IP网络接纳控制研究评述*)

高文宇　陈松乔　王建新
(中南大学信息科学与工程学院　长沙410083)

摘　要　接纳控制是实现IP网络QoS最重要的手段之一，长期以来一直受到广大研究者的关注，基于不同理论的各种接纳控制机制不断出现。这些接纳控制机制在一定程度上都能满足QoS的需求，但也不同程度地在诸如可伸缩性、健壮性等方面存在一些问题。本文通过对其中多个接纳控制机制的实现原理的分析，考察了这些接纳控制机制的优点和存在的问题，并给出了一些相应的解决办法，这些也是有待进一步研究的问题。

关键词　接纳控制，QoS，可伸缩性，健壮性

The Research of Admission Control in IP Networks

GAO Wen-Yu　CHEN Song-Qiao　WANG Jian-Xin
(College of Information Science and Engineering, Central South University, Changsha 410083)

Abstract　Admission Control is one of the most important components of the implementation of IP networks' QoS. And it has been received lots of attention for a long time. Many admission control schemes have been developed, and all of them can meet part of the requirement of QoS. But they also have such demerits as non-scalability, non-robust etc. In this paper, the fundament of a few admission control schemes is analyzed, then their merits and demerits are analyzed, moreover, some solutions are given, and which will be the topics of futural research.

Keywords　Admission control, QoS, Scalability, Robust

1　引言

随着Internet的迅速发展以及网上实时业务的不断增多，现有IP网络只能提供单一的"尽力型"(best effort)服务，远远不能满足多数实时业务的服务质量要求(Quality of Services)。另外，来自不同的应用和不同的用户对Internet提供的服务也提出了业务区分的要求，他们希望自己的网络连接请求能有一定的质量保证(如带宽、端到端的延时率、丢包率等)或较高的优先级，当然这种请求的满足需要与一定的网络政策(policy)结合起来。例如，更高的质量保证意味着更高的服务费用。在这种情况下，网络的QoS研究及实施就应运而生了。

接纳控制作为一种预防性的流量控制手段是实现网络QoS保障的重要手段，实施连接接纳控制的网络，要求用户在请求接入网络时将自己的通信量传输特征和要求的服务质量告诉网络，网络根据用户的通信量特征和网络现存的资源情况，决定是否接纳用户的连接请求。

本文作为国家自然科学基金重大研究计划项目"基于预测的可扩展接纳控制研究"的一部分，拟通过对现有的接纳控制机制研究和比较，归纳其优、缺点，为进一步的研究提供基础性的支持。文章第2部分对IETF给出的两种服务模型IntServ和DiffServ做了一个简要介绍，这是进行接纳控制研究的背景材料，第3部分是对基于测量的接纳控制机制的分析，第4部分是对基于端到端的探测接纳控制机制的分析，第5部分是基于带宽代理的接纳控制机制的分析，第6部分是对一种基于令牌的接纳控制机制的分析，最后是总结。

2　IntServ和DiffServ

IntServ[1]和DiffServ[2,3]是由IETF分别于1994和1998年提出的两种用于解决Internet的QoS问题的服务模型。相关工作还在不断继续，特别是针对DiffServ服务模型，涉及的许多问题还是开放的问题，不断有新的RFC形成。

2.1　IntServ

IntServ/RSVP服务模型在IETF RFC1633中进行了定义。RFC1633将资源预留协议RSVP作为IntServ结构中的主要信令协议。其基本思想就在于以资源预留的方式来实现QoS保障，IntServ/RSVP提供了3种级别的服务：

·端到端的质量保证型服务(Guaranteed Service)：保证带宽、限制延迟、无丢包。

·可控负载型服务(Controlled-Load Service)：类似于当前的一个负载较轻网络中实现的尽力而为业务的服务质量。

·尽力型服务(Best Effort Service)：类似当前Internet提供的尽力而为的服务。

从理论上讲IntServ/RSVP模型完全可以保证为IP网络提供QoS保障。但随后在一些网上的实验表明这种服务模型有一个致命的缺点，那就是可扩展性差。由于需在每条路径经过的所有路由器上保存状态信息，因此，随着网络中被接纳的流的增多，会给路由器带来沉重的负担，最终导致服务的不可用。因此，在一个运营商规模的网络中使用IntServ来实现QoS几乎是不可能的。

