



(12)发明专利申请

(10)申请公布号 CN 108650192 A

(43)申请公布日 2018.10.12

(21)申请号 201810397479.5

H04L 12/823(2013.01)

(22)申请日 2018.04.28

(71)申请人 国网福建省电力有限公司

地址 350003 福建省福州市鼓楼区五四路
257号

申请人 国网福建省电力有限公司电力科学
研究院
福州百榕软件有限公司

(72)发明人 陈泽鑫 曹力立 汤振立 陈嘉鸿

(74)专利代理机构 福州元创专利商标代理有限
公司 35100

代理人 蔡学俊

(51)Int.Cl.

H04L 12/819(2013.01)

H04L 12/833(2013.01)

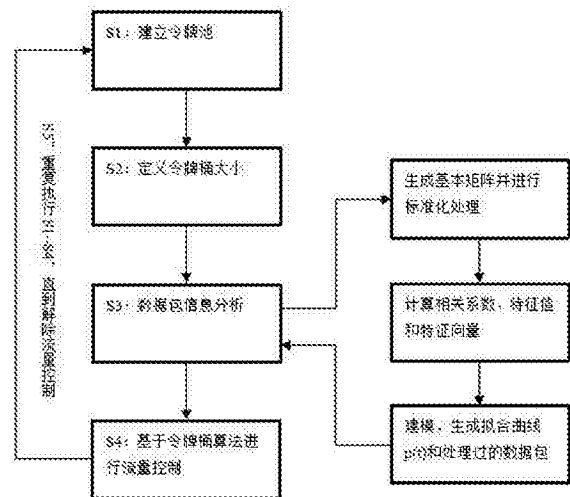
权利要求书2页 说明书4页 附图2页

(54)发明名称

一种基于令牌桶优化算法的流量控制方法

(57)摘要

本发明涉及一种基于令牌桶优化算法的流量控制方法,建立令牌池;设置令牌桶大小;数据包信息分析;分析通过网关的所有数据包,通过分析数据包的源地址、目的地址、源端口、目的端口及协议类型,构造基本向量;基于令牌桶优化算法进行流量控制;根据产生的基本向量,采用单速率三色标记处理通过的数据包,实时检测并控制流量;当一个有n个字节的数据包经过令牌桶时,就从令牌桶中消耗n个令牌,并接收数据包,允许这些数据包根据网络拥塞情况通过、丢弃或不处理;重复执行所述步骤S1~S4,直到收到解除流量控制信号为止。本发明能够快速进行实时、精确地对流量进行控制。



1. 一种基于令牌桶优化算法的流量控制方法,其特征在于,还包括如下步骤:

步骤S1:建立令牌池;令牌 会按一定的速率以 $1/\text{rate}$ 秒周期性或一次性把自身添加到令牌桶中,其中, rate 是算法生成令牌的速率;

步骤S2:设置令牌桶大小;根据历史数据的测量值及未来的预估值设置令牌桶的大小 burst ,每个令牌只允许匹配预设数量的字节数;

步骤S3:数据包信息分析;分析通过网关的所有数据包,通过分析数据包的源地址、目的地址、源端口、目的端口及协议类型,构造基本向量;

步骤S4:基于令牌桶优化算法进行流量控制;根据所述步骤S3产生的所述基本向量,并采用单速率三色标记处理通过的数据包,实时检测并控制流量;当一个有 n 个字节的数据包经过令牌桶时,就从令牌桶中消耗 n 个令牌,并接收数据包,允许这些数据包根据网络拥塞情况通过、丢弃或不处理;

步骤S5:重复执行所述步骤S1~S4,直到收到解除流量控制信号为止。

2. 根据权利要求1所述的一种基于令牌桶优化算法的流量控制方法,其特征在于,在所述步骤S3中,还包括如下步骤:

步骤S31:将所述源地址、目的地址、源端口、目的端口及协议类型5个维度的向量生成一个基本矩阵,对 $X = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$,此基本矩阵进行标准化处理:

$$x_{ij}^* = \begin{cases} \frac{x_{ij} - x_j}{\text{var}(x_j)}, & i = 1, 2, \dots, n; j = 1, 2, \dots, p, \end{cases}$$

其中, $x_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$, $\text{var}(x_j) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - x_j)^2$, $j = 1, 2, \dots, p$;

步骤S32:计算相关系数矩阵:

$$R = \begin{pmatrix} r_{11} & \dots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{n1} & \dots & r_{np} \end{pmatrix},$$

其中, $r_{ij} = \frac{1}{n-1} \sum_{i=1}^n x_{ij} x_{ij}$, $i, j = 1, 2, \dots, p$, 它的特征值 $(\lambda_1, \lambda_2, \dots, \lambda_p)$ 和相应的特征向量 $a_i = (a_{i1}, a_{i2}, \dots, a_{ip})$, $i = 1, 2, \dots, p$;

步骤S33:根据相关系数矩阵、特征值及相应的特征向量进行平滑数据建模,经过移动中值滤波、阈值处理、合成及调整处理后,获得一条与时间 t 相关的光滑拟合曲线 $p(t)$ 和经过处理过的数据包, $p(t)$ 表示可以接受的正常流量与时间 t 相关的光滑拟合曲线,且表示可以接受的正常流量。

3. 根据权利要求1所述的一种基于令牌桶优化算法的流量控制方法,其特征在于,在所述步骤S4中,还包括如下步骤:

步骤S41:记令牌桶填充令牌的平均速率为CIR,有突发流量时系统所允许数据包传输的最大速率为CBS,每次数据包发送或接收时突发允许超过CBS的最大流量速率为EBS,以及正常使用的C令牌桶和超出C容量的突发E令牌桶;两个令牌桶中令牌每隔 $1/\text{CIR}$ 时间就添加

一个令牌,先往C令牌桶中添加令牌;当C令牌桶满后,再往E令牌桶添加;当C令牌桶和E令牌桶都满后,就会丢弃新的令牌;

步骤S42:令 T_c 和 T_e 分别表示C令牌桶和E令牌桶中的令牌数,最初状态下, $T_c=CBS$, $T_e=EBS$;其中,CBS数值和EBS数值分别表示两桶的总容量,到达的数据流速率是 r ,

如果 $r=rate$,每个到达的数据包请求都能匹配到相应的令牌,这些数据包能不被限速地通过网关到达主机系统;

如果 $r<rate$,到达的数据包只消耗部分令牌,在令牌桶里就积累未使用的令牌直到令牌桶被装满,并且剩余的令牌能在突发请求数据时被使用;这些数据包能不被限速地通过网关到达主机系统;

如果 $r>rate$,如果请求的数据包持续地到来,将触发流量控制机制,并产生以下3种情况:

将发生丢包、卡顿、拒绝响应;

对流量限制的数据包做标记后传输,当网络过载时就将这些有标记的包丢弃;

令牌桶中没有足够的令牌对应相应的请求或数据包,通过排队等待后续令牌生成足够的令牌来发送数据包;

步骤S43:令牌桶优化算法根据数据包长度确定包通过或拒绝,记一数据包长度为 L :

a) 如果 $L<T_c$,数据包被标记为绿色,表示此数据包能正常通过系统,发送后C令牌桶剩余的令牌为 (T_c-L) ;

b) 如果 $T_c<L<T_e$,数据包被标记为黄色,从E令牌桶中取走额外的令牌,E令牌桶中的令牌减少 $(L-T_c)$;

c) 如果 $L>T_e$,把数据包标记为红色,系统直接丢弃不发送,这样两个令牌桶中的总令牌都不变。

一种基于令牌桶优化算法的流量控制方法

技术领域

[0001] 本发明涉及电力网络领域,特别是一种基于令牌桶优化算法的流量控制方法。

背景技术

[0002] 随着网络新技术不断出现和新应用的爆炸性发展,以及电力领域对这些新技术的使用导致内部网络的流量呈指数级别的增长。同时,网络结构也越来越复杂,如何让使用者在不影响办公效率的情况下提高用户体验就成了研究的主要方向。

