



[12] 发明专利说明书

专利号 ZL 200610066367.9

[45] 授权公告日 2010年3月24日

[11] 授权公告号 CN 100596049C

[22] 申请日 2006.3.30

[21] 申请号 200610066367.9

[73] 专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼岛

[72] 发明人 钱志龙 程立 李磊

[56] 参考文献

CN1467952A 2004.1.14

WO2005/109804A2 2005.11.17

CN1194530A 1998.9.30

审查员 张玉

[74] 专利代理机构 北京集佳知识产权代理有限公司

代理人 逯长明

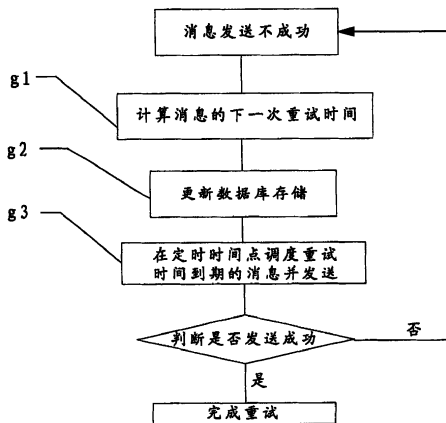
权利要求书 1 页 说明书 13 页 附图 4 页

[54] 发明名称

一种消息重发方法和系统

[57] 摘要

本发明公开一种消息重发方法和系统，能够用尽可能少的重试次数、尽可能及时地将消息通知发送到消息接收方。所述方法包括：对需要重试的消息设置重试周期，以当前时间加上对应的重试周期作为重试时间；所述重试周期随着重试次数的增加而增大；发送重试时间到期的消息。当所述重试次数或所述重试周期达到预置的值时，放弃自动重发。本发明还公开一种可靠的系统间消息通知方法和系统，能够保证消息通知的可靠到达。所述方法和系统支持不同系统间的多协议传输，接收方不需要实现复杂的交互协议就能够可靠接收通知消息，适合广泛应用于互联网。此外，还支持多业务处理，可作为一个通用的业务应用平台。同时，还可以进行多业务和多协议的灵活扩展。



1、一种消息重发系统，其特征在于，包括：

数据库，用于保存待发送的新消息和等待重试的消息；

业务插件，用于根据是否有返回结果，以及返回结果的格式和内容是否有效决定本次发送是否需要重试；

通知执行器，用于发送新消息或重试的消息，若不需要重试，则从所述数据库中删除本条消息；否则对应所述消息设置重试周期，以当前时间加上对应的重试周期作为重试时间，并更新所述数据库中的对应消息；所述重试周期随着重试次数的增加而增大；

通知恢复器，用于检查等待重试的消息，如果存在重试时间到期的消息，则通知所述通知执行器。

2、根据权利要求1所述的一种消息重发系统，其特征在于：所述的通知恢复器定时运行。

3、一种系统间消息通知方法，其特征在于，包括以下步骤：

将通知消息存储；

发送存储的新通知消息或重试的通知消息；

业务插件根据是否有返回结果，以及返回结果的格式和内容是否有效决定本次发送是否需要重试；

若不需要重试，则删除存储的本条通知消息；否则对该通知消息设置重试周期，以当前时间加上对应的重试周期作为重试时间，并更新存储；所述重试周期随着重试次数的增加而增大；

检查等待重试的通知消息，如果存在重试时间到期的通知消息，则执行发送步骤。

4、根据权利要求3所述的一种系统间消息通知方法，其特征在于，还包括：同时发送新通知消息以及重试的通知消息。

5、根据权利要求3所述的一种系统间消息通知方法，其特征在于：所述检查为定时检查。

一种消息重发方法和系统

技术领域

本发明涉及数据传输领域，特别涉及一种消息重发方法和系统。

背景技术

在数据传输领域，以消息通知作为交互模式的应用系统越来越多。在消息传输时，由于双方系统或传输链路的故障，导致消息在传输过程中丢失而不能及时、可靠的送达。针对这一问题，形成了各种不同的消息重发机制，用于重新发送未成功送达的消息。采用重发机制，在第一次发送不成功的情况下，能够进行多次的发送尝试，在很大程度上解决了消息发送不能送达的问题。但对于中长期的通信故障，由于多次的尝试重发，即浪费发送时间又占用系统资源，造成了发送效率的低下。

在网络协议设计中，针对数据传输失败进行回退重传的算法很多。其中使用最广的一个算法是二进制指数回退（Binary Exponential Backoff, BEB）算法。下面以以太网 802.3 协议为例说明该算法的详细过程。

802.3 协议使用 CSMA/CD 算法解决在共享信道上的数据帧冲突问题。该算法持续侦听信道，直到信道空闲；一旦信道空闲，立即发送数据；若数据冲突，立即停止发送；然后等待一个重试周期后再次尝试发送。计算重试周期的方法是二进制指数后退算法。它首先将时间分片，每个时间片的长度是 t （51.2 微秒）。则在第 i 次冲突时，重试周期 T 是在 0 到 $(2^i - 1) * t$ 之间的一个随机的长度为 t 的整数倍的时间。

二进制指数回退算法使得当数据传输发生故障时，重试时间间隔呈指数迅速增长。这种针对随机故障模式设计的重试时间间隔计算方法并不适用于互联网系统间消息通信的常见故障模式，往往会造成故障恢复时间过长。互联网系统间消息通信发生故障的原因一般是网络繁忙造成传输超时、网络设