由于IntServ/RSVP体系存在着诸多问题，一种新的体系结构便应运而生，这就是区分服务体系结构DiffServ。

*)基金项目：国家自然科学基金重大研究计划(90304010)。高文宇　博士研究生，主要研究方向为计算机网络。陈松乔　教授，博士生导师，主要研究方向为计算机网络、软件工程。王建新　博士，副教授，主要研究方向为计算机网络。

2.2 DiffServ

DiffServ 服务模型最早是在 IETF RFC2475中进行定义的。DiffServ 的最大特点就是简单有效、扩展性强。其实施特点是采用聚合的机制将具有相同特性的若干通信流聚合起来，为整个聚合流提供服务，而不再面向单个通信流。也就是说在 DiffServ 网络边界路由器上保持每流状态，核心路由器只负责数据包的转发而不保持状态信息。这种 Core-Stateless 结构有很强的扩展性。其基本实现方法是：

简化网络内部节点的服务机制。在网络内部的核心路由器中只保存简单的 DSCP(DiffServ CodePoint)与 PHB(Per-Hop Behavior)的对应机制，在数据流进入核心路由器时只根据数据包头部 DS(Differentiated Services)域中的 DSCP 进行转发，而业务流状态信息的保存与流监控机制的实现等都在网络边界节点进行，内部节点是状态无关的。

聚合网络内部核心路由器的服务对象。采用流聚集的方式进行传输控制，具有相同 DSCP 的业务流组成一个宏流(macro-flow)，核心路由器的服务对象即是宏流而不是单流(micro-flow)，单流信息只在网络边界节点保存和处理。

与 IntServ 类似，Diff-Serv 也定义了三种服务类型：

·奖赏服务(Premium Services，PS)：为用户提供低延迟、低抖动、低丢包率和保证带宽的端到端或者网络边界到边界的传输服务。

·确保服务(Assured Services，AS)：确保服务是从统计上保证用户的带宽，其初衷是在网络拥塞的情况下，也能保证用户有一定量的预约带宽。AS 的着眼点是带宽和丢包率，而不太注重延迟和抖动。

·尽力型服务：类似于目前 Internet 上尽力而为的服务。

IETF 定义的 IntServ 和 DiffServ 服务模型为实现 QoS 接纳控制提供了一个原则性的框架和一些基础性的支持，下面我们将对四种接纳控制机制进行详细的分析讨论。

3 基于测量的接纳控制机制

3.1 基于测量的接纳控制的基本原理

一般来说，基于测量的接纳控制主要包括两个组成部分，一是测量单元，另一部分是接纳控制决策单元。测量单元根据不同的测量策略动态测量网络运行状态并将测量得到的信息提供给接纳控制决策单元；接纳控制决策单元则根据来自测量单元的测量信息以及预先定义好的一些接纳控制准则做出接纳控制决策。

下面就是文[4～6]中提到的几种典型的测量策略和接纳控制准则。

3.2 测量策略

为了做出正确的决策，接纳控制模块必须对当前的网络拥塞和网络中被使用的资源总量有一个精确的测量。其中一种广为应用的测量策略就是时间窗口法(Time Window)。