[0003] 迄今为止,国内外在流量控制在如何结合电力领域安全性、体验和效率方面已经有了大量的研究工作,在算法方面的研究仍然主要体现在基于经典算法如reno算法、vegas算法和cubic算法等。采用这些单一、简单的算法会遇上计算结果精度受限、流量控制能力不足等问题。

[0004] 因此,为了应对内部网络流量拥塞情况,提高对突发大流量的防御和控制能力,需要研究新的流量控制优化算法,该优化方法应该能够实时监控通过网关的正常范围值内的流量,发现网络流量异常时,能够提供自动限速并反馈相应情况,并且能够根据流量大小来自适应处理等,提高流量控制能力。

发明内容

[0005] 本发明的目的在于提供一种基于令牌桶优化算法的流量控制方法,以克服现有技术中存在的缺陷。

[0006] 为实现上述目的,本发明的技术方案是:一种基于令牌桶优化算法的流量控制方法,其特征在于,还包括如下步骤:

步骤S1:建立令牌池;令牌会以一定的速率以 $1/\text{rate}$ 秒周期性或一次性把自身添加到令牌桶中,其中, rate 是算法生成令牌的速率;

步骤S2:设置令牌桶大小;根据历史数据的测量值及未来的预估值设置令牌桶的大小 burst ,每个令牌只允许匹配预设数量的字节数;

步骤S3:数据包信息分析;分析通过网关的所有数据包,通过分析数据包的源地址、目的地址、源端口、目的端口及协议类型,构造基本向量;

步骤S4:基于令牌桶优化算法进行流量控制;根据所述步骤S3产生的所述基本向量,并采用单速率三色标记处理通过的数据包,实时检测并控制流量;当一个有 n 个字节的数据包经过令牌桶时,就从令牌桶中消耗 n 个令牌,并接收数据包,允许这些数据包根据网络拥塞情况通过、丢弃或不处理;

步骤S5:重复执行所述步骤S1~S4,直到收到解除流量控制信号为止。

[0007] 在本发明一实施例中,在所述步骤S3中,还包括如下步骤:

步骤S31:将所述源地址、目的地址、源端口、目的端口及协议类型5个维度的向量生成

一个基本矩阵 $X = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mp} \end{pmatrix}$, 对此基本矩阵进行标准化处理:

$$x_{ij} = \begin{cases} (x_{ij} - x_j) \\ \text{var}(x_j) \end{cases}, i = 1, 2, \dots, n; j = 1, 2, \dots, p,$$

其中, $x_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$, $\text{var}(x_j) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - x_j)^2$, $j = 1, 2, \dots, p$;

步骤S32: 计算相关系数矩阵:

$$R = \begin{pmatrix} r_{11} & \dots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{n1} & \dots & r_{np} \end{pmatrix},$$

其中, $r_{ij} = \frac{1}{n-1} \sum_{t=1}^n x_{it}x_{jt}$, $ij = 1, 2, \dots, p$, 它的特征值 $(\lambda_1, \lambda_2, \dots, \lambda_p)$ 和相应的特征向量 $a_i = (a_{i1}, a_{i2}, \dots, a_{ip})$, $i = 1, 2, \dots, p$;

步骤S33: 根据相关系数矩阵、特征值及相应的特征向量进行平滑数据建模, 经过移动中值滤波、阈值处理、合成及调整处理后, 获得一条与时间 t 相关的光滑拟合曲线 $p(t)$ 和经过处理过的数据包, $p(t)$ 表示可以接受的正常流量与时间 t 相关的光滑拟合曲线, 且表示可以接受的正常流量。

