

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 12/26 (2006.01)

H04L 12/56 (2006.01)



# [12] 发明专利申请公布说明书

[21] 申请号 200710125436.3

[43] 公开日 2009年1月21日

[11] 公开号 CN 101350740A

[22] 申请日 2007.12.21

[21] 申请号 200710125436.3

[71] 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为  
总部办公楼

[72] 发明人 张卫东 曹彦东 兑继英

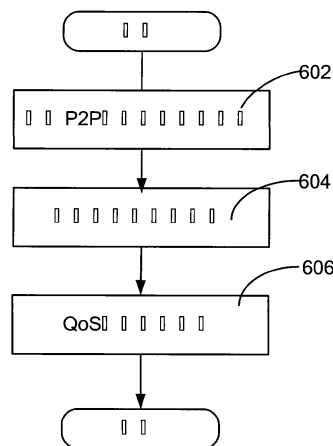
权利要求书 2 页 说明书 12 页 附图 5 页

## [54] 发明名称

一种导致 QoS 异常因素的检测方法和系统

## [57] 摘要

本发明提供一种导致 QoS 异常因素的检测方法包括如下步骤：接收服务器分片后的点对点 P2P 的数据码流；根据数据码流判断节点的数据流是否存在数据丢失信息；若存在数据丢失信息，根据域值参数确定导致 QoS 异常的因素。同时，本发明实施例还提供一种导致 QoS 异常因素的检测系统。本发明使得 QoS 异常信息能头准确定位到域，QoS 异常因素的也可以被准确检测，保证了 P2P 系统中 QoS 异常信息可以被准确检测，使得系统中的 QoS 能够真正得到保证。



1. 一种导致 QoS 异常因素的检测方法，其特征在于，该方法包括如下步骤：

接收服务器分片后的点对点 P2P 的数据码流；

根据所述数据码流判断节点的数据流是否存在数据丢失信息；

若存在数据丢失信息，根据域值参数确定导致 QoS 异常的因素。

2. 如权利要求 1 所述的导致 QoS 异常因素的检测方法，其特征在于，所述根据域值参数确定导致 QoS 异常的因素包括：根据域值参数确定下行带宽异常，和/或根据域值参数确定上行带宽异常。

3. 如权利要求 2 所述的导致 QoS 异常因素的检测方法，其特征在于，所述根据域值参数确定下行带宽异常包括：QoS 异常的父节点数目与父节点总数的比值超过域值时，判定为该父节点 QoS 异常。

4. 如权利要求 2 所述的导致 QoS 异常因素的检测方法，其特征在于，所述根据域值参数确定上行带宽异常包括：上报本节点 QoS 异常的子节点数目与子节点总数的比值超过域值时，判定为该子节点 QoS 异常。

5. 如权利要求 1 所述的导致 QoS 异常因素的检测方法，其特征在于，所述服务器分片后的点对点 P2P 的数据码流包括：数据分片的分片号和/或时间戳信息。

6. 如权利要求 5 所述的导致 QoS 异常因素的检测方法，其特征在于，该方法利用数据分片和时间戳信息来计算数据流的延时。

7. 如权利要求 5 所述的导致 QoS 异常因素的检测方法，其特征在于，该方法还包括：根据数据丢失信息和/或延时信息来判断父节点 QoS 是否异常。

8. 如权利要求 1 所述的导致 QoS 异常因素的检测方法，其特征在于，所述统计根据域值参数确定导致 QoS 异常的因素包括：上行带宽异常因素和下行带宽异常因素。

9. 如权利要求 8 所述的导致 QoS 异常因素的检测方法，其特征在于，该方法包括步骤：

计算父节点 QoS 异常因素来确定下行带宽是否异常，计算子节点 QoS 异常因素来确定上行带宽是否异常。

10.如权利要求 1 所述的导致 QoS 异常因素的检测方法，其特征在于，该方法包括：当确定 QoS 异常因素为非上行带宽异常和非下行带宽异常时，确定 QoS 异常因素为网络异常因素。

11.一种导致 QoS 异常因素的检测系统，其特征在于，该系统包括：  
接收模块，用于接收服务器分片后的点对点 P2P 的数据码流；  
数据丢失信息判断模块，用于根据所述数据码流判断节点的数据流是否存在数据丢失信息；

异常因素确定模块，用于根据域值参数确定导致 QoS 异常的因素。

12.如权利要求 11 所述的导致 QoS 异常因素的检测系统，其特征在于，该系统进一步包括：

分片存储模块，用于存储将所述服务器分片后的点对点 P2P 的数据码流。

13.如权利要求 12 所述的 QoS 异常检测系统，其特征在于，所述异常因素确定模块还包括：

QoS 指数计算单元，用于计算父节点的 QoS 指数并判断该 QoS 指数是否异常；

异常类型判断单元，用于判断 QoS 异常的类型，该类型包括上行带宽异常和下行带宽异常。

## 一种导致 QoS 异常因素的检测方法和系统

### 技术领域

本发明涉及 QoS (Quality of Service, 服务器质量) 异常检测技术, 尤其涉及一种 QoS 异常检测方法、系统及装置。

### 背景技术

随着科学的发展和技术的进步, 越来越多的媒体技术走进千家万户, 其中 P2P (Peer to Peer, 点对点) 技术是流媒体直播技术中应用的较为广泛的技术, 也成为业界研发的重点。