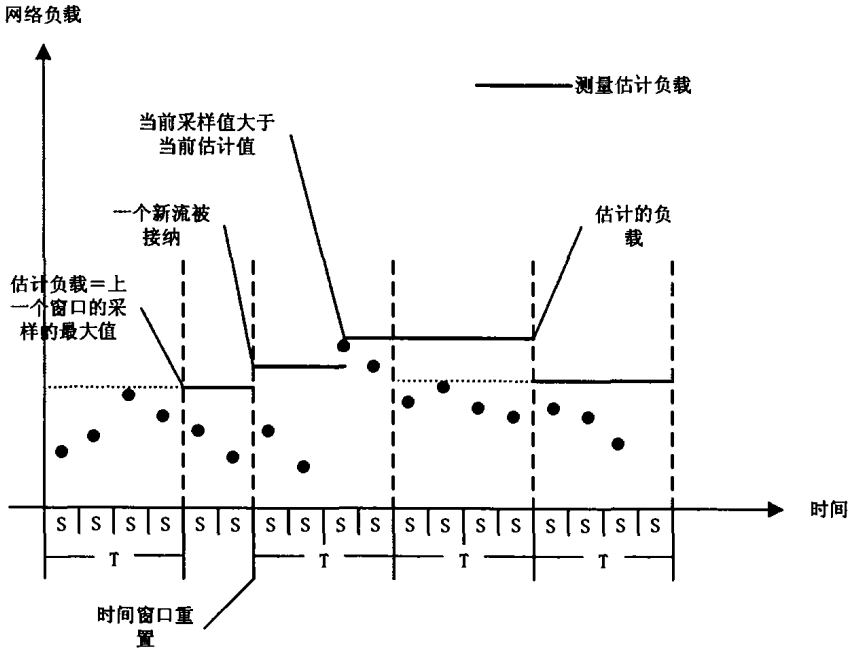

图1 时间窗口测量策略

时间窗口测量策略简单地说就是每隔一段时间 S 对网络负载进行一次测量，然后每经过一个时间窗口 $T(T=n*S)$对估计的网络负载进行一次更新。

图1是一个“测量窗口”测量策略的示意图。图中，每隔时间 S 对网络负载进行一次采样，并将结果保存起来。经过一个时间窗口 $T(T=n*S)$后，估计的负载值用上一个窗口中的最大测量负载来替代。另外，每当一个新流被接纳时，估计值根据新流带来的增加量而增加，并且时间窗口重置(重新计时)。还有，只要当前采样值大于当前估计值，则估计值上升到当前采样值。

在时间窗口测量策略中，参数 S 和 T 是两个非常敏感的参数。较小的 S 意味着更频繁的采样频率，这将得到一个较高的负载估计值，从而使得接纳控制机制趋于保守；相反增大 S 值会使接纳控制机制能接纳更多的流量请求。同理，较小的 T 值意味着更频繁地更新测量估计值，使得测量策略更快地“忘记”网络的突发状态，而较大的 T 值意味着测量策略“记忆”一个较长的测量历史记录。

鉴于 S 和 T 的选择对接纳控制决策影响的敏感性，有的研究者提出，根据对网络负载的测量结果动态地决定 S 和 T 的大小，从而在保障接纳控制正确性的前提下提高网络的利用率。

其它的测量还有如点采样(Point Samples)和指数平滑法(Exponential Averaging)等。

3.3 接纳准则

接纳准则就是接纳控制机制接纳或拒绝一个通信流所采用的原则。

接纳控制机制如果决定接纳一个新流，不仅要保证能满足新流提出的QoS需求，还要保证由于新流的加入不会影响到原已接受的通信流的QoS。接纳准则就是提供这样一些保证的条件。以下就是两个经常使用的接纳准则。

3.3.1 速率和(Rate Sum) 接纳准则可简单表示为：

$v+r\leqslant\alpha*u \quad (0<\alpha<1)$

式中 v 是当前已经被使用(分配)的资源，r 是提出连接请求的新流的速率，u 是所有可用的网络带宽，α 表示一个对网络带宽资源的目标利用率，α 越大表示期望对网络带宽的利用率越大(即允许接纳更多的流)，当然也带来更多的网络拥塞风险。因为基于测量的机制得到的对网络负载的估计值是一个非精确值，所以引入 α 以对网络资源保持一个余量，防止网络拥塞。

3.3.2 等效容量(Equivalent Capacity) 等效容量 $C(\varepsilon)$ 是某一类通信流的速率累加(聚集流)的估计值，并假设实际的速率会以概率 ε 超过该估计值 $C(\varepsilon)$。接纳控制决策就基于 $C(\varepsilon)$、新流的峰值速率 P 以及分配给这一类流的带宽容量 C。当一个新流满足以下条件时将被接纳：$C(\varepsilon)+P\leqslant C$。其中主要有两种计算等效容量的方案：