[0008] 在本发明一实施例中, 在所述步骤S4中, 还包括如下步骤:

步骤S41: 记令牌桶填充令牌的平均速率为CIR, 有突发流量时系统所允许数据包传输的最大速率为CBS, 每次数据包发送或接收时突发允许超过CBS的最大流量速率为EBS, 以及正常使用的C令牌桶和超出C容量的突发E令牌桶; 两个令牌桶中令牌每隔 $1/\text{CIR}$ 时间就添加一个令牌, 先往C令牌桶中添加令牌; 当C令牌桶满后, 再往E令牌桶添加; 当C令牌桶和E令牌桶都满后, 就会丢弃新的令牌;

步骤S42: 令 T_c 和 T_e 分别表示C令牌桶和E令牌桶中的令牌数, 最初状态下, $T_c = \text{CBS}$, $T_e = \text{EBS}$; 其中, CBS数值和EBS数值分别表示两桶的总容量, 到达的数据流速率是 r ,

a) 如果 $r = \text{rate}$, 每个到达的数据包请求都能匹配到相应的令牌, 这些数据包能不被限速地通过网关到达主机系统;

b) 如果 $r < \text{rate}$, 到达的数据包只消耗部分令牌, 在令牌桶里就积累未使用的令牌直到令牌桶被装满, 并且剩余的令牌能在突发请求数据时被使用; 这些数据包能不被限速地通过网关到达主机系统;

c) 如果 $r > \text{rate}$, 如果请求的数据包持续地到来, 将触发流量控制机制, 并产生以下3种情况:

- i. 将发生丢包、卡顿、拒绝响应;
- ii. 对流量限制的数据包做标记后传输, 当网络过载时就将这些有标记的包丢弃;
- iii. 令牌桶中没有足够的令牌对应相应的请求或数据包, 通过排队等待后续令牌生成足够的令牌来发送数据包;

步骤S43: 令牌桶优化算法根据数据包长度确定包通过或拒绝, 记一数据包长度为 L :

a) 如果 $L < T_c$, 数据包被标记为绿色, 表示此数据包能正常通过系统, 发送后C令牌桶剩余的令牌为 $(T_c - L)$;

b) 如果 $T_c < L < T_e$, 数据包被标记为黄色, 从E令牌桶中取走额外的令牌, E令牌桶中的令

牌减少(L-Tc)；

c) 如果 $L > T_e$, 把数据包标记为红色, 系统直接丢弃不发送, 这样两个令牌桶中的总令牌都不变。

[0009] 相较于现有技术, 本发明具有以下有益效果: 发明提出的一种基于令牌桶优化的流量控制方法, 综合效率优先和先行处理数据包等的组合方法进行求解, 能够快速进行实时、精确地对流量进行控制, 有效平衡了用户体验和办公效率的问题。

附图说明

[0010] 图1为本发明中基于令牌桶优化的流量控制方法的流程图。

[0011] 图2为本发明中基于令牌桶优化算法进行流量控制的示意图。

具体实施方式

[0012] 下面结合附图, 对本发明的技术方案进行具体说明。

[0013] 如图1和图2所示, 本实施例提供了一种基于令牌桶优化的流量控制方法, 具体包括以下步骤:

步骤S1: 建立令牌池: 令牌 token 会按一定的速率以 $1/\text{rate}$ 秒周期性或一次性把自身添加到令牌桶bucket中, 建立令牌桶算法所需的令牌池, 其中rate是算法生成令牌token的速率;

步骤S2: 定义令牌桶大小: 根据历史数据的测量值及未来的预估值来定义令牌桶bucket的大小burst, 每个token只允许匹配一定数量的字节数;

步骤S3: 数据包信息分析: 主要分析通过网关的所有数据包, 分析数据包的源地址、目的地址、源端口、目的端口及协议类型这5个维度来构造基本向量;

