这些信息可以通过读取 TCP 头中的相应字段来获取。需要说明的是, 获取链接识别信息的目的在于能够确定数据包所属的链接, 具体可选的信息的具体内容并不仅限于以上四个举例, 因此以上举例不应理解为对本发明实施例的限定。

- 10 进一步可选的, 如图 6 所示上述设备, 还包括:

队列控制单元 601, 用于在上述判断子单元 503 确定上述数据包没有按序到达上述数据包转发设备之后, 建立与上述数据包所属链路对应的缓存队列。

采用在确定上述数据包没有按序到达上述数据包转发设备之后, 建立缓存队列可以防止不必要的缓存队列占用存储资源。

- 15 进一步地, 如图 7 所示上述设备, 还包括:

缓存信息获取单元 701, 用于若上述数据包的序号小于或等于上述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号, 则确定上述数据包所属链路对应的缓存队列是否为空;

- 20 上述发送单元 404, 还用于若上述缓存信息获取单元 701 确定上述数据包所属链路对应的缓存队列为空, 则向接收端发送上述数据包;

上述存储控制单元 403, 还用于若缓存信息获取单元 701 确定上述数据包所属链路对应的缓存队列不为空, 则将上述数据包存入与上述数据包所属链接对应的缓存队列。

- 25 本实施例是基于在确定上述数据包没有按序到达上述数据包转发设备之后, 建立缓存队列的基础上提出的, 对于比上述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号小的数据包, 可能并不是因时延导致到达的时间滞后, 可以将这类数据包直接发送给接收端。

- 30 更具体地, 上述发送单元 404, 具体用于在定时器计时达到预定值时或者上述缓存队列中的数据包达到预定个数时, 按照上述缓存队列中的数据包的序号从小到大依次向接收端发送上述缓存队列中的数据包。

需要说明的是，以上关于定时器以及数据包达到预定个数的举例，均属于触发转发设备发送数据包的条件，这个触发条件可以有很多，并不仅限于以上两种，本发明实施例对此不予限定。

更具体地，上述存储控制单元 403，具体用于将上述数据包存入与上述数据包所属链接对应的缓存队列，且缓存队列按照序号从小到大排序。

需要说明的是，在发送单元 404 发送数据包时可以按照序号的从小到大发送，以实现本发明实施例的技术目的，因此在实施例 10 中，在发送之前就进行排序的方案可以与协议原有的数据包的发送方案兼容。在本实施例中采用存储控制单元 403 对缓存队列中的数据包进行排序作为一个优选方案，不应理解为对本发明实施例的限定。

以转发 TCP 数据包为例，以上数据包转发设备可以是 TCP 数据包的发送端到接收端之间的任意转发设备，比如基站。如图 2A 所示，在下行方向上，上述 TCP 数据包的发送端在核心网，核心网发送 TCP 数据包到达基站，基站将 TCP 数据包发送给接收端。在图 2A 中所示的接收端为手机，转发设备为基站，可以理解的是接收端可以有很多种类，可能是终端设备也可能是其他网络侧设备，转发设备也可以有很多种类，本发明实施例对此不予限定。

比如，可以在 LTE 系统的网元如 eNodeB 中添加一个 TCP 处理模块，来实现上述数据包转发设备中各组成单元的功能，该 TCP 处理模块可以对下行的数据包进行处理。该 TCP 处理模块检测每个下行的 TCP 数据包是否按序到达 eNodeB，如果检测到前后两个 TCP 数据包的序号不连续，则可以确定 TCP 数据包没有按序到达 eNodeB，那么可以在 eNodeB 上缓存乱序的数据包，并缓存随后一段时间内到达 eNodeB 的数据包并对其进行排序。

举例说明：在图 2B 中上方为发生乱序时数据包到达 eNodeB 的情况，下方是经过本发明实施例装置处理后数据包到达接收端的情况，此处的接收端是 TCP 数据包的目的端，在图 2A 中举例为手机。在图 2B 中，time 为 TCP 数据包的到达时间，tcp_seq 为 TCP 数据包的序号，info 为其它信息（图 2 中显示为文件传送协议数据的长度）。从图 2 上部可以看到序号为 3284316717 的数据包到达 eNodeB 时本应该出现在序号为 3284325257 和 3284318177 的数据包之间，由于乱序而出现在 3284321097 后面，经过本方案的处理后，该数据包回到了正确的位置。因此，采用本发明实施例设备可以降低乱序率。