第一种方案是由 Guerin 等人提出的[7]，假定聚集流的速率能够用正态分布 (u,σ^2) 来建模，则 $C(\varepsilon)$ 可以表示为：

$$C(\mu,\sigma^2,\varepsilon)=\mu+\sqrt{2\ln\frac{1}{\varepsilon}+\ln\frac{1}{2\pi}}\sigma$$

聚集流的速率的均值 u 和方差 σ^2 可以使用前面提到的测量策略通过测量得到。该模型适用于计算多个类似流(similar flow)聚集后的等效容量。然而，在实时的通信流中每个单流会表现出各自不同的差异。因此，这种机制在某些情况下会低估等效容量。

第二种方案是由 Floyd 在 Hoeffding 的基础上提出的[8]，在该方案中，计算等效容量时并没有假设聚集流服从正态分布，因此这种方案对于来自同质源的小流是有效的。只要得到给定 n 个源的峰值速率 $\{P_i\}_{1\leqslant i\leqslant n}$，则等效容量可由下式估计出来：

$$C(\mu,\{P_i\}_{1\leqslant i\leqslant n},\varepsilon)=\mu+\sqrt{\frac{\ln(1/\varepsilon)\sum_{i=1}^{n}(P_i)^2}{2}}$$

式中的平均速率 u 可以通过测量得出，峰值速率可以从通信源的描述中得出。

相对而言，两种计算等效容量的方法，前者与聚集流的到达速度捆绑得更紧密，但是，当聚集流中得单流数量很少时，前者会低估实际的等效容量。当聚集流中单流增多时，这两种方案都会因为更大的统计复用而变得更精确。

估计误差 ε 在等效容量的计算中起着一个非常重要的作用。更大的 ε 意味着接纳控制算法变得更“贪婪”，即估计的等效容量会变得更小，于是更多的流将被接纳(别的参数保持不变)。因此，建议选择 ε 的最好方法是通过运行经验或实验来确定 ε。

4 基于端到端探测的接纳控制机制

4.1 基本原理

基于端到端探测的接纳控制机制最本质的特征就是依靠端到端的探测包的发送来决定目前的网络中是否有足够的资源来满足一个新的通信流的QoS请求[9~12]。

一个连接过程主要由两部分组成：一是探测阶段，二是数据传输阶段(若探测结果显示该流可以被接纳)。一个通信源如果想建立一个连接，则它首先发送一些探测包(探测包的特征，如平均速率、突发率等应与实际的数据包特征一致)给目的端，这些探测包的优先级较已经被接纳的实际的数据包的优先级要低(也就是说，要求路由器维持两个不同的包队列，一个是高优先级的实际数据包的队列，一个是低优先级的探测包的队列，这样可以保证探测包的传输不至于影响已经被接纳的流的QoS)，目的端在接收到第一个探测包时，启动对探测包的测量和统计分析(如探测包的到达速率，延时、抖动、丢包率等)。

在一定的测量过程完成后，目的端(接收者)根据测量得到的关于探测包的各种统计特征以及发送端要求的QoS评估在探测包流经的传输路径上是否有足够的资源来满足发送端的QoS要求，并将此评估结果发回给发送端，最后由发送端决定是否发送数据包。当然，还应当设置一个有效期限，在此有效期限内没有收到目的端的返回信息，则退出(取消)这次连接请求，以后再试。

4.2 评价

在基于端到端的探测的接纳控制机制中，核心路由器是无状态的(core stateless)，这与 DiffServ 的要求是一致的，唯一要求路由器做的就是区分探测包和实际数据包的优先级，以避免探测包流与实际的数据包流争夺网络资源，从而影响已经被接纳的实际数据包流的QoS。因此，基于端到端的探测的接纳控制机制非常简单、易行。在任何规模和范围的网络中都可以轻易地实现，并且不会因为网络拓扑的变化而导致接纳控制不可行。