步骤S4: 基于令牌桶优化算法进行流量控制: 根据步骤S3产生的基本向量并应用单速率三色标记来处理通过的数据包, 从而实时检测并控制流量。当一个有n个字节的数据包经过bucket时, 就从bucket中消耗n个token, 并接收数据包, 允许这些数据包根据网络拥塞情况通过、丢弃或不处理;

步骤S5: 重复执行步骤S1~S4, 直到管理员解除流量控制。

[0014] 进一步的, 在本实施例中, 步骤S3具体包括以下步骤:

步骤S31: 将5个维度的向量生成一个基本矩阵 $x = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$, 对此基本矩阵进行

标准化处理: $x'_{ij} = \frac{x_{ij} - x_j}{\sqrt{\text{var}(x_j)}} (i = 1, 2, \dots, n; j = 1, 2, \dots, p)$,

其中, $x_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$, $\text{var}(x_j) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - x_j)^2 (j = 1, 2, \dots, p)$;

步骤S32: 计算相关系数矩阵 $R = \begin{pmatrix} r_{11} & \dots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{n1} & \dots & r_{np} \end{pmatrix}$, 其中, $r_{ij} = \frac{1}{n-1} \sum_{i=1}^n x_{i0} x_{ij} (i, j = 1, 2, \dots, p)$,

它的特征值 $(\lambda_1, \lambda_2, \dots, \lambda_p)$ 和相应的特征向量 $a_i = (a_{i1}, a_{i2}, \dots, a_{ip}), i = 1, 2, \dots, p$;

步骤S33: 以基本矩阵的相关系数矩阵、特征值及相应的特征向量进行平滑数据建模,

经过移动中值滤波、阈值处理、合成及调整等一系列处理后,获得一条与时间 t 相关的光滑拟合曲线 $p(t)$ 和经过处理过的数据包, $p(t)$ 表示可以接受的正常流量与时间 t 相关的光滑拟合曲线, $p(t)$ 表示可以接受的正常流量。

[0015] 进一步的,在本实施例中,步骤S4具体包括以下步骤:

步骤S41:定义令牌桶填充令牌的平均速率为CIR,有突发流量时系统所允许数据包传输的最大速率为CBS,每次数据包发送或接收时突发允许超过CBS的最大流量速率为EBS,以及正常使用的C令牌桶和超出C容量的突发E令牌桶;两个令牌桶中令牌每隔 $1/CIR$ 时间就添加一个令牌,先往C令牌桶中添加令牌;当C令牌桶满后,再往E令牌桶添加;当C令牌桶和E令牌桶都满后,就会丢弃新的令牌;

步骤S42:令 T_c 和 T_e 分别表示C令牌桶和E令牌桶中的令牌数,最初状态下, $T_c=CBS$, $T_e=EBS$;其中,CBS数值和EBS数值分别表示两桶的总容量,到达的数据流速率是 r ,

a) 如果 $r=rate$,每个到来的数据包请求都能匹配到相应的token,这些数据包可以不被限速地通过网关到达主机系统;

b) 如果 $r<rate$,到达的数据包只消耗了一部分token,在bucket里就会积累未用的token直到bucket被装满,并且剩余的token可以在突发请求数据时被使用,这些数据包也可以不被限速地通过网关到达主机系统;

c) 如果 $r>rate$,这种情况表明token会很快被用完,导致的结果可能会让网络变得拥塞或中断一段时间。如果请求的数据包持续地到来,将触发流量控制机制,将发生以下3种情况:

- i. 将发生丢包、卡顿、拒绝响应等情况;
- ii. 对这些流量限制的数据包做标记后传输,当网络过载时就将这些有标记的包丢弃;
- iii. bucket中没有足够的token对应相应的请求或数据包,它们就会排队等待后续token生成足够的token来发送数据包。