传统的流媒体直播系统中, 客户端的流媒体数据均由流媒体服务器 (Media Server, MS) 实时提供, 虽然骨干网络不断升级, 但随着用户 (Client) 的数量不断增加, 也给骨干网络带来冲击, 同时也加大了媒体服务器的负荷。因此, 媒体服务器的性能及 QoS 成为系统的瓶颈。为了解决媒体服务器性能和 QoS 的问题, P2P 的技术得到了广泛的应用, 其主要是通过网络中的节点 (Peer) 实时接收并上传媒体数据, 通过用户之前的相互协作, 充分利用边缘网络资源, 不再要求媒体服务器提供全部数据, 甚至不需要媒体服务器的任何数据, 就可以观看直播的节目。

P2P 直播流媒体技术虽然突破了传统直播系统的瓶颈, 但也引入了新的问题。在传统直播系统中, 所有的媒体数据都来自媒体服务器, 媒体服务器是由运营商提供, 相对稳定, 可提供可靠的服务。而在 P2P 直播系统中, 用户的媒体数据来源除了媒体服务器外, 还有部分来自其他用户。由于用户的行为不可预测、不可控制, 所以用户不能提供可靠的服务。

业界解决上述问题的方法是通过统计一定时间内发出的数据量来计算上行带宽, 即:  $\text{带宽} = \text{数据量} / \text{时间}$ , 由此得出节点提供服务的能力, 作为节

点服务质量的依据，从而进行 QoS 异常检测，通过 QoS 异常检测信息来保证 QoS。

发明人在实现本发明的过程中发现现有技术中至少存在如下问题：上行带宽是通过公式（带宽=数据量/时间）计算的，但是由于统计的时间是有限的，而且数据的传输不可能保证一直是一个稳定的带宽传输，所以计算结果的误差较大；并且该方案是通过计算节点发出的数据量来统计的，实际上节点发出的数据量与接收者接收的数据量还可能存在差异，导致该方法计算得到的带宽数据及其不稳定，系统的 QoS 并不能真正得到保证。

## 发明内容

有鉴于此，实有必要提供一种导致 QoS 异常因素的检测方法，来解决现有的 P2P 系统中 QoS 异常信息不能被准确检测的问题。

同时，提供一种导致 QoS 异常因素的检测系统，来解决现有的 P2P 系统中 QoS 异常信息不能被准确检测的问题。

一种导致 QoS 异常因素的检测方法，包括如下步骤：

接收服务器分片后的点对点 P2P 的数据码流；

根据所述数据码流判断节点的数据流是否存在数据丢失信息；

若存在数据丢失信息，根据域值参数确定导致 QoS 异常的因素。

一种导致 QoS 异常因素的检测系统，包括：

接收模块，用于接收服务器分片后的点对点 P2P 的数据码流；

数据丢失信息判断模块，用于根据所述数据码流判断节点的数据流是否存在数据丢失信息；

异常因素确定模块，用于根据域值参数确定导致 QoS 异常的因素。

与现有技术相比，本发明实施例通过接收服务器分片后的点对点 P2P 的数据码流；根据数据码流判断节点的数据流是否存在数据丢失信息；在节点的数据流存在数据丢失信息时，根据域值参数确定导致 QoS 异常的因素，使得 QoS 异常信息能准确定位到域，QoS 异常因素的也可以被准确检测，

保证了 P2P 系统中 QoS 异常信息可以被准确检测,使得系统中的 QoS 能够真正得到保证。

#### 附图说明

图 1 为本发明实施例节点拓扑网络结构示意图。

图 2 为本发明实施例 QoS 异常检测系统结构示意图。

图 3 为本发明实施例 P2P 直播码流分片示意图。

图 4 为本发明实施例第一个检测周期 T1 的记录示意图。

图 5 为本发明实施例第 n 个检测周期 Tn 的记录示意图。

图 6 为本发明实施例 QoS 异常检测方法整体流程示意图。

图 7 为本发明实施例 QoS 异常检测方法详细流程示意图。

#### 具体实施方式

请参看图 1,图 1 为本发明实施例节点拓扑网络结构示意图。本实施例以 P2P 直播系统为例进行说明,在 P2P 直播系统中,网络拓扑结构存在多种模型,但可就一个节点在拓扑中的位置,可以定义如下:节点 T (Peer T) 存在  $n$  ( $n \geq 1$ ) 个父节点为节点 T 提供数据流,同时给  $m$  ( $m \geq 1$ ) 个子节点提供数据流。

节点 T 的 QoS 异常种类包括:上行带宽异常、下行带宽异常和网络类异常,上行带宽异常也可以称为父节点异常,下行带宽异常也可以称为子节点异常。请结合参看图 2,本发明实施例 QoS 异常检测系统 200 包括:接收模块 202、分片存储模块 204、数据丢失信息判断模块 206、异常因素确定模块 208 和异常处理模块 222。