但是这种接纳控制机制中得到的探测结果是一个比较粗糙的值，并不能精确地反映网络的运行状态。另外，探测时间的长短也会对接纳控制的准确性产生影响。还有，在IP网络中，包的传输路径是动态变化的，有可能探测包和实际发送的数据包会沿不同的路径到达目的地，在这种情况下，探测实际上是无效的，因此这种方法不能提供严格的QoS保证。

在该接纳控制机制中，还存在一个“带宽窃取”的问题。因为在探测及接纳控制过程中，没有一种很好的机制来衡量新流对已有的流的QoS的影响，所以在某些情况下新流的加入会造成已有的流的QoS的下降。

5 基于带宽代理的接纳控制机制

5.1 基本原理

带宽代理(Bandwidth Broker，BB)的概念最早作为在 DiffServ 模型中支持“奖赏服务”的手段在 IETF RFC2638中提出[3]。

在此方案中，接纳控制，资源管理和政策决策(policy decision)都由每个网络域中的中央控制的 BB 来执行。但目前 BB 的研究还不够完备，RFC 对于 BB 的描述是概括性的。对此，后来的研究者提出了一种改进的 BB 结构[13~15]。在此 BB 结构下，核心路由器只执行数据平面功能(data plane function)，如包调度和包转发，而所有的 QoS 状态信息保存在 BB 中，所有的 QoS 控制平面功能(QoS control plane functions)，如接纳控制、资源预留都由 BB 来执行。图2图示了 BB 体系结构的基本组成部分和工作流程以及 BB 与数据平面(data plane)的关系。

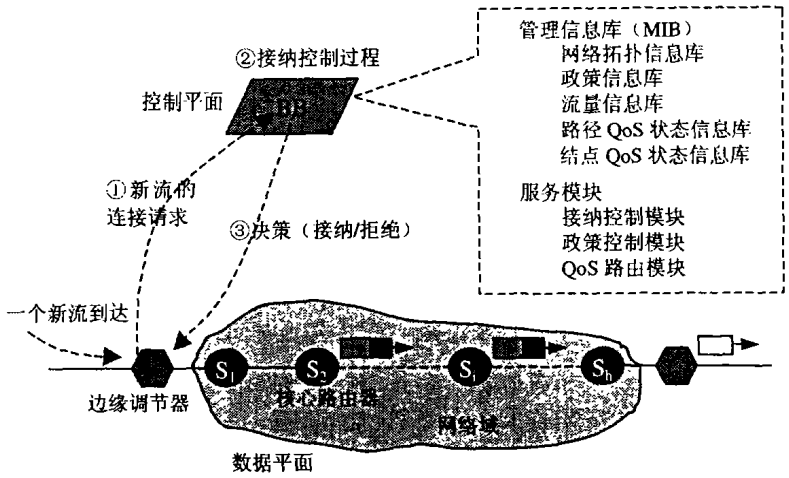

图2　带宽代理(BB)结构及接纳控制流程示意图

如图2所示，BB 包含多个服务模块，如接纳控制，路由和政策控制(routing and policy control)。路由模块协同路由器取得整个网络域的拓扑信息并负责进行路由选择和路径建立。政策控制模块维持政策信息库并负责网络政策管理。接纳控制模块维持整个网络域的 QoS 状态信息(这些 QoS 状态信息被分存于多个 MIB 库中)并负责接纳控制和资源预留。

图中，当一个新流请求网络连接服务时，新流的请求首先由网络的边缘调节器转发至该网络域的 BB，由 BB 对新流的特性和 QoS 请求进行分析，并结合集中保存的各种网络状态信息进行接纳控制决策。

若该连接被接纳，则新流通过边缘调节器进行流量调节和整型。边缘流量调节器在整个体系中起着关键性的作用，它保证每个被接纳的流的包不会以超过预留速率的速率进入核心网络(network core)。即满足以下条件：

$$\hat{a}_1^{j,k+1}-\hat{a}_1^{j,k}\geqslant\frac{L^{j,k+1}}{r^j}$$

其中 $\hat{a}_1^{j,k}$ 表示流 j 的第 k 个包 $p^{j,k}$ 到达第一个核心路由器的时间，$L^{j,k}$ 表示包 $p^{j,k}$ 的大小，r^j 表示为流 j 预留的带宽。