备发生故障、应用系统发生故障、计划内停机维护等等。这些故障的发生是随机的，但故障的恢复由于人工发现和维护的介入，因此不是随机的，而是具有一定的模式。此外，互联网上的故障的恢复需要的时间往往以小时和天来计，而在这段时间内会积累大量未发出的消息，需要一种高效灵活的消息保存、管理与重传调度算法和系统。因此，建立适用于互联网消息重传方法和系统，在系统资源占用和消息通知恢复的及时性之间取得灵活的平衡，是很有意义的工作。

发明内容

本发明要解决的技术问题是提供一种消息重发方法和系统，用尽可能少的重试次数、尽可能及时地将消息发送到消息接收方。特别，本发明是针对互联网上消息通知失败的常见原因而设计的。

本发明提供了一种消息重发系统，该系统包括：数据库，用于保存待发送的新消息和等待重试的消息；业务插件，用于根据是否有返回结果，以及返回结果的格式和内容是否有效决定本次发送是否需要重试；通知执行器，用于发送新消息或重试的消息，若不需要重试，则从所述数据库中删除本条消息；否则对应所述消息设置重试周期，以当前时间加上对应的重试周期作为重试时间，并更新所述数据库中的对应消息；所述重试周期随着重试次数的增加而增大；通知恢复器，用于检查等待重试的消息，如果存在重试时间到期的消息，则通知所述通知执行器。优选的，所述的通知恢复器定时运行。

本发明还提供了一种系统间消息通知方法，包括以下步骤：

将通知消息存储；

业务插件根据是否有返回结果，以及返回结果的格式和内容是否有效决定本次发送是否需要重试；

发送存储的新通知消息或重试的通知消息；

若不需要重试，则删除存储的本条通知消息；否则对该通知消息设置重

试周期，以当前时间加上对应的重试周期作为重试时间，并更新存储；所述重试周期随着重试次数的增加而增大；

检查等待重试的通知消息，如果存在重试时间到期的通知消息，则执行发送步骤。

优选的，所述系统间消息通知方法还包括：同时发送新通知消息以及重试的通知消息。优选的，所述检查为定时检查。

与现有技术相比，本发明具有的优点是：

所述方法和系统针对短期和中长期通信故障模式提出现实、高效的超时和消息重发解决方案。随着重试次数的增加，其重试周期也越来越长。重试周期的变化，兼顾了通知恢复的及时性和效率。对于短期的通信故障，系统能够在故障恢复之后及时将消息通知送达接收方；对于中长期通信故障，系统能够避免多次重试的开销。

附图说明

下面结合附图和具体实施方式对本发明作进一步详细的说明。

图 1 是本发明重试周期确定方法的概念模式图；

图 2 是一种可靠的系统间消息通知系统的系统结构示意图；

图 3 是本发明一种消息重发方法的执行步骤流程图；

图 4 是消息通知的恢复流程图；

图 5 是一种可靠的系统间消息通知方法的步骤流程图；

图 6 是消息通知的注册流程图；

图 7 是消息通知的发送流程图。

具体实施方式

基于互联网的电子商务在当今社会中占据了越来越重要的地位。越来越多的商业实体通过互联网组织信息流、资金流和物流，以消息通知作为交互模式的互联网应用系统也越来越多。以银行为例，用户在银行完成支付后，银行需要将用户支付的情况通过消息通知的形式告知商户系统；以第三方安全交易平台为例，当用户在交易平台上推进交易流程后，交易平台需要将交易的当前状态通知相关的外部商户系统。这些都是典型的基于互联网的消息通知。本发明适用于任何协作的系统之间，但特别是针对互联网上消息通知失败的常见原因而设计的，因此以下优选消息通知为例进行说明。

互联网上跨系统的消息通知失败的可能原因有以下几种：网络暂时繁忙，造成传输协议超时；网络短暂中断；对方服务器暂时繁忙，造成不响应请求；对方系统存在 BUG，造成不响应请求或者请求处理出错；对方服务器宕机，造成不响应请求；网络长时间故障，造成长达数小时以至于数天的中断；对方系统长时间故障，造成长达数小时以至于数天不可用；对方系统不存在或者永久关闭。

从上述原因可见，造成消息通知不成功的原因既有可能在数分钟内消失或缓解，也有可能长达数小时或者数天。消息通知的最佳重试策略应为：当原因可以在数分钟内消失或缓解时，消息通知能够及时地发送给对方；而当原因需要数小时或者数天才能消失时，系统也能将消息通知发送给对方，而且不做太多无谓的尝试。因此，本发明优选采用下述的一种重试周期确定方法进行重试时间间隔的确定。所述重试时间间隔即为重试周期。

参照图 1，是本发明重试周期确定方法的概念模式图。对所述确定方法

说明如下:

假定有一组编号从 1 到 $n+1$ 的盒子, 1 到 n 号盒子上都绑定一个定时器, 盒子的编号越大, 定时间隔越长, 例如第一次间隔 2 分钟, 第二次间隔 5 分钟, 第 3 次间隔 10 分钟等等。第 $n+1$ 号盒子上没有定时器, 认为消息接收方永久不可达, 放弃自动重试, 等待人工恢复。

一个消息通知任务第 i 次 ($i = 1..n$) 发送不成功, 则进第 i 个盒子;

第 i 个盒子的定时器每次触发都会重试本盒子中的所有消息通知任务;

当一个消息通知任务第 $n+1$ 次发送不成功时, 则进第 $n+1$ 个盒子, 由于第 $n+1$ 个盒子上没有绑定定时器, 因此系统将会放弃对该消息通知任务的自动重试, 转入人工处理。

