计算机科学 2002Vol. 29№. 12

包交换网络中调度方法进展:扩展性分析

Processing of Scheduling Algorithm in Packet Switching Network:Analysis of Scalability

杨明川 钱华林

(中国科学院计算机网络信息中心 北京 100080)

Abstract Due to the fast increasing in traffic and applications in Internet and the great improvement of transfer layer technology, the quality of service provision in Internet is more and more focused on scalability issue recently. Scalability should be supported in many elements in QoS architecuture. Among them, the scalability in packet scheduling algorithm is of the great improtance. In this paper, we analyze the processing of packet scheduling algorithm from scalability's point of view. Our work suggests that the scalable packet scheduling is mainly between delay performance and other performance, such as utilization, fairness.

Keywords QoS, Packet scheduling, Packet switching network, Scalability

1. 引言

现在 Internet 主要提供无服务质量(QoS)保证的尽力服务(best effort)。随着 Internet 朝着提供包括数据、声音、视频等多服务统一的多媒体通讯平台发展,传统的 Internet 已经不能满足不同的应用在吞吐率、延迟、延迟抖动、丢失率等方面的不同要求。服务质量控制作为网络提供保证服务的手段在近十多年的时间里得到了广泛的研究。传统的服务质量研究主要基于集成服务(Intserv/RSVP)[1,2]模型。这种模型的基本思想是为每一个流提供端到端的服务质量控制,它的实现依赖于几个方面的技术,一个是用于资源预留的信令协议,一个是转发节点的每流控制,再有就是在端到端的路径上的每节点控制。这几个方面的技术尽管可以确保比较好的服务质量性能,但是同时带来了复杂性和扩展性问题[3]。这些问题随着 Internet 技术和需求的变化而越来越突出。

Internet 的技术和需求的变化主要表现在几个方面:首先是 Internet 的承载网络正在发生根本的改变。这个改变以光通讯技术的进步和广泛应用为基础,它使得网络上的原始可用带宽迅速增长,现在大型骨干网络上普遍采用了 10G 以上带宽的光纤链路。支持 Tb 级的链路带宽的技术也已经出现。而且,可以预见网络的带宽还会以较快的速度迅速增长。随着通讯链路有了更高的带宽,更好的可靠性,网络的体系结构也在发生变化,特别是以 IP Over DWDM 和 MPLS/MPλS/GMPLS 为核心的技术正在改变传统网络的复杂的体系结构,使得网络传输更加灵活,可靠。接入技术也得到了迅速的发展,例如 xDSL,cable modern 等,使得用户访问网络的接入瓶颈不再存在。网络技术的巨大进步极大的促进了 Internet 需求的迅速增加。它一方面表现在用户数量的增加,另一方面表现每个用户对带宽的需求的增加。综合起来,从服务质量控制的角度,这些改变体现在:①网络中流的数量大大增加;②每个包在每个节点的处理时间大大缩短;③网络中可用的原始带宽大大增加。

因此,Internet 对服务质量控制提供了新的背景和要求,成为服务质量研究的新的驱动力。例如,流的数量增加意味着在核心节点的每流控制会是不可扩展的,转发时间的缩短要求降低每个包操作的复杂性,而网络带宽的增加意味着我们可以通过降低利用率来换取性能,等等。当然,这些因素是互相影响的。新的服务质量机制需要这些新的背景下寻求一个更合理的折中。实际上,比较原有的注重每个流的性能和公平性的服务质量控制策略,新的服务质量的控制更加注重扩展性、灵活性。反映在服务质量框架模型上,区分服务[4]得到了越来越多的关注[5~7]。

本文从扩展性的角度讨论了包调度算法近年来的进展,以及这些进展如何反映新的需求。包调度算法是支持服务质量保证的一个核心环节。在一个包交换网络中,来自不同连接的包会在每个交换节点同其它连接的包互相影响,如果没有适当的控制,这种影响会显著地影响网络的性能[8]。通常,包调度可以从两个层面上来讨论,对于那些具有相同或相似的服务保证需求的流(属于某个服务类型的流),包调度算法决定这些不同流的包发送到链路上的顺序。其目的是保证不同的流公平地共享资源并保证给定的服务质量性能。对于那些具有不同的服务需求的流(属于不同的服务类型),包调度算法根据一定的策略决定不同类型的流量如何占用资源。例如优先级的方式,加权共享的方式等等。本文主要讨论前一个层面上的包调度,集中于提供服务保证的调度算法。对包调度的其他层面也有所涉及。