同时，当包经过“边缘流量调节器”进入了核心网络时，在包头被附加了包状态信息(这些信息是在网络边缘被初始化时加入的)。由包携带的包状态信息包含3类信息：1)一个(速率，延时)参数对，这个参数对由 BB 根据流的 QoS 请求确定；2)包的虚拟时间戳；3)包的虚拟时间调节项。其中由包携带的虚拟时间戳 $\tilde{\omega}_i^{j,k}$ 表示流 j 的第 k 个包 $p^{j,k}$ 到达第 i 个核心路由器的虚拟时间，并满足以下两个条件：

$$\cdot\tilde{\omega}_i^{j,k+1}-\tilde{\omega}_i^{j,k}\geqslant\frac{L^{j,k+1}}{r^j}$$

$$\cdot\hat{a}_i^{j,k}\leqslant\tilde{\omega}_i^{j,k}$$

其中 $\hat{a}_i^{j,k}$ 表示包 $p^{j,k}$ 到达第 i 个核心路由器的实际时间。

随后，数据包在核心网络中被传递和转发，当包经过每个核心路由器时包头所携带的“包虚拟时间戳”被引用和更新。“包虚拟时间戳”的更新计算只需用到随包携带的包状态信息以及少数在路由器上预先设定好的参数。因此该机制无需在核心路由器保存状态信息，也是 core-stateless 的。

依据上面的一些约束条件，则可以计算出端到端的延时为：

$$d_{e2e}^j=d_{edge}^j+d_{core}^j=T_{on}^j\frac{P^j-r^j}{r_j}+(q+1)\frac{L^{j,max}}{r^j}+(h-q)d'+\sum_i(\psi_i+\pi_i)$$

其中的推导过程和详细的说明参见文[13,14]。

5.2　接纳控制准则

如果一个新流 j 能够被接纳，则它的端到端延时 QoS 要求 $D^{v,req}$ 应满足：

$$D^{v,req}\geqslant d_{e2e}^j$$

根据上式可解出需预留的带宽 r^v，并根据目前的网络状态信息确定是否有足够的带宽来满足新流的请求。

5.3　评价

在该 BB 结构中，QoS 状态信息都被保存在每个网络域的 BB 中并由 BB 进行管理。核心路由器完全无需进行接纳控制、状态管理等工作从而使得核心路由器的工作效率更高。这种数据平面和控制平面的分离机制还带来以下两个优点：一是允许网络服务提供者引入新的 QoS 服务而无需在核心路由器进行软/硬件升级；二是易于部署更为复杂的 QoS 管理和接纳控制算法以优化网络应用。

同时，在这种结构下，接纳控制基于全路径信息实现，而非基于“逐跳”(hop by hop)来实现。这种方法极大地减少了接纳控制算法的复杂度并便于实现网络服务的优化。

但基于 BB 的机制也存在一个问题，就是 BB 本身，以及连接 BB 和各个网络边缘的链路是否会成为性能的瓶颈，从而影响可伸缩性。当然，DiffServ 模型中，区分服务机制和流聚集对于提高 BB 的可伸缩性是很有益处的。通过层次化的，多个 BB 的协同也可以解决可伸缩性问题。

另外，由于集中控制的 BB，使得 BB 的故障则会导致整个网络服务的瘫痪，因此如何保持 BB 的高可靠性也是一个问题，而且由于信息的集中存放(网络拓扑信息，流状态信息等)，因此信息的动态更新也需有特别的考虑。

跨网络区域的服务所需的多个 BB 之间的协同也是基于 BB 的结构中应解决的问题，IETF 的相关 RFC 虽做了一些描述，但远未达到实用水平。

6　基于令牌的接纳控制机制

6.1　基本原理

最近，有的研究者提出了一种基于令牌的接纳控制机制[16]。在该机制中，将一个网络域区分为由边缘路由器和核心路由器组成。边缘路由器就是处于网络的入/出口点的路由器，而核心路由器则是在网络域内部进行通信转发的路由器。在这种结构下通过类似 OSPF 的链路状态路由协议以便让边缘路由器能够取得网络的拓扑信息。这主要是为了让边缘路由器使用端到端的路径信息来完成接纳控制和带宽预留(而非使用 hop-by-hop 的预留方式)。而在不同的网络域之间可以使用诸如 MPLS 协议等路由连接机制来实现不同域之间