值得注意的是，上述设备的实施例中，所包括的各个单元只是按照功能逻辑进行划分的，但并不局限于上述的划分，只要能够实现相应的功能即可；另外，各功能单元的具体名称也只是为了便于相互区分，并不用于限制本发明的保护范围。各功能单元的功能可以由接收机、存储器、处理器和发射机等元件实现。

本领域普通技术人员可以理解实现上述实施例方法中的全部或部分步骤是可以通过程序来指令相关的硬件完成，上述的程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，磁盘或光盘等。

本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处，综上，本说明书内容不应理解为对本发明的限制。

权 利 要 求

1、一种数据包的转发方法，其特征在于，包括：

数据包转发设备接收数据包，所述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包；

5 所述数据包转发设备判断所述数据包是否按序到达所述数据包转发设备，若否，则将所述数据包存入与所述数据包所属链接对应的缓存队列；

所述数据包转发设备按照数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。

2、根据权利要求 1 所述方法，其特征在于，所述判断所述数据包是否按序到达所述数据包转发设备，包括：

获取所述数据包的链接识别信息以及所述数据包的序号；

通过所述链接识别信息确定所述数据包所属链接；

15 判断所述数据包的序号是否大于所述所述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号，若是，则确定所述数据包没有按序到达所述数据包转发设备。

3、根据权利要求 2 所述方法，其特征在于，在确定所述数据包没有按序到达所述数据包转发设备之后，建立与所述数据包所属链路对应的缓存队列。

4、根据权利要求 3 所述方法，其特征在于，还包括：

20 若所述数据包的序号小于或等于所述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号，则确定所述数据包所属链路对应的缓存队列是否为空，若为空，则向接收端发送所述数据包，若不为空，则将所述数据包存入与所述数据包所属链接对应的缓存队列。

5、根据权利要求 1 至 4 任意一项所述方法，其特征在于，所述按照数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包包括：

25 在定时器计时达到预定值时或者所述缓存队列中的数据包达到预定个数时，按照所述缓存队列中的数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。

6、根据权利要求 1 至 4 任意一项所述方法，其特征在于，所述将所述数据包存入与所述数据包所属链接对应的缓存队列包括：

将所述数据包存入与所述数据包所属链接对应的缓存队列，且缓存队列按照序号从小到大排序。

7、根据权利要求 1 至 4 任意一项所述方法，其特征在于，所述数据包为长期演进 LTE 的网络中的传输控制协议 TCP 数据包。

5 8、一种数据包转发设备，其特征在于，包括：

接收单元，用于接收数据包，所述数据包为要求数据包无损按序到达接收端的通信协议封装的数据包；

判断单元，用于判断所述数据包是否按序到达所述数据包转发设备；

10 存储控制单元，用于若所述判断单元判断结果为否，则将所述数据包存入与所述数据包所属链接对应的缓存队列；

发送单元，用于按照数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。

9、根据权利要求 8 所述设备，其特征在于，

所述判断单元，包括：获取子单元、链接确定子单元以及判断子单元；

15 所述获取子单元，用于获取所述数据包的链接识别信息以及所述数据包的序号；

所述链接确定子单元，用于通过所述链接识别信息确定所述数据包所属链接；

20 所述判断子单元，用于判断所述数据包的序号是否大于所述所述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号，若是，则确定所述数据包没有按序到达所述数据包转发设备。

10、根据权利要求 9 所述设备，其特征在于，还包括：

队列控制单元，用于在所述判断子单元确定所述数据包没有按序到达所述数据包转发设备之后，建立与所述数据包所属链路对应的缓存队列。

25 11、根据权利要求 10 所述设备，其特征在于，还包括：

缓存信息获取单元，用于若所述数据包的序号小于或等于所述数据包所属链接已经发送的数据包中序号最大的序号的下一个序号，则确定所述数据包所属链路对应的缓存队列是否为空；

30 所述发送单元，还用于若所述缓存信息获取单元确定所述数据包所属链路对应的缓存队列为空，则向接收端发送所述数据包；

所述存储控制单元,还用于若缓存信息获取单元确定所述数据包所属链路对应的缓存队列不为空,则将所述数据包存入与所述数据包所属链接对应的缓存队列。

12、根据权利要求 8 至 11 所述设备,其特征在于,

5 所述发送单元,具体用于在定时器计时达到预定值时或者所述缓存队列中的数据包达到预定个数时,按照所述缓存队列中的数据包的序号从小到大依次向接收端发送所述缓存队列中的数据包。