此外, 本发明中的通知系统针对不同类型外部系统的特点, 对盒子上的定时器进行了调整, 使之符合相应外部系统的故障模式。

如上所述, 本发明方法的核心思想是: 重试周期随着重试次数的增加而增大。采用图 1 所示的重试周期确定方法, 如果消息通知任务重试的次数较少, 说明造成消息通知发送不成功的原因在很短时间内消失了, 则由于每次重试的时间间隔也较小, 故可以保证故障恢复时消息通知的及时发送; 如果消息通知任务重试的次数较多, 说明造成消息通知发送不成功的原因属于长期通信故障, 则后来重试的时间间隔也较长, 可以有效避免太多无效的重试尝试, 减少系统资源占用, 但也可以保证消息通知的确定送达。即上述重试周期确定方法确保了短期通信故障能够及时恢复消息通知, 而对于长期通信故障则能够避免大量无效的通知尝试, 兼顾了通知恢复的及时性和效率。

在上述说明中, 放弃自动重发的条件是以重试次数达到预置的值进行说明的, 但本发明并不局限在这一条件下, 当重试周期达到预置的值时, 也会放弃自动重发。根据不同的应用系统或不同业务情况, 选择任一方式设置预置值, 都是为了避免无限制的重发尝试。

本发明所述的重试周期确定方法中，优选的，重试周期的增加值随着重试次数的增加而变化，即当前的重发周期和上次重发周期的差值与下次重发周期和当前重发周期的差值是不同的。如在图 1 中，第 3 个盒子与第 2 个盒子的定时间间隔的差值是 5 分钟，而第 4 个盒子与第 3 个盒子的定时间间隔的差值是 20 分钟，即重试周期的增加值如 5 分钟和 20 分钟，并不是固定变化的。当然，也不排除采用重试周期的增加值固定变化的方法。对于优选方法，所述重试周期增加值的变化是根据不同的外部系统设置的。针对不同类型外部系统的特点，每种情况下重试周期增加值的大小和重试时间间隔值是不同的。这样的设置兼顾了通知恢复的及时性和效率，更符合应用需求。

参照图 2，是一种可靠的系统间消息通知系统的系统结构示意图，包含本发明一种消息重发系统。所述消息重发系统包括：

数据库 13，用于保存待发送的新消息和等待重试的消息；

通知执行器 152，用于发送新消息或重试的消息，若消息发送成功，则从数据库 13 中删除本条消息；否则对应所述消息设置重试周期，以当前时间加上对应的重试周期作为重试时间，并更新数据库 13 中的对应消息；所述重试周期随着重试次数的增加而增大；

通知恢复器 151，用于检查等待重试的消息，如果存在重试时间到期的消息，则通知通知执行器 152。

如图 2 所示，业务应用 11 可以为某个商业应用，它是消息通知的发起者。外部系统 17 可以为某个商业系统，它是消息通知的接收者。本发明所提供的通知系统用于发送所述业务应用 11 发来的通知消息，并能够确保该通知消息通过互联网可靠到达所述的外部系统 17。

数据库 13 可以是任何通用的关系型数据库，用以保存待发送的新通知消息和曾经发送不成功等待重发的通知消息。通知客户端 12 可以为供业务应用 11 使用的一个客户端模块，嵌入到消息的业务系统中，业务应用 11 可以使

用通知客户端 12 注册一个通知消息。一旦通知消息注册成功，系统就能够确保该通知消息到达消息的接收方，即使此时消息接收方不在线，或者网络暂时无法连通。所述注册可以为，通知客户端 12 将业务应用 11 发来的通知消息存放到数据库中，并通过消息队列 14 触发消息即时发送。所述业务系统是指针对业务应用 11，进行相应的业务处理的系统。结合网上银行与商户之间的消息通知实例，所述消息的事务处理是指用户在网上银行进行实际支付，若支付完成则事务处理提交，否则还处于事务处理中；所述通知消息指用户的支付结果，即是否完成支付。所述事务是一个技术上的专有名词，表示具有 ACID 特性的一组操作，事务的 ACID 特性是实现消息通知与业务操作 100% 一致性的关键。当创建事务时，应确保事务具有某些可以自行管理的特性，这些特性就称为 ACID 特性。ACID 是指：原子性 (Atomicity)、一致性 (Consistency)、隔离性 (Isolation) 和持久性 (Durability)。

将通知客户端 12 嵌入到消息的业务系统中，可以在同一个数据库事务中完成业务处理以及通知任务的注册，完全可以避免业务处理和通知任务不一致的情况。所述一致为一通知消息只有当事务处理完成提交后，该通知消息才注册成功，进行即时发送。所述数据库事务中的数据库可以是系统中保存消息通知请求的数据库。一般说来，这个数据库与业务系统保存业务数据的数据库是同一个。如果是同一个数据库，则可以避免使用复杂低效的分布式事务机制就能实现通知消息的注册与业务数据的处理位于同一数据库事务中。但也可以使用不同的数据库，并通过分布式事务机制保证通知消息注册与业务数据处理的一致性。

消息队列 14 是一种标准的用于系统间异步消息通讯的中间件。通过使用消息队列，消息的发送过程和接收过程之间是异步的，同时使得消息发送方和消息接收方不直接进行通讯，而是通过消息队列进行间接通讯。这样做最大程度地减小了消息发送方和消息接收方之间的相互依赖，使得消息发送方

与消息接收方之间可以相对独立地进行工作。目前的消息队列一般都能够做到当接收到消息之后，只要消息接收方工作正常，可以即时将消息传递给消息接收方。本实施例中，消息队列 14 用以触发通知消息的即时发送。