的路由连接。上述所有这些机制用于保证为一个通信流从入口到出口建立一条"虚拟链路"。

当网络接到一个新通信流的连接请求，入口点的边缘路由器根据所掌握的路径信息计算从入口到出口的路径上是否有足够的资源来满足新流的请求，如果可以，则在该路径所包含的所有链路上为新流预留带宽资源。

为了防止边缘路由器重复分配链路的带宽资源，在该体系下，使用了一种"令牌"机制，简单地说，就是将所有连接在一起的边缘路由器在逻辑上视为一个"令牌环"，然后，一个称之为"令牌"的特殊的包在这些边缘路由器之间按一定的顺序循环游动，"令牌"包中保存了网络域中所有的链路的剩余带宽的最新信息及其它的一些网络状态信息。只有当某个边缘路由器拥有"令牌"时(即令牌巡游到该路由器)才允许该边缘路由器对接到的连接请求进行处理，即是否接纳新流(而且也只有当边缘路由器拥"令牌"中的最新的链路剩余带宽信息，才可能进行接纳控制决策)。如果一个新流被接纳，则该边缘路由器还要负责根据分配给新流的带宽信息更新"令牌"中保存的链路剩余带宽信息，然后将"令牌"传至下一个边缘路由器。

另外，如果某个边缘路由器上被接纳蹦某些流结束服务，释放了足够多的资源时，即使令牌目前尚未到达该路由器，该路由器也可对刚被释放的资源预先进行分配，处理缓存中的连接请求(若缓存中有请求存在)。

6.2 评价

在这种令牌巡游的机制下，接纳控制的可伸缩性受到令牌巡游一周所需延时的限制，因为，在最坏的情况下，边缘路由器要等待令牌巡游一周以后才能处理在该路由器上收到的连接请求，因此，随着网络规模的扩大，加入令牌巡游的边缘路由器的增多，令牌巡游一周的延时可能会很大。另外，"令牌"包中需包含全网络域中所有的链路的带宽状态信息，因此随着网络规模的扩大，"令牌"包也会急剧增大，同样会带来令牌延时和管理难度增大等问题。这些都会导致接纳控制的效率和性能的急剧下降。

对于令牌巡游带来的延时问题，一是可以划分网络区域，使得在一个网络域能限制在一个合适的规模下，从而保证令牌巡游带来的延时在一个可接受的范围内。而不同的域的网络分别维持自己的令牌巡游机制，跨域之间的通信则依靠前面提到的路由连接机制来协同实现。二是采用一种预先分配机制，即在网络不繁忙的情况下，允许边缘路由器在未拥有令牌时，对接到的连接请求进行预先分配。表示网络是否繁忙采用链路标记技术，即通过测量动态对网络中繁忙的链路进行标记，对于没有标记的链路，允许边缘路由器对该链路进行前述的预分配。

在该机制中还存在一个潜在的健壮性问题。虽然"令牌"包被当作最高优先级的包在网络中传递，"令牌"丢失和发生错误的可能性极小，但是由于设备故障导致的"令牌"错误或丢失的可能性仍然存在(如某个持有"令牌"的边缘路由器发生故障)，而"令牌"的错误和丢失也将导致网络服务的不可用。所以，应当有一种可靠易行的"令牌"恢复机制以确保令牌的高可靠性。

结论 从上面对各种接纳控制机制的分析和研究中可以看出，各种接纳控制机制都有它们各自的优点和缺点。某些接纳控制机制还有它的适应范围，并针对不同级别的QoS服务进行处理。综合起来，从以下几个因素来考虑(QoS保障程度，网络利用率，可伸缩性，健壮性等)，可以得到以下结论：