13、根据权利要求 8 至 11 所述设备,其特征在于,

10 所述存储控制单元,具体用于将所述数据包存入与所述数据包所属链接对应的缓存队列,且缓存队列按照序号从小到大排序。

— 1/3 —

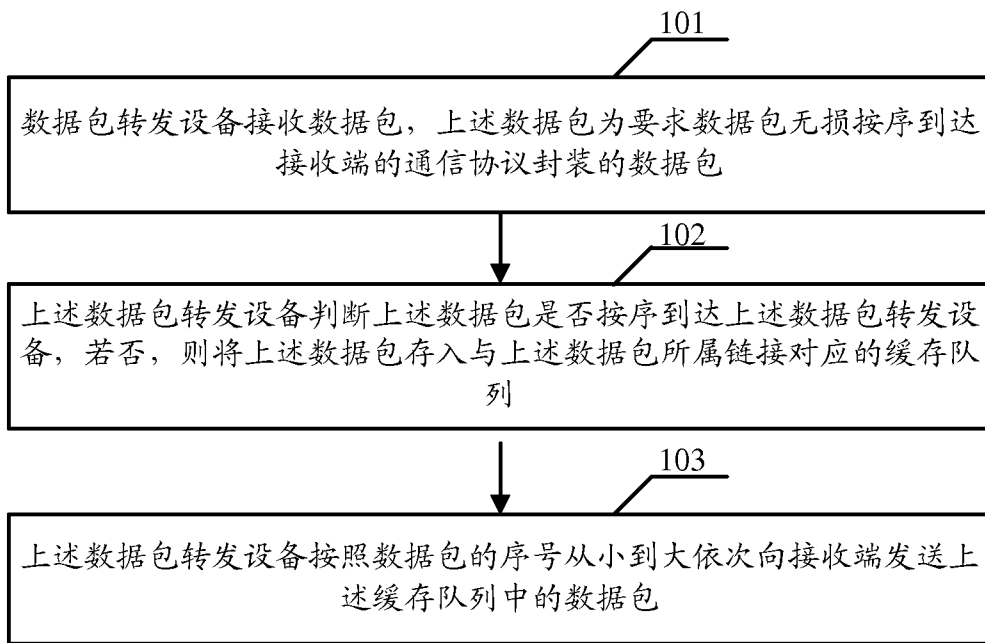


图 1

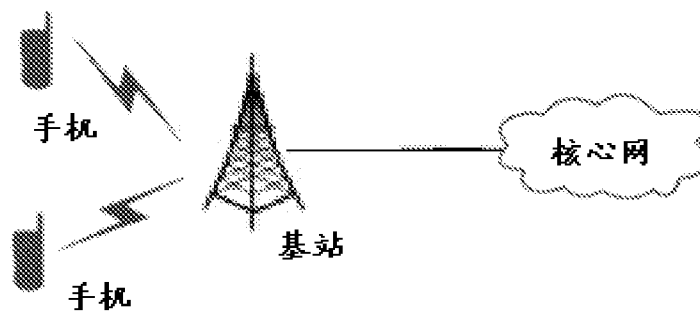


图 2A

tcp.seq	tcp.len	info
3284315257	1460	FTP Data: 1460bytes
3284318177	1460	FTP Data: 1460bytes
3284319637	1460	FTP Data: 1460bytes
3284321097	1460	FTP Data: 1460bytes
3284316717	1460	FTP Data: 1460bytes

time	tcp.seq	tcp.len	info
13.687616	3284315257	1460	FTP Data: 1460bytes
13.687637	3284316717	1460	FTP Data: 1460bytes
13.687653	3284318177	1460	FTP Data: 1460bytes
13.688010	3284319637	1460	FTP Data: 1460bytes
13.688031	3284321097	1460	FTP Data: 1460bytes
13.688047	3284322557	1460	FTP Data: 1460bytes
13.688384	3284324017	1460	FTP Data: 1460bytes