包调度算法作为支持 Internet 扩展性的重要部分近年来得到广泛的研究。传统的包调度算法通常都是基于每流的端到端控制,例如 PGPS(Packetized Generalized Processor Sharing)方法,适合在 Intserv/RSVP 模型下使用。这些方法强调:1)每个流共享资源的公平性,包括平均情况下的公平性,最坏情况下的公平性和额外资源占用的公平性等。2)每个流的严格性能保证,包括延迟、延迟抖动和丢失率。这样,每流的状态维护就不可避免。最终的实现需要在两个方面消耗资源,第一个方面是每流队列的维护和管理,包括包进入队列消耗的时间,队列状态更新消耗的时间等。第二个方面是包调出策略消耗的时间,通常需要对每个队列的当前状态进行比较。这两个方面的操作的复杂度都与当前队列的数量有关,如前面的分析,随着网络上流的数量增加和每个包处理时间的缩短,其实现越来越困难。实际上,公平性的目的在于有效地分配资源,避免恶意流过多占用资源,影响其它流的性能,绝对的公平性因为实现的复杂性是没有必要的。公平性考虑只要能够做到适当的流隔离(或者区分),控制恶意流过多占用资源即可。另一方面,绝对的性能保证对大多数流而言是不必要

杨明川 博士研究生,研究方向为计算机网络体系结构,服务质量。钱华林 研究员,博士导师,研究方向为高性能网络体系结构。

的，大多数流仅需要实现一定的吞吐率保证即可，而对需要性能保证的流，如所谓的任务关键(task critical)流，我们可以利用优先级队列和利用率折中来实现。总之，为了有效的提供服务质量，包调度算法应该以一种简单的，高效的方式实现满足用户要求的服务性能。具体的说，包调度算法有如下的趋势：

- 使用汇集的方式，避免在节点维护每个流的状态。
- 简化调度策略，避免进行每个包的调度决策。
- 使用基于域的模型，而不是端到端的模型，使得需要支持扩展性的节点从复杂的每流控制中解脱出来。由边界来完成复杂的流量整形，标记等工作。
- 使用端到端的合作方式，通过路径上的多个节点协作提高调度性能。
- 考虑利用利用率折中提高性能。
- 与其它 QoS 控制方法，如缓冲区管理、流量整形等协同工作，进一步简化复杂性。

2. 流量模型和分析方法

在网络中有多种模型可用来刻画一个流的流量特征。流量模型可以用来分析流量在网络中的行为，为网络元素的服务质量控制提供依据。例如许可控制(Admission Control)需要利用流量模型来预测缓冲区大小。流量模型也可以用来分析网络的性能。传统上，流量模型基于随机过程[8]。例如 on-off 模型、Poisson 模型、Markovian 模型。这些模型主要用于刻画不同媒体流的流量特征，例如 on-off 可以刻画声音流量，Poisson 刻画数据流量，Markovian 刻画视频流量等等。从理论上可以把网络流量分解为这些流量模型的组合。但是通常这些模型不是太简单，不足以刻画流量的重要特征，就是太复杂，无法进行有效的计算。而且网络流量的研究表明，网络的流量存在自相似性而体现分形的性质。这对有效地刻画流量是不利的。另外一个因素是随着网络的传输，以前规则的(服从一定的流量模型)流量特征会变化，而且变化的结果很难有效的跟踪。这些因素使得基于随机过程的流量模型在适用上有很大的困难。

另外一种流量模型通常不具体刻画流量的过程特征，而只是提供一个流量的约束(bound)。例如(Xmin，Xave，I，Smax)[9]模型要求满足任何两个包的间隔大于 Xmin，在长度为 I 的时间内，平均的间隔长度为 Xave，最大的包大小不超过 Smax。这种方法的优点在于任意的流都可以表示成特定流量约束，而且不满足一定约束的流可以通过流量整形来满足。基于约束的流量模型分析起来也比较简单。现在适用得较多的是(σ,ρ)模型[10,11]，它要求在任意间隔长度 u，流量不超过：$\sigma+\rho u$。实际上 σ 可以看作是令牌桶的大小，而 ρ 可以看作是令牌桶的长期速率。(σ,ρ)的一个重要性质是如果一个满足(σ_0,ρ_0)的流经过了多个节点，在这些节点的最大延迟为 d，则流仍然满足 $\sigma=\rho_0 d+\sigma_0$，$\rho=\rho_0$ 的(σ,ρ)约束。这个性质可以用来分析流在网络中传输的变化。目前，该模型已被广泛的应用于调度算法的研究。