通知服务器 15 可以为一台或一组独立运行的，用以专门负责通知消息发送和通知消息重发的服务器，可以包括通知恢复器 151、通知执行器 152 以及多种协议适配器 153。其中，通知恢复器 151 为负责调度未成功发送的通知消息在恰当的时间进行重新发送的模块；通知执行器 152 为负责执行实际的通知消息发送的模块；多种协议适配器 153 中，每种协议适配器支持一种传输协议，通过互联网完成与外部系统 17 的实际数据传输。通知业务处理插件库 16 为各种业务插件 161 的管理器。业务插件 161 负责通知消息发送前的预处理和返回的通知结果处理，所述预处理和结果处理均与业务应用紧密相关，不同的业务应用，其处理过程也不相同。

参照图 3，是本发明一种消息重发方法的执行步骤流程图。当消息第一次发送不成功时，进入消息重发步骤：

步骤 g1，计算消息的下一次重发时间。通知执行器根据重试周期确定方法计算下一次重试需要等待的时间间隔，并用该间隔加上当前时间作为下一次消息发送重试的时间点。

步骤 g2，更新数据库存储。通知执行器根据计算的重试时间，更新数据库中对应消息通知请求的发送时间，并放弃本轮通知，等待通知恢复器的重新调度。

步骤 g3，在定时时间点调度重试时间到期的消息并发送。参照图 4，是消息通知的恢复流程图。当所述通知消息需要重新发送时，进入通知恢复流程。优选的，通知恢复器定时运行，在每一轮运行中，执行以下步骤：

步骤 c1，通知恢复器等待定时时间，当时间点到开始运行。

步骤 c2，从数据库中查找重试时间到期的消息通知请求。检查需要重试

的消息通知请求，将通知执行器计算所得的重试时间与本轮的定时时间相比较，若有重试时间点小于或等于定时时间点的情况，则存在重试时间到期的消息通知请求，继续步骤 c3；否则放弃本轮重试，继续步骤 c4。

步骤 c3，对于重试时间到期的通知消息，向通知执行器发送重试请求，逐条交给通知执行器进行通知发送。

步骤 c4，等待下一定时时间点，进行下一轮运行。

通知恢复流程的定时时间一般是固定的，比如每隔 1 分钟运行一次。定时时间间隔的大小关系到通知恢复的及时性，因此，这个时间间隔一般设置得越小越好，但必须在通知服务器能够承受的性能范围内。此外，这个定时时间间隔与消息通知重试的间隔并不相同，消息通知重试的间隔时间是通过重试周期确定方法针对每一条通知消息单独计算和设置的。

在步骤 g3 中，若所述步骤 c3 的发送成功，则本次消息重发成功完成；否则继续执行步骤 g1。

本发明还提供了一种可靠的系统间消息通知方法，参照图 5，是一种可靠的系统间消息通知方法的步骤流程图。当业务应用需要向外部系统发送一通知消息时，执行以下步骤即可可靠送达：

步骤 s1，业务应用将自己的通知需求封装成一个消息通知请求，发送至通知客户端。

步骤 s2，通知客户端将所述通知消息进行注册。一旦通知消息注册成功，业务应用就可以继续其它业务处理，所述系统就能够确保该通知消息到达消息的接收方，即使此时消息接收方不在线，或者网络暂时无法连通。

步骤 s3，当通知消息注册成功后，通知系统进行通知消息的发送。若发送成功，则消息通知任务结束；否则继续步骤 s4。

步骤 s4，进行消息发送的恢复处理，调度未成功发送的通知消息在恰当的时间进行重新发送，继续步骤 s3。

在此过程中，步骤 s4 与步骤 s1、步骤 s2 和步骤 s3 可以同时执行。步骤 s4 即上述的步骤 g3，可参照图 4 消息通知的恢复流程图。下面分别对步骤 s2 和 s3 进行详细说明。

参照图 6，是消息通知的注册流程图，包括以下步骤：

步骤 a1，通知客户端接收所述业务应用发来的消息通知请求；所述消息通知请求可以包括消息通知的内容。

步骤 a2，通知客户端将所述通知请求存放到数据库中，直到通知消息发送成功，这样可以避免在发送过程中由于网络、服务器或程序系统不稳定造成的通知消息丢失且不可恢复的问题。

步骤 a3，通知客户端判断所述消息的事务处理是否完成提交，若还处于事务处理中，则继续执行步骤 a4、步骤 a5 和步骤 a6；若事务已完成提交，则通知客户端自动触发消息队列，直接执行步骤 a6。结合网上银行与商户之间的消息通知实例，所述消息的事务处理是指用户在网上银行进行实际支付，若支付完成则事务处理提交，否则还处于事务处理中；所述通知消息指用户的支付结果，即是否完成支付。本实施例中，由于通知客户端嵌入到业务系统中，能够在业务事务内完成将通知任务注册到数据库中，这样避免使用复杂的机制就实现了业务处理和通知注册的一致性。

步骤 a4，通知客户端向事务管理器注册一个事务同步器。所述事务管理器是对事务进行管理的软件，包括事务同步器。所述事务同步器设置在通知客户端，用于实现在事务提交之后通过消息队列向通知服务器发送一个立即发送通知消息的请求，可以保证通知的及时性。