图 2B

— 2/3 —

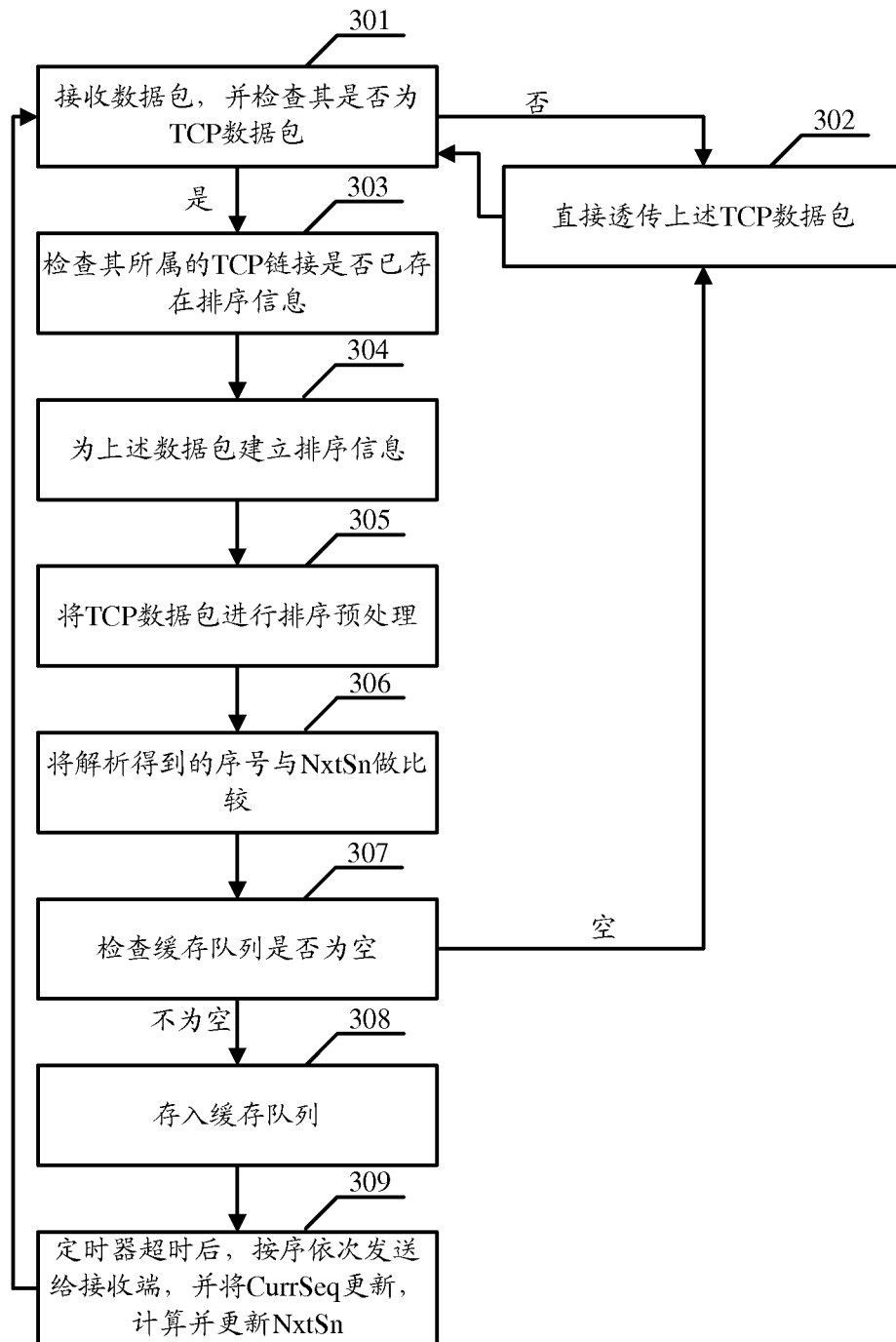


图 3

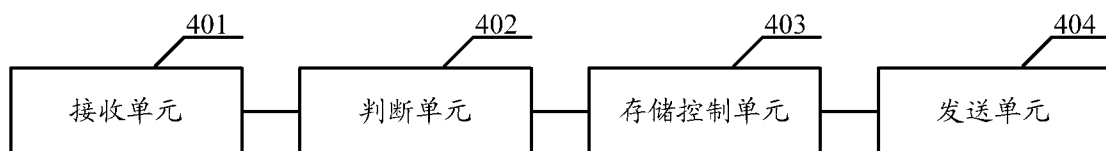


图 4