接收模块 202 用于接收服务器分片后的点对点 P2P 的数据码流,本实施例中以媒体服务器从直播源接收到一定的数据码流为例(如:1316B)。

分片存储模块 204,用于存储将服务器分片后的点对点 P2P 的数据码流。本实施例中,将数据分片结构请结合参看图 3 的数据分片,即:在该数据加

上一个头 head 信息，head 信息中包括分片编号（顺序编号，该编号与播放顺序保持一致），时间戳（取媒体服务器的当前系统时间）。由此，可以得到任意两个分片之间的时间间隔，假设直播源与媒体服务器之间的网络是可靠的，则该间隔可以视为两个分片之间的数据能维持的时间。例如，本实施例中从分片 100 到分片 126，可以播放的时间近似等于  $1285789-1278636=7153\text{ms}$ 。

数据丢失信息判断模块 206，用于根据数据码流判断节点的数据流是否存在数据丢失信息。异常因素确定模块 208，用于根据域值参数确定导致 QoS 异常的因素。本实施例中，异常因素确定模块 208 还包括：周期触发检测单元 210、父节点 QoS 指数计算单元 212、异常统计单元 214、异常类型判断单元 216、父节点异常上报单元 218 和子节点异常数据检测单元 220。

周期触发检测单元 210，用于检测时间周期并触发 QoS 检测。本实施例中，QoS 检测周期 T 触发，第一次检测时，记录父节点 F 发送过来的大分片 S1 及该分片的时间戳 T1，以后第 n ( $n \geq 2$ ) 个周期检测时，获取父节点 F 发送的最大分片 Sn 及该分片的时间戳 Tn。

父节点 QoS 指数计算单元 212，用于计算父节点 F 的 QoS 指数并判断该 QoS 指数是否异常。本实施例中，以下两种情况视为异常：

A) 假设数据正常时，S(n-1) 到 Sn 分片需要有 An 个来自父节点 F，实际从 S(n-1) 分片到 Sn 分片有 Bn 个来自父节点 F，如果 S(n-1) 到 Sn 之间来自父节点 F 的数据分片丢失率达到域值，视为异常，即：

$$(A_n - B_n) / B_n > L$$

B) 如果数据正常，在 n 个周期应该接收到可以播放  $(n-1) * T$  的分片，如果数据流延时达到域值，视为异常，即：

$$(n-1) * T - (T_n - T_1) > D$$

其中，本实施例中域值参数包括丢失率域值 L 和延时域值 D。

L；数据丢失率域值，即：在一段时间内应该接收到某个父节点 w 个数据分片，子节点实际接收 X 个，丢失  $(L - X) / W$ ，当该值大于 L 时，

数据流异常;

D; 时间延时域值, 即: 在一段时间 R 内, 子节点接收到的数据分片可以播放的时间为 V, 时间延时为 R - V, 当该值大于 D 时, 数据流异常。

检测 A) 时也可以采用如下方法:

假设数据正常时, S1 到 Sn 分片需要有 An 个来自父节点 F, 实际从 S1 分片到 Sn 分片有 Bn 个来自父节点 F, 如果 S1 到 Sn 之间来自父节点 F 的数据分片丢失率达到域值, 视为异常, 即:

$$(A_n - B_n) / B_n > L$$

以下以具体应用实例来说明: 设置检测周期 T = 6000ms, 数据丢失率域值为 L = 20%, 时间延时域值 D = 8000ms;

假设节点 F 为节点 T 的父节点, 提供分片号为被 5 整除的分片, 第一个检测周期, 数据请参看图 4(白色代表还未接收, 其他灰色方格代表已经接收)。第一个检测周期记录 S1=75, T1=122432; 第 n=5 个周期检测记录 Sn = 995, Tn = 144978;

### 1. 计算分片丢失率

在 n = 5 的周期内应该接收来自节点 F 的分片数 A5 = (995 - 775) / 5 + 1 = 23, 实际接收 B5 = 21, 分片丢失率 = (23 - 21) / 23 = 8.7% < 20%, 分片丢失率正常。

### 2. 计算时间延时

在 n=5 的检测时, 接收的最大分片的时间戳 T5 = 144978, 第一个周期记录最大分片的时间戳 T1 = 122432, 当前周期时间延时 = (n-1)\*T - (T5-T1) = (5-1)\*6000 - (144978-122432) = 1454ms < 8000ms, 时延正常。

因此, 父节点 F 在第 n=5 个周期 QoS 检测正常。

异常统计单元 214, 用于统计 QoS 异常节点数目。本实施例中, 根据计算出来的父节点 QoS 异常数目进行统计。

异常类型判断单元 216, 用于判断 QoS 异常的类型, 本实施例中 QoS 异常的类型包括: 上行带宽异常、下行带宽异常和网络异常等。



子节点下行能力异常的判定:

QOS 周期 T 触发, 假设在该周期检测中, 节点有 n 个父节点, 通过单个父节点 QOS 检测策略可以检测出与 y 个父节点之间的数据传输异常, 如果  $y/n > Y$ , 则节点的下行能力不足;

域值参数:

Y; 下行带宽不足域值, 即: 假设节点有 n 个父节点, 检测发现与 y 个父节点之间的数据传输异常, 当  $y/n > Y$  时, 则该节点的下行带宽不足。

子节点判断并非自身下行能力不足时, 则向父节点发送接收数据异常通知。

父节点上行能力异常的判定:

QOS 周期 T 触发, 假设在该周期检测中, 节点有 k 个子节点, 其中在该周期内有 u 个子节点上报本节点数据传输异常, 如果  $u/m > U$ , 则节点的上行能力不足;

域值参数:

U; 上行带宽不足域值, 即: 假设节点有 m 个子节点, 检测发现与 u 个子节点之间的数据传输异常, 当  $u/m > U$  时, 则该节点的上行带宽不足。

网络异常的判定:

当 QOS 检测存在某些数据流异常, 通过子节点下行能力异常判断并非子节点因素所致, 通知相应的父节点, 而父节点通过上行能力异常判断也并非父节点因素导致, 该情况视为网络异常。

父节点异常上报单元 218, 用于上报当前节点 QoS 异常, 本实施例中, 该异常为非下行异常, 该异常可以包括: 上行异常和网络异常等。

子节点异常数据检测单元 220, 用于检测当前周期接收子节点上报 QoS 异常的数目。本实施例中, 计算子节点 QoS 异常数目与计算父节点 QoS 异常数目类似, 不再赘述。

本发明实施例通过接收服务器分片后的点对点 P2P 的数据码流; 根据数据码流判断节点的数据流是否存在数据丢失信息; 在节点的数据流存在数据

丢失信息时，根据域值参数确定导致 QoS 异常的因素，使得 QoS 异常信息能头准确定位到域，QoS 异常因素的也可以被准确检测，保证了 P2P 系统中 QoS 异常信息可以被准确检测，使得系统中的 QoS 能够真正得到保证。通过时间戳机制屏蔽了码率的波动因素，并且通过时间戳机制引进了时延异常，可以及时检测出数据流的异常，并综合考虑各种导致数据流异常的因素，通过统计原理来判断，比较 QoS 异常的场景，有利于拓扑结构的优化调整，防止对 QoS 异常的不准确判定。

请结合参看图 6，本发明实施例 QoS 异常检测方法包括如下步骤：

步骤 602，接收服务器分片后的点对点 P2P 的数据码流，本实施例中以媒体服务器从直播源接收到一定的数据码流为例（如：1316B）。本实施例中，将数据分片结构请结合参看图 3 的数据分片，即：在该数据加上一个头 head 信息，head 信息中包括分片编号（顺序编号，该编号与播放顺序保持一致），时间戳（取媒体服务器的当前系统时间）。由此，可以得到任意两个分片之间的时间间隔，假设直播源与媒体服务器之间的网络是可靠的，则该间隔可以视为两个分片之间的数据能维持的时间。例如，本实施例中从分片 100 到分片 126，可以播放的时间近似等于  $1285789-1278636=7153\text{ms}$ 。

步骤 604，根据所述数据码流判断节点的数据流是否存在数据丢失信息。

步骤 606，根据域值参数确定导致 QoS 异常的因素。本实施例中，以下两种情况视为异常：

A)假设数据正常时， $S(n-1)$  到  $S_n$  分片需要有  $A_n$  个来自父节点 F，实际从  $S(n-1)$  分片到  $S_n$  分片有  $B_n$  个来自父节点 F，如果  $S(n-1)$  到  $S_n$  之间来自父节点 F 的数据分片丢失率达到域值，视为异常，即：

$$(A_n - B_n) / B_n > L$$

B)如果数据正常，在  $n$  个周期应该接收到可以播放  $(n-1)*T$  的分片，如果数据流延时达到域值，视为异常，即：

$$(n-1) * T - (T_n - T_1) > D$$

其中，本实施例中域值参数包括丢失率域值  $L$  和延时域值  $D$ 。

L; 数据丢失率域值, 即: 在一段时间内应该接收到某个父节点  $w$  个数据分片, 子节点实际接收  $X$  个, 丢失  $(L - X)/W$ , 当该值大于  $L$  时, 数据流异常;

D; 时间延时域值, 即: 在一段时间  $R$  内, 子节点接收到的数据分片可以播放的时间为  $V$ , 时间延时为  $R - V$ , 当该值大于  $D$  时, 数据流异常。

检测 A) 时也可以采用如下方法:

假设数据正常时,  $S1$  到  $S_n$  分片需要有  $A_n$  个来自父节点  $F$ , 实际从  $S1$  分片到  $S_n$  分片有  $B_n$  个来自父节点  $F$ , 如果  $S1$  到  $S_n$  之间来自父节点  $F$  的数据分片丢失率达到域值, 视为异常, 即:

$$(A_n - B_n) / B_n > L$$

请结合参看图 7, 本发明实施例 QoS 异常检测方法, 包括如下步骤:

步骤 702, 接收服务器分片后的点对点 P2P 的数据码流, 本实施例中以媒体服务器从直播源接收到一定的数据码流为例 (如: 1316B)。本实施例中, 将数据分片结构请结合参看图 3 的数据分片, 即: 在该数据加上一个头 head 信息, head 信息中包括分片编号 (顺序编号, 该编号与播放顺序保持一致), 时间戳 (取媒体服务器的当前系统时间)。由此, 可以得到任意两个分片之间的时间间隔, 假设直播源与媒体服务器之间的网络是可靠的, 则该间隔可以视为两个分片之间的数据能维持的时间。例如, 本实施例中从分片 100 到分片 126, 可以播放的时间近似等于  $1285789 - 1278636 = 7153\text{ms}$ 。