步骤 a5，当事务完成提交之后，事务同步器自动触发消息队列。

步骤 a6，消息队列向通知执行器发送一个立即发送通知消息的请求，驱动通知执行器进行发送，保证了通知的即时性。

参照图 7，是消息通知的发送流程图，包括以下步骤：

步骤 b1, 通知执行器接收到通知客户端通过消息队列发来的新消息通知请求, 或者消息恢复器发来的重试消息通知请求。如果同时收到消息队列发来的新消息通知请求和消息恢复器发来的重试消息通知请求, 则优选的, 通知执行器执行多线程处理, 可以同时发送新通知消息以及重试的通知消息。当然, 也可以采用确定优先级进行发送。

步骤 b2, 通知执行器根据消息通知请求的类型, 从业务插件库中找到相应的业务插件, 将消息通知请求交给业务插件进行预处理, 得到实际的外部系统通知地址、通知协议和通知参数。

步骤 b3, 通知执行器根据通知协议选择合适的协议适配器, 将通知消息内容、通知地址和通知参数交给协议适配器进行实际的消息发送。

步骤 b4, 如果网络、双方服务器与系统均正常工作, 则消息通过互联网到达外部系统, 外部系统接收到消息并处理后返回处理结果; 否则所述通知执行器能够自动检测到通知消息未到达外部系统, 继续步骤 b8 和 b9。

步骤 b5, 通知执行器将返回结果交给业务插件, 由业务插件完成相应的业务处理。在上述例子中, 当收到外部商户的返回结果之后, 业务插件需要将交易的状态变为已发货, 并且记录发货单的详情, 通知用户发货情况。由于通知服务器本身是通用的, 与具体业务无关, 因此这些处理是由业务插件来完成的。

步骤 b6, 业务插件判断本次发送是否需要重试。业务插件是根据是否有返回结果, 以及返回结果的格式和内容是否有效决定。仍以上述例子为例, 如果返回结果中包含有效的商户发货信息, 则业务插件判断消息通知是成功的; 如果没有返回结果, 或者返回结果格式无效, 或者外部商户在返回结果中直接表明暂时无法处理, 需要过一段时间重试, 则业务插件判断消息通知未成功, 需要重试。

因为消息发送不成功的可能原因不但包括网络或系统故障，还取决于业务处理是否能够成功，而通用的通知服务器可以判定网络与系统的故障，但无法判定业务处理是否成功，因此将判定职责交给业务插件去完成。当然，明显的网络与系统故障是通知服务器本身能够判断的，因此，当发生这种情况时，由通知服务器直接决定需要重试。

步骤 b7，如果业务插件表明本次消息不需要重试，则通知执行器从数据库中删除本条消息通知请求，成功完成通知发送。

步骤 b8，如果由于网络等原因发送的通知消息未到达外部系统，或业务插件表明本次消息需要重试，则通知执行器根据重试策略计算下一次重试需要等待的时间间隔，并用该间隔加上当前时间作为下一次消息发送重试的时间点。

步骤 b9，对于等待重新发送的通知消息，通知执行器根据计算的重试时间，更新数据库中对应消息通知请求的发送时间，并放弃本轮通知，等待通知恢复器的重新调度。

在上述通知发送流程中，通过扩展业务插件，可以实现在同一个消息通知系统中支持各种类型的业务通知；通过扩展协议处理器，可以实现在同一个消息通知系统上支持不同协议的通知。此外，通知的接收者不需要实现任何特殊的可靠消息通知协议，只需要根据业务的要求返回消息处理的结果，就能够可靠接收来自消息提供方的消息。

上述本发明一种消息重发方法和系统，用尽可能少的重试次数、尽可能及时地将消息发送到消息接收方。所述方法和系统针对短期和中长期通信故障模式提出现实、高效的超时和消息重发解决方案。随着重试次数的增加，其重试周期也越来越长。重试周期的变化，兼顾了通知恢复的及时性和效率。对于短期的通信故障，系统能够在故障恢复之后及时将消息通知送达接收方；对于中长期通信故障，系统能够避免多次重试的开销。

本发明还提供了一种可靠的系统间消息通知方法和系统，适用于任何协作的系统之间，尤其解决了系统之间通过互联网进行消息通知时，由于网络内在的不可靠性和系统软、硬件故障造成的通知消息丢失且不可恢复的问题，确保消息的可靠、及时到达。本发明支持不同系统间的多协议传输，接收方不需要实现复杂的交互协议就能够可靠接收通知消息，适合广泛应用于互联网。所述方法和系统还支持多业务处理，可作为一个通用的业务应用平台。同时，可以进行多业务和多协议的灵活扩展。

以上对本发明所提供的一种消息重发方法和系统、以及一种可靠的系统间消息通知方法和系统进行了详细介绍，本文仅以优选实施例对本发明进行说明。特别是，本发明所述的方法和系统适用于任何协作的系统之间，并不仅局限在基于互联网的系统之间，只是优选保证了互联网间的消息重发和可靠消息通知，更适合广泛应用于互联网。所以，并不能因此即局限本发明的权利范围，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。综上所述，本说明书内容不应理解为对本发明的限制。