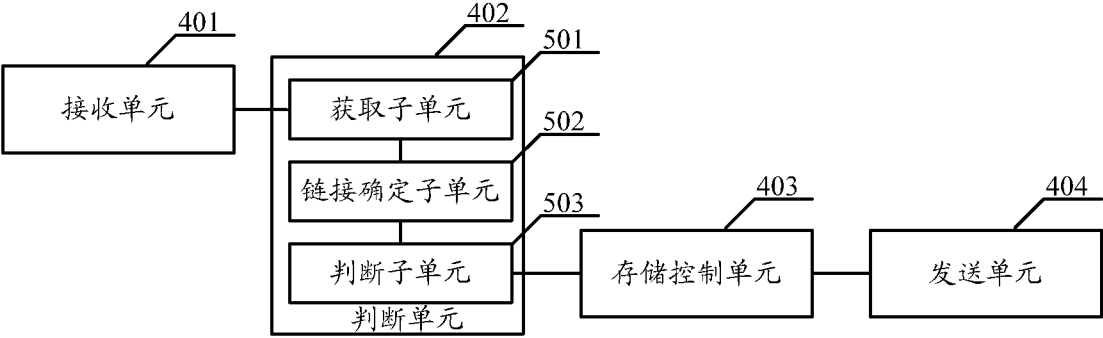


图 5

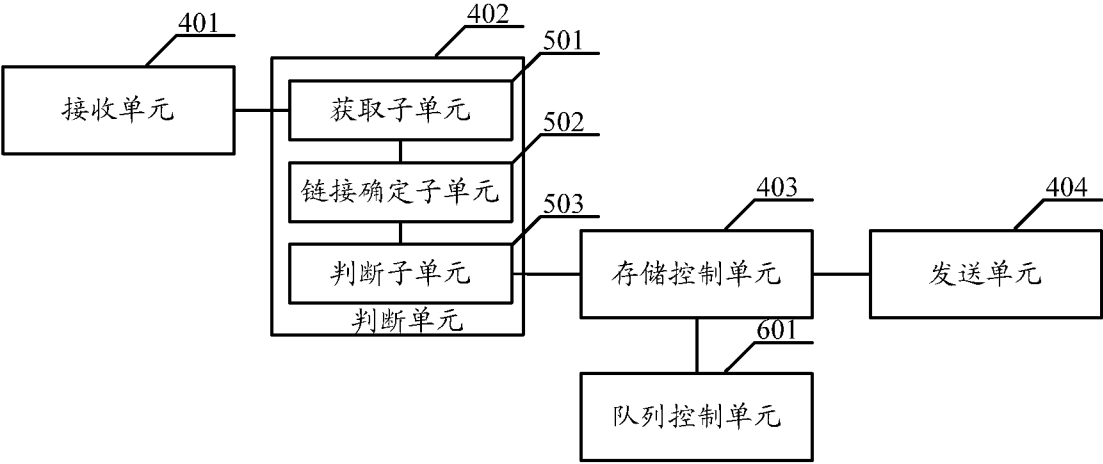


图 6

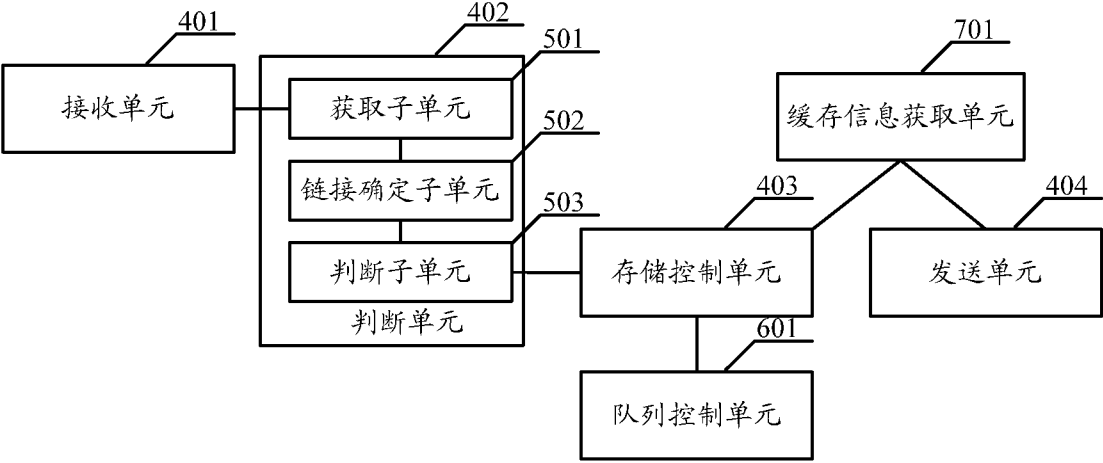


图 7