步骤 704, 检测时间周期并触发 QoS 检测。本实施例中, QoS 检测周期  $T$  触发, 第一次检测时, 记录父节点  $F$  发送过来的大分片  $S1$  及该分片的时间戳  $T1$ , 以后第  $n$  ( $n \geq 2$ ) 个周期检测时, 获取父节点  $F$  发送的最大分片  $S_n$  及该分片的时间戳  $T_n$ 。

步骤 706, 根据所述数据码流判断节点的数据流是否存在数据丢失信息, 计算父节点  $F$  的 QoS 指数并判断该 QoS 指数是否异常。本实施例中, 以下两种情况视为异常:

A) 假设数据正常时,  $S(n-1)$  到  $S_n$  分片需要有  $A_n$  个来自父节点  $F$ , 实

际从  $S(n-1)$  分片到  $S_n$  分片有  $B_n$  个来自父节点 F, 如果  $S(n-1)$  到  $S_n$  之间来自父节点 F 的数据分片丢失率达到域值, 视为异常, 即:

$$(A_n - B_n) / B_n > L$$

B) 如果数据正常, 在  $n$  个周期应该接收到可以播放  $(n-1)*T$  的分片, 如果数据流延时达到域值, 视为异常, 即:

$$(n-1) * T - (T_n - T_1) > D$$

其中, 本实施例中域值参数包括丢失率域值  $L$  和延时域值  $D$ 。

$L$ ; 数据丢失率域值, 即: 在一段时间内应该接收到某个父节点  $w$  个数据分片, 子节点实际接收  $X$  个, 丢失  $(L - X) / W$ , 当该值大于  $L$  时, 数据流异常;

$D$ ; 时间延时域值, 即: 在一段时间  $R$  内, 子节点接收到的数据分片可以播放的时间为  $V$ , 时间延时为  $R - V$ , 当该值大于  $D$  时, 数据流异常。

检测 A) 时也可以采用如下方法:

假设数据正常时,  $S_1$  到  $S_n$  分片需要有  $A_n$  个来自父节点 F, 实际从  $S_1$  分片到  $S_n$  分片有  $B_n$  个来自父节点 F, 如果  $S_1$  到  $S_n$  之间来自父节点 F 的数据分片丢失率达到域值, 视为异常, 即:

$$(A_n - B_n) / B_n > L$$

以下以具体应用实例来说明: 设置检测周期  $T = 6000\text{ms}$ , 数据丢失率域值为  $L = 20\%$ , 时间延时域值  $D = 8000\text{ms}$ ;

假设节点 F 为节点 T 的父节点, 提供分片号为被 5 整除的分片, 第一个检测周期, 数据请参看图 4(白色代表还未接收, 其他灰色方格代表已经接收)。第一个检测周期记录  $S_1=75$ ,  $T_1=122432$ ; 第  $n=5$  个周期检测记录  $S_n = 995$ ,  $T_n = 144978$ ;

### 1. 计算分片丢失率

在  $n=5$  的周期内应该接收来自节点 F 的分片数  $A_5 = (995 - 775) / 5 + 1 = 23$ , 实际接收  $B_5 = 21$ , 分片丢失率  $= (23 - 21) / 23 = 8.7\% < 20\%$ , 分片丢失率正常。

## 2. 计算时间延时

在  $n=5$  的检测时, 接收的最大分片的时间戳  $T5 = 144978$ , 第一个周期记录最大分片的时间戳  $T1 = 122432$ , 当前周期时间延时 =  $(n-1)*T - (T5-T1) = (5-1)*6000 - (144978-122432) = 1454\text{ms} < 8000\text{ms}$ , 时延正常。

因此, 父节点 F 在第  $n=5$  个周期 QoS 检测正常。

步骤 708, 统计 QoS 异常节点数目。本实施例中, 根据计算出来的父节点 QoS 异常数目进行统计。

步骤 710, 判断 QoS 异常的类型是否属于下行异常, 若属于下行异常, 执行步骤 712, 若不属于下行异常执行步骤 714。本实施例中, 子节点下行能力异常的判定:

QoS 周期 T 触发, 假设在该周期检测中, 节点有  $n$  个父节点, 通过单个父节点 QoS 检测策略可以检测出与  $y$  个父节点之间的数据传输异常, 如果  $y/n > Y$ , 则节点的下行能力不足;

域值参数:

$Y$ ; 下行带宽不足域值, 即: 假设节点有  $n$  个父节点, 检测发现与  $y$  个父节点之间的数据传输异常, 当  $y/n > Y$  时, 则该节点的下行带宽不足。

子节点判断并非自身下行能力不足时, 则向父节点发送接收数据异常通知。

步骤 712, 针对下行异常进行处理, 执行完步骤 716 后, 结束本流程。

步骤 714, 上报当前节点 QoS 异常, 本实施例中, 该异常为非下行异常, 该异常可以包括: 上行异常和网络异常等。

步骤 716, 检测当前周期接收子节点上报 QoS 异常的数目, 本实施例中, 计算子节点 QoS 异常数目与计算父节点 QoS 异常数目类似, 不再赘述。

步骤 718, 判断是否是上行异常, 若是上行异常执行步骤 720, 若不是上行异常, 执行步骤 722。父节点上行能力异常的判定:

QoS 周期 T 触发, 假设在该周期检测中, 节点有  $k$  个子节点, 其中在该周期内有  $u$  个子节点上报本节点数据传输异常, 如果  $u/m > U$ , 则节点