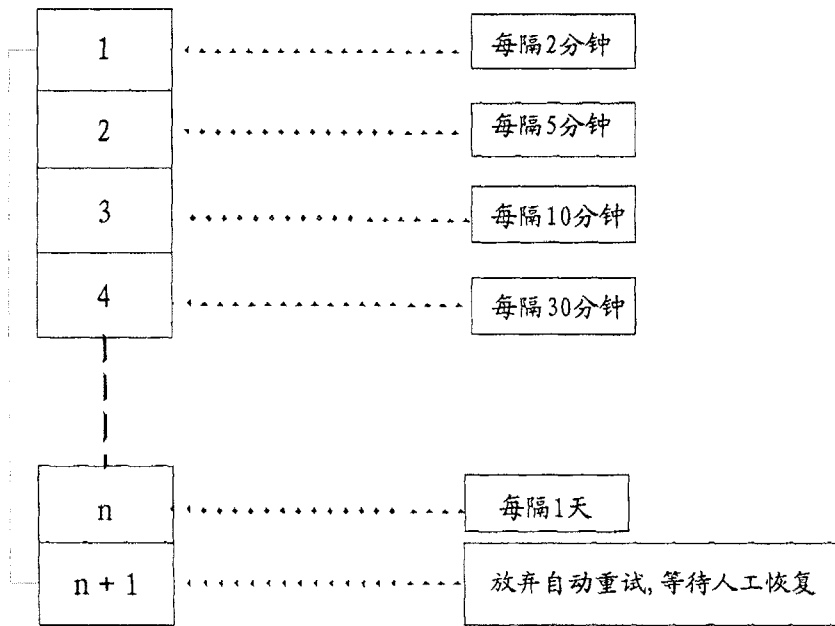


图 1

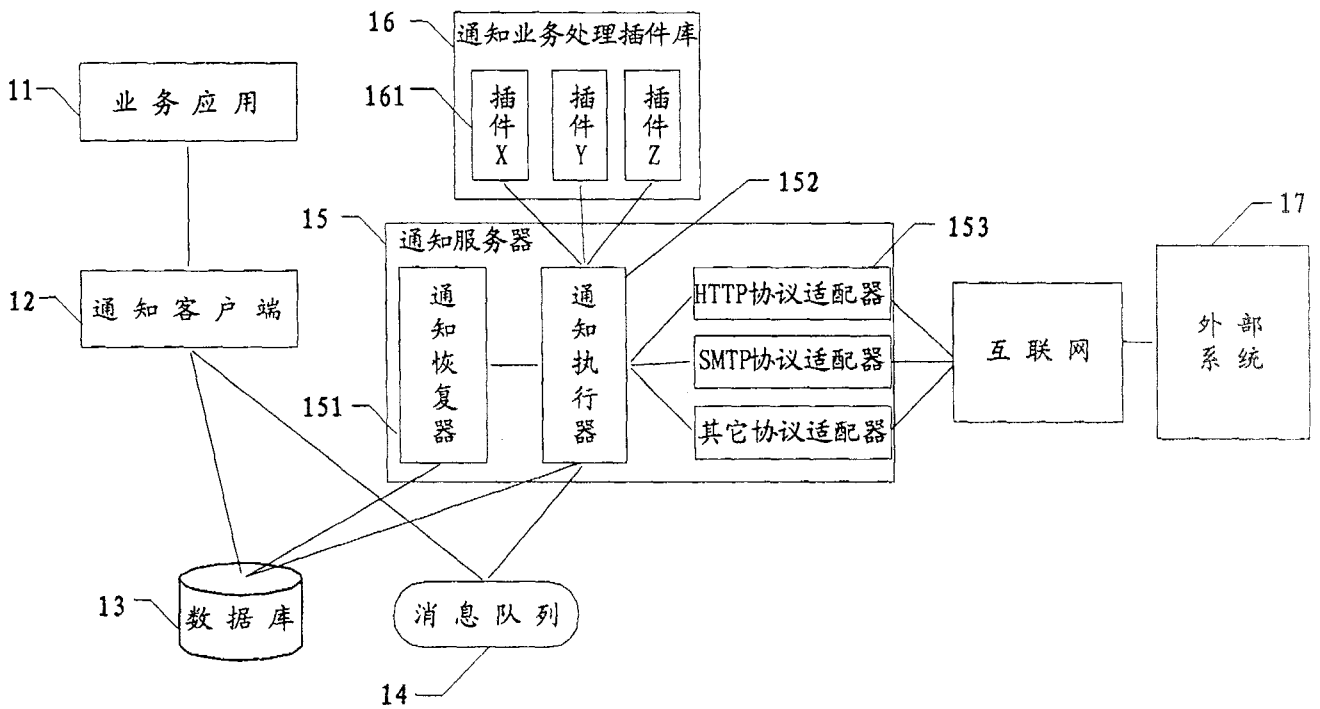


图 2