[0016] 步骤S43:算法会根据数据包长度来确定包通过或拒绝,现有某一个数据包长度为 L ,

a) 如果 $L<T_c$,数据包被标记为绿色,表示此包可正常通过系统,发送后C bucket剩余的token为 (T_c-L) ;

b) 如果 $T_c<L<T_e$,数据包被标记为黄色,这时要从E bucket中取走额外的token,E bucket中的token减少 $(L-T_c)$;

c) 如果 $L>T_e$,把数据包标记为红色,系统直接丢弃不发送,这样两个bucket中的总token都不变。

[0017] 综上所述,本发明对通过网关的数据流量进行标准化处理后,采用令牌桶算法作为核心控制算法,并对此算法的一些步骤进行相应的优化,使之能够更有效率地、更精确地处理数据包,提高了电网的内部网络对网络流量的把控能力。

[0018] 以上是本发明的较佳实施例,凡依本发明技术方案所作的改变,所产生的功能作用未超出本发明技术方案的范围时,均属于本发明的保护范围。

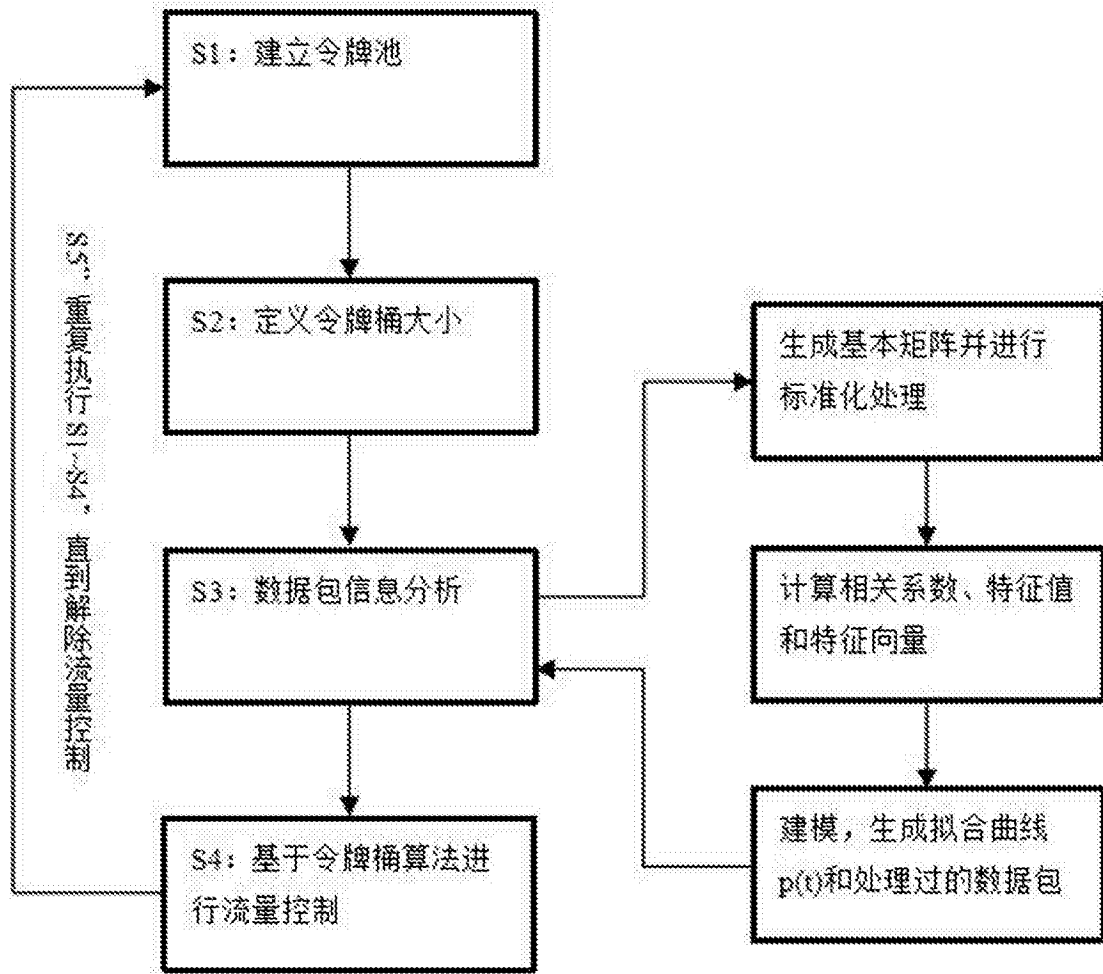


图1

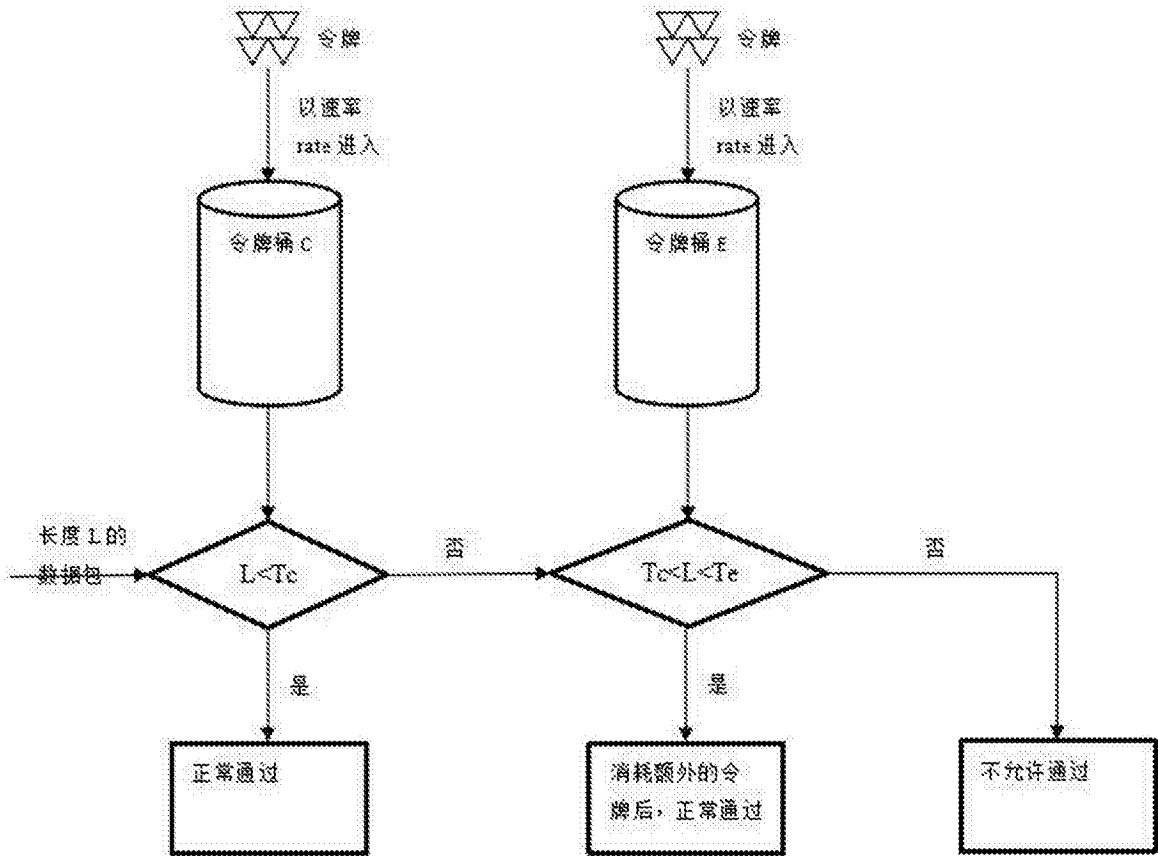


图2