的上行能力不足;

域值参数:

U; 上行带宽不足域值, 即: 假设节点有  $m$  个子节点, 检测发现与  $u$  个子节点之间的数据传输异常, 当  $u/m > U$  时, 则该节点的上行带宽不足。

网络异常的判定:

当 QoS 检测存在某些数据流异常, 通过子节点下行能力异常判断并非子节点因素所致, 通知相应的父节点, 而父节点通过上行能力异常判断也并非父节点因素导致, 该情况视为网络异常。

步骤 720, 执行上行异常处理, 执行完步骤 720 后, 结束本流程。

步骤 722, 执行网络异常处理, 执行完步骤 722 后, 结束本流程。

本发明实施例通过接收服务器分片后的点对点 P2P 的数据码流; 根据数据码流判断节点的数据流是否存在数据丢失信息; 在节点的数据流存在数据丢失信息时, 根据域值参数确定导致 QoS 异常的因素, 使得 QoS 异常信息能准确定位到域, QoS 异常因素的也可以被准确检测, 保证了 P2P 系统中 QoS 异常信息可以被准确检测, 使得系统中的 QoS 能够真正得到保证。通过时间戳机制屏蔽了码率的波动因素, 并且通过时间戳机制引进了时延异常, 可以及时检测出数据流的异常, 并综合考虑各种导致数据流异常的因素, 通过统计原理来判断, 比较 QoS 异常的场景, 有利于拓扑结构的优化调整, 防止对 QoS 异常的不准确判定。

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程序来指令相关的硬件来完成, 所属的程序可以存储于一计算机可读存储介质中, 该程序在执行时, 包括如下全部或部分步骤:

接收服务器分片后的点对点 P2P 的数据码流;

根据数据码流判断节点的数据流是否存在数据丢失信息;