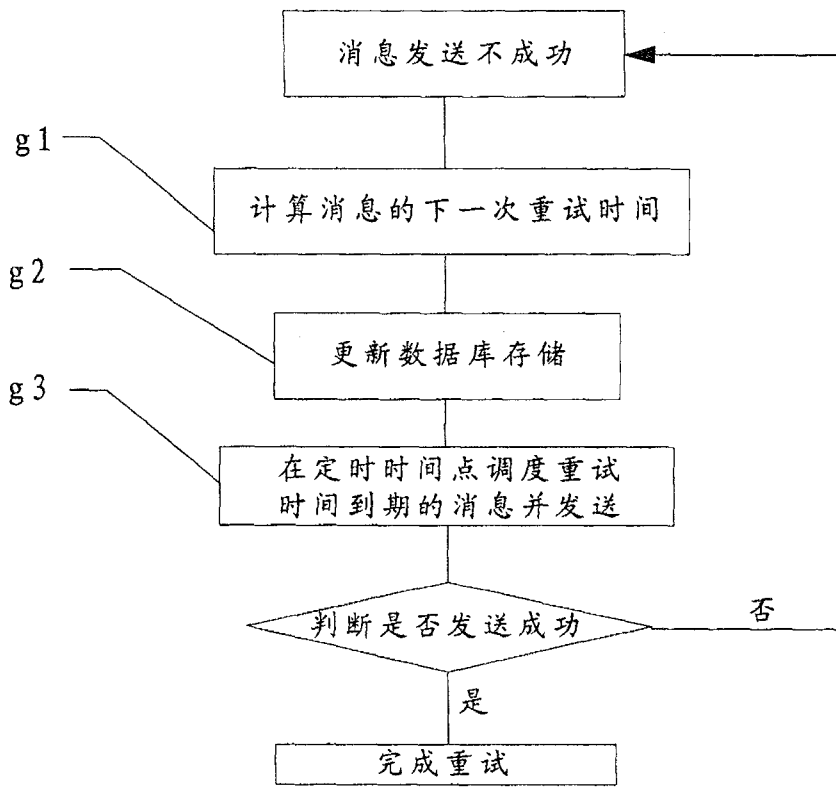


图 3

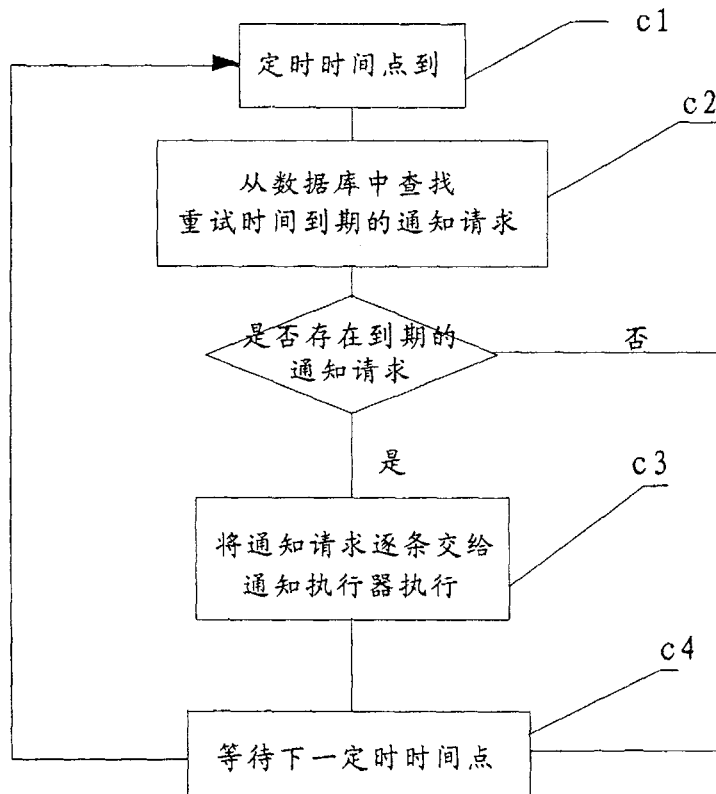


图 4

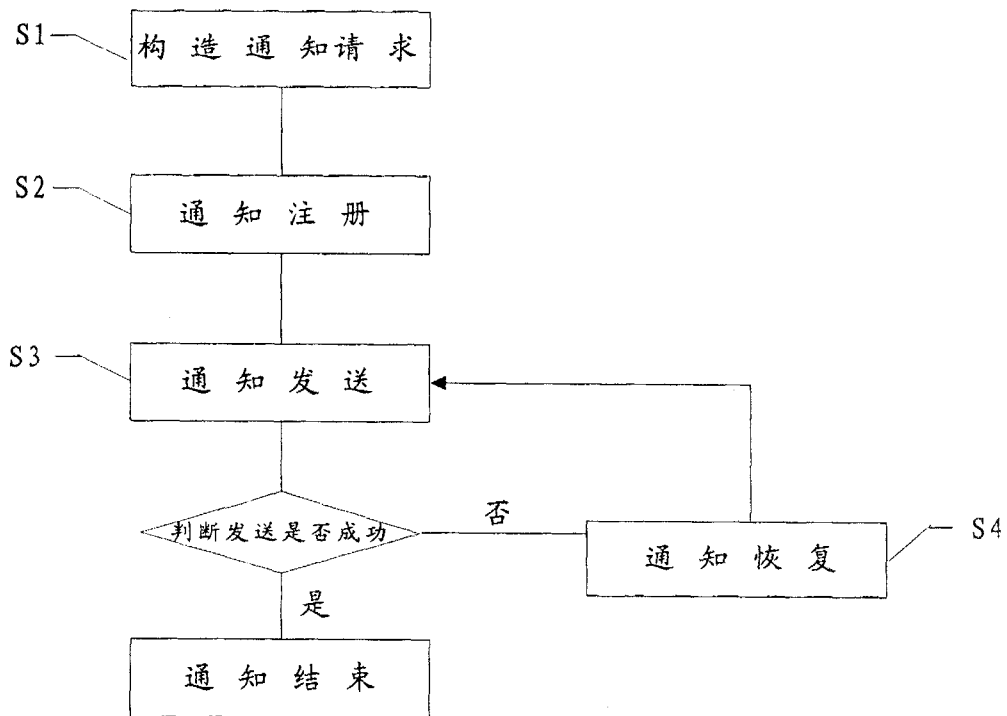