在Internet全网内采纳一种接纳控制机制是不现实的，针对不同的业务和QoS要求应灵活采用不同的接纳控制机制。

在一个大规模的网络范围内采用集中式的接纳控制机制也是不现实的，因为网络服务本来就是分布式的，即使采用BB结构的接纳控制机制，也应参照DiffServ模型，将网络划分为不同的区域，在区域内采用一个中央控制的BB，整个大网络内通过层次化的多个BB的协同来完成大网内的接纳控制。鉴于此，则不同的网络区域的QoS接纳控制协同也是一个值得研究的问题。

要实现有效的接纳控制，单纯依赖于一些确定的模型或预先的估计是难以达到目标的，应当结合对网络运行状态的动态测量。并且在一个网络域中，拥有集中存放的、一致的网络状态信息(如网络拓扑结构、链路带宽信息等)对于实现精确的、端到端的细粒度的，接纳控制是非常有帮助的。

为了提高接纳控制机制的可伸缩性，使用流聚集(flow aggregating)是非常有效的办法，但是如何处理流聚集的动态性还有待进一步研究。因为在一个聚集流中，可能会动态地加入新流，也可能有流动态地离去，因此如何动态描述聚集流的特征，并动态调整为聚集流分配带宽资源也是一个问题。

在未来的研究中，我们的目标是要得到一个基于预测的可扩展接纳控制机制。通过预先对网络状态的非精确描述，结合对网络运行状态的动态测量，实现对网络未来运行状态的准确预测，以此来实现高性能的、精确的、可扩展的接纳控制。

参考文献

1 Braden R, Clark D, Shenker S. Integrated Services in the Internet Architecture: an Overview. IETF, RFC 1633, Jun. 1994
2 Blake S, et al. An architecture for differentiated services. IETF, RFC 2475, Dec. 1998
3 Nichols K, Jacobson V, Zhang L. A two-bit differentiated services architecture for the Internet. IETF, RFC 2638, Jul. 1999
4 Tang N, Tsui S, Wang L. A survey of admission control algorithms. Dec. 1998
5 Jamin S, Shenker S, Danzig P. Comparison of measurement-based admission control algorithms for Controlled-Load Service. In: Proc. of IEEE INFOCOM 1997, Apr. 1997
6 Tse D, Grossglauser M. Measurement-based call admission control: Analysis and simulation. In: Proc. of IEEE INFOCOM 1997, Apr. 1997
7 Guerin R, Ahmadi H, Naghshineh M. Equivalent capacity and its application to bandwidth allocation in high-speed networks. IEEE Journal on Selected Areas in Communications, 1991, 9: 968～981
8 Hoeffding W. Probability inequalities for sums of bounded random variables. American Statistical Association Journal, 1963, 58: 13～30
9 Elek V, Karlsson G, Ronngren R. Admission control based on end-to-end measurements. In: Proc. of IEEE INFOCOM 2000, Tel Aviv, Israel, Mar. 2000
10 Bianchi G, Capone A, Petrioli C. Throughput analysis of end-to-end measurement-based admission control in IP. In: Proc. of IEEE INFOCOM 2000, Tel Aviv, Israel, Mar. 2000
11 Breslau L, et al. Endpoint admission control: Architectural issues and performance. In: Proc. of ACM SIGCOMM 2000, Aug. 2000
12 Hill R, Kung HT. A Diff-Serv enhanced admission control scheme. In: Proc. of IEEE Globecom 2001, Nov. 2001
13 Zhang Z, Duan Z, Gao L, Hou Y. Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services. In: Proc. of ACM SIGCOMM 2000, Oct. 2000
14 Zhang Z, Duan Z, Hou Y. Virtual time reference system: A unifying scheduling framework for scalable support of guaranteed services. IEEE Journal on Selected Areas in Communications, 2000, 18(12): 2684～2695
15 Zhang Z, Duan Z, Hou Y. On scalable design of bandwidth broker. IEICE Transactions on Communications, 2001, E84-B(8): 2011～2025
16 Bhatnagar S, Nath B. Distributed admission control to support guaranteed services in core-stateless networks. IEEE, 2003