若存在数据丢失信息, 根据域值参数确定导致 QoS 异常的因素。

所述的存储介质, 如: ROM/RAM、磁盘、光盘等。

综上所述, 以上仅为本发明的较佳实施例而已, 并非用于限定本发明的

---

保护范围。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

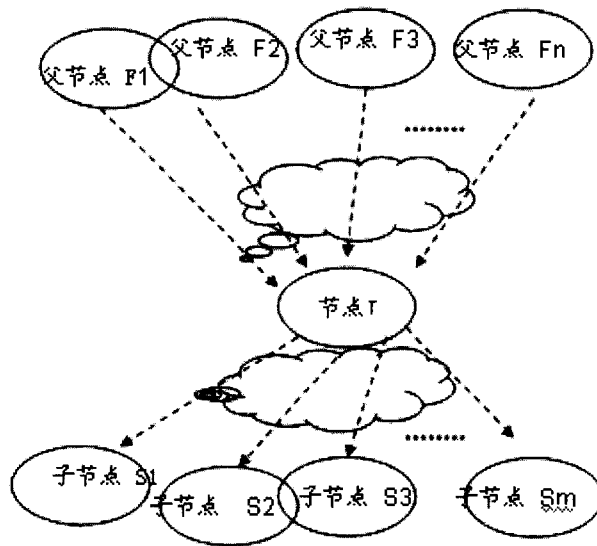


图 1



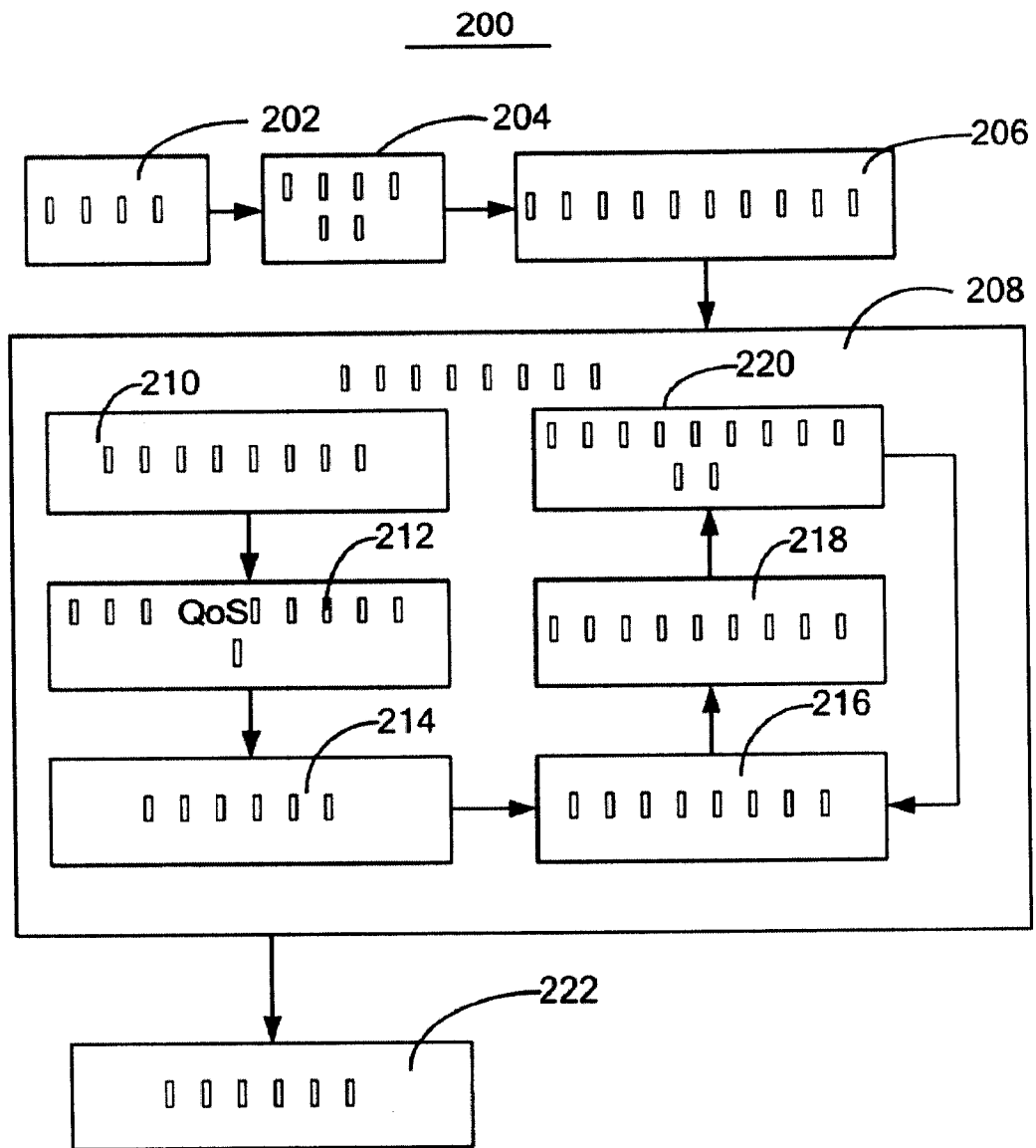