图 5

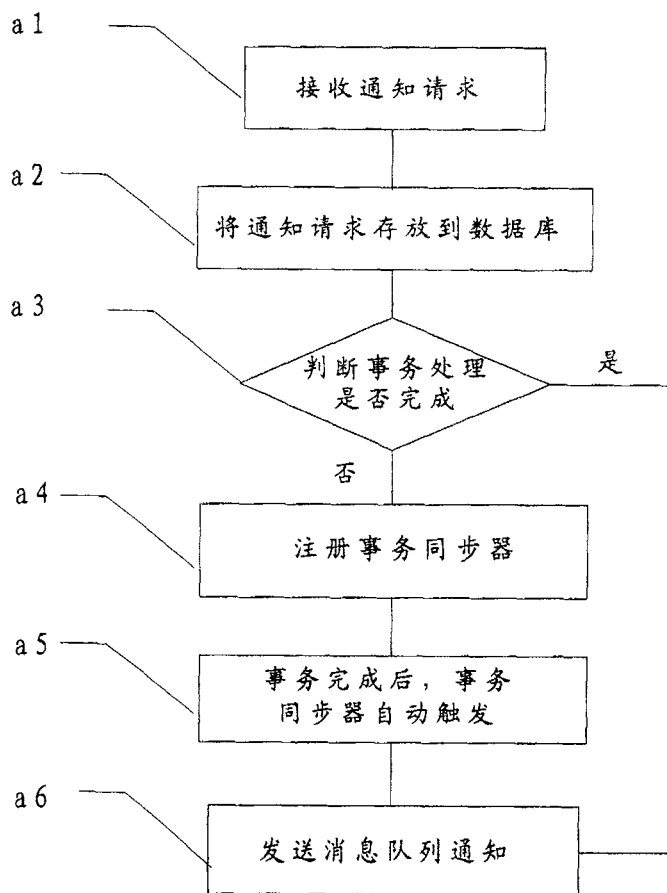


图 6

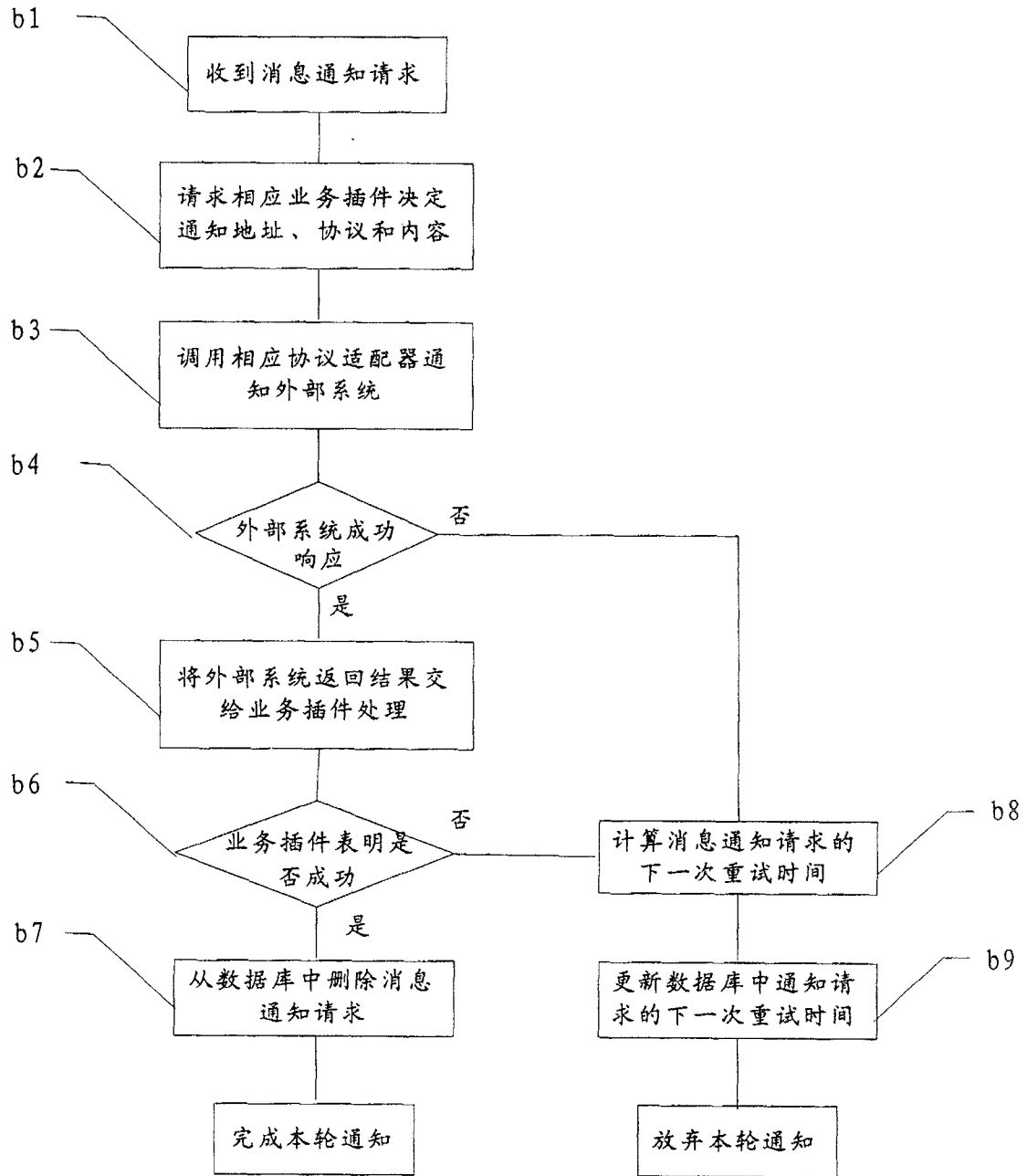


图 7