图 2

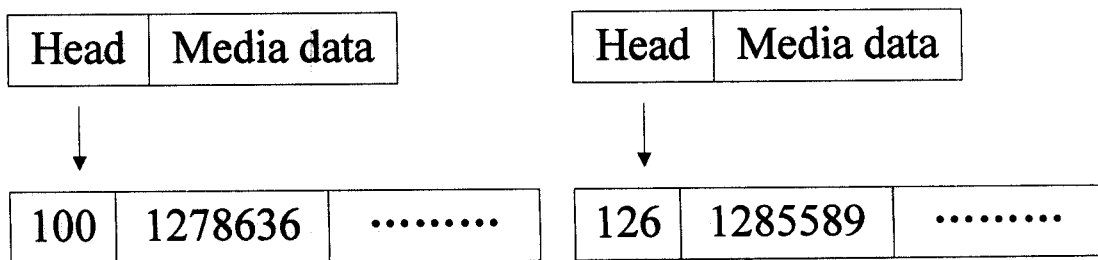


图 3

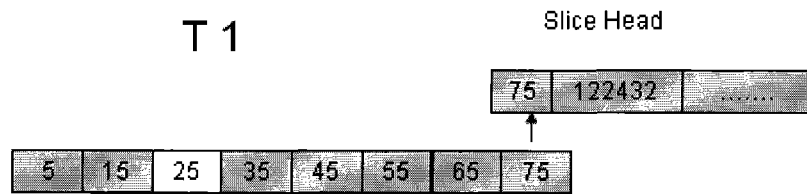


图 4

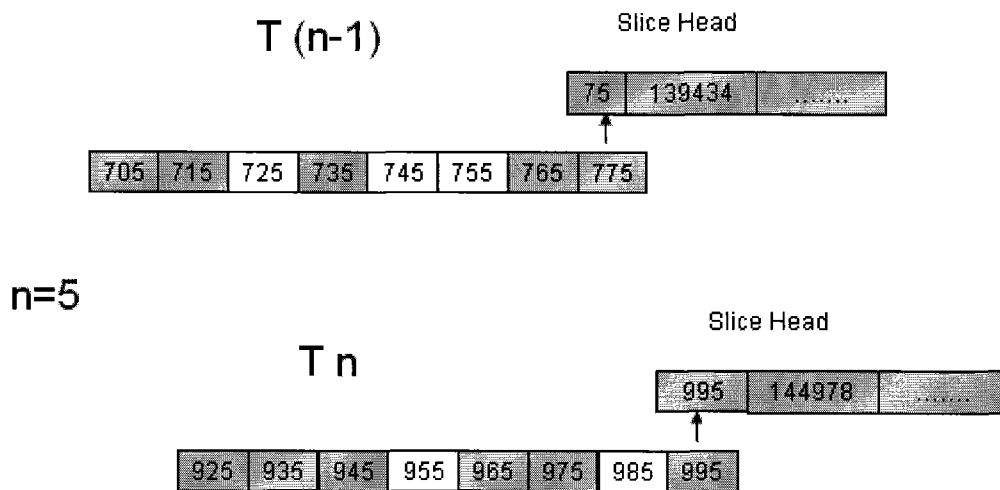


图 5

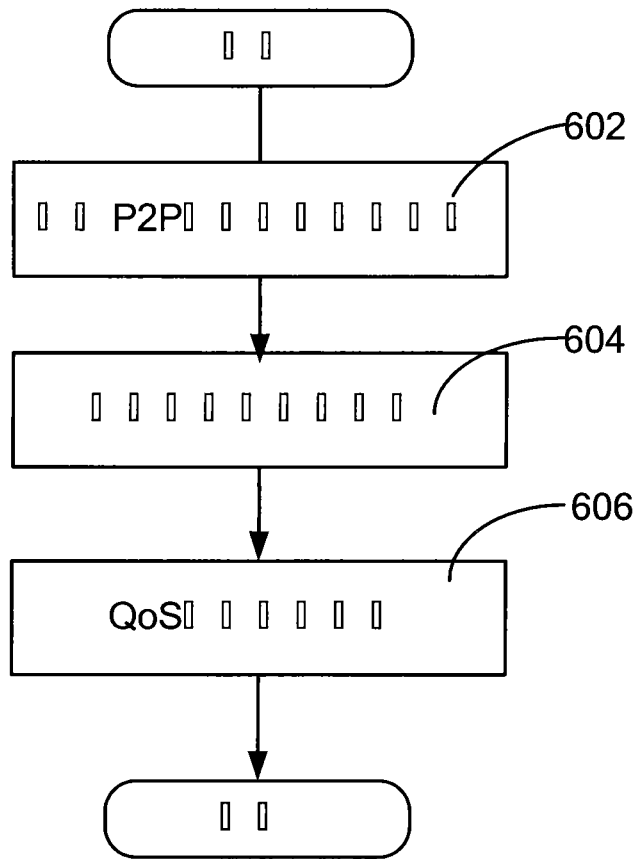


图 6

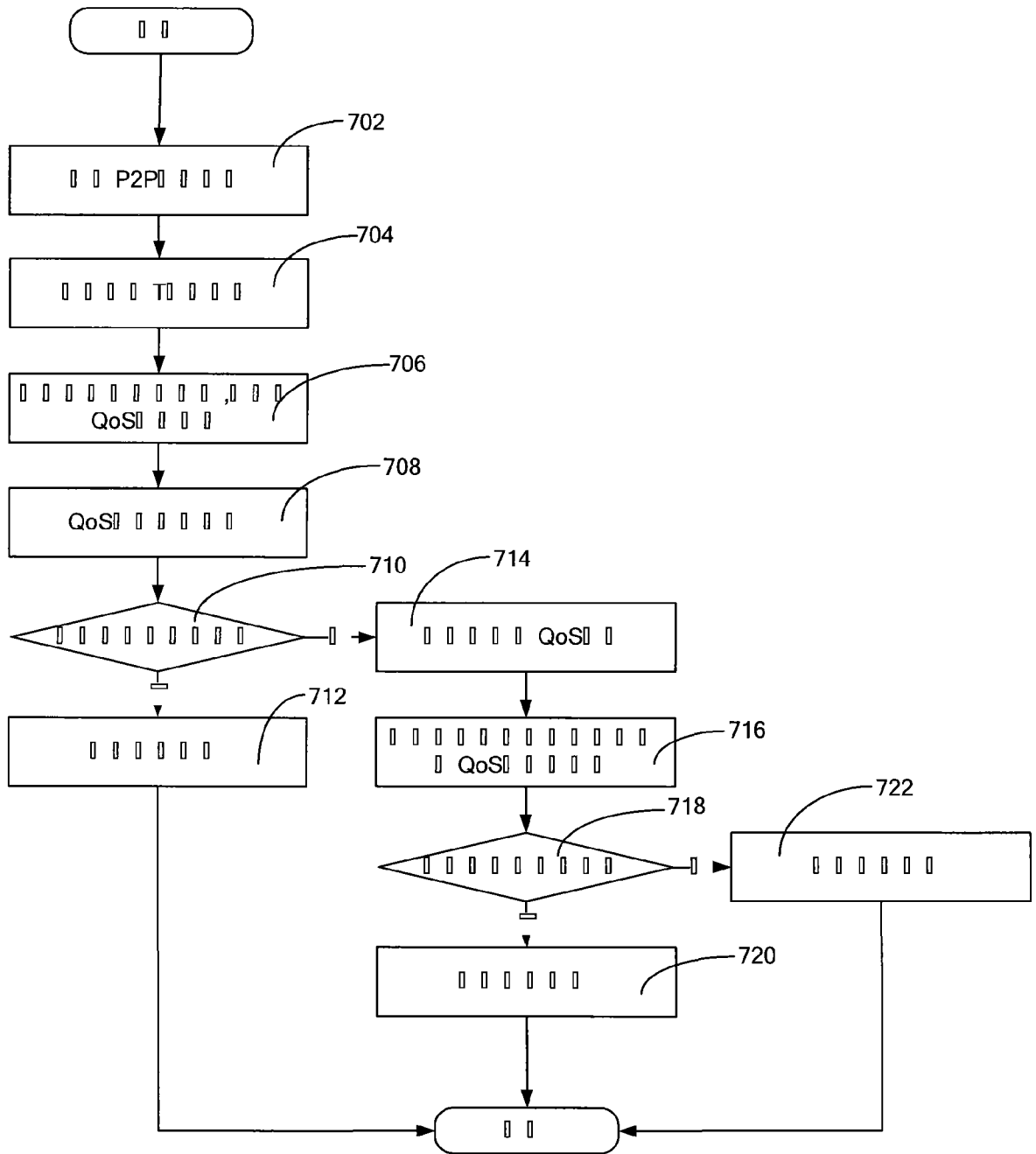


图 7