根据不同的流量模型，有多种分析方法。基本的分析方法有两种，一种是基于统计的分析方法，这种方法通常依赖基于随机过程的流量模型，基于统计的方法的优点是可以分析平均情况下的网络性能，例如延迟界限(delay bound)。但是根据前面的分析，基于随机过程的分析在分析的可靠性和复杂度上存在许多问题，因此在实际应用上有许多困难。另外一种分析方法是基于最坏情况的分析，它以服务曲线分析方法[12]为代表，成为广泛采用的分析手段。基于服务曲线的方法首先需要确定在某个链路的输入流量曲线和相应的服务曲线，从这两个曲线可以得出队列延迟和缓冲区大小随时间的变化。如图 1[12]。

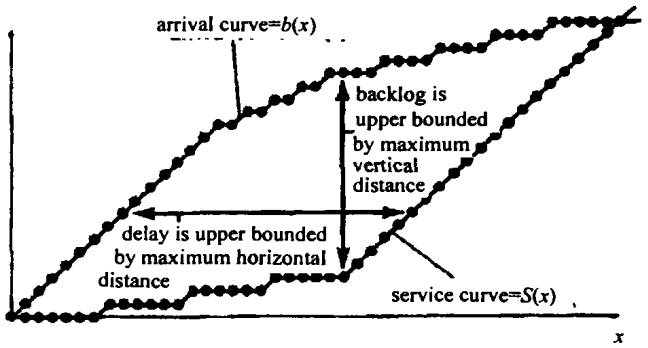

图 1　服务曲线方法示意图

图中，两条曲线的垂直距离表示缓冲区占用，水平距离表示延迟。基于服务曲线的分析和应用通常有两种方式[17]：1)分配法：即根据一定的策略或者需求为流指派服务曲线，然后分析指派的服务曲线封装(envelopment)获得最坏情况的延迟界限，并导出调度策略。2)导出法：根据给定的调度算法导出服务曲线，并分析其最坏的延迟。其它的最坏情况分析还包括使用队列理论的分析。

从端到端的性能分析的角度来看，性能分析还可以分为基于分解的方法和基于集成的方法。基于分解的方法把端到端的延迟分解为各个节点的延迟之和。因此，它侧重讨论每个节点的延迟。这种方法的优点是简单，缺点是往往过多的估计了最坏的延迟，因为在端到端的路径上，每个节点都遭遇最坏情况几乎是不可能的。另外，流传输过程中的流量特征的变化影响了分析的有效性。基于集成的方法考虑在路径上多个节点，考虑这些节点的相互影响。在文[17]中指出，通过计算路径上的网络服务曲线，可以导出路径上多个节点的总的延迟。基于集成的方法可能需要一定的基于合作的调度相配合，我们在后面的部分将具体的介绍基于合作的调度。

3. 传统的调度算法

传统的调度算法根据调度策略可以分为两类：基于速率的方法和基于延迟的方法。基于速率的方法的基本思想是通过模拟按位流抵达处理(fluid access process)来获得速率保证。例如 WFQ[13](Weight Fair Queue)模拟 FFQ(Fluid Fair Queueing)或者是 GPS(Generalized Processor Sharing)，从而实现一种 PGPS(Packetized Generalized Processor Sharing)调度算法。虚拟时钟(Virtual Clock)[14]仿真时分复用(TMD)系统。基于速率的调度的重要性质是它可以实现速率保证，包括和速率相关的端到端延迟界限(delay bound)、吞吐率和延迟抖动(delay jitter)等。基于速率的调度可以有效地隔离流之间的相互影响。

传统的 PGPS 调度算法主要基于单个节点分析，它的一个核心问题是保证调度的公平性，包括最坏情况下的公平性。WF^2Q 是一种提供最坏情况下公平的调度算法。基于速率的方法主要有两个方面的局限：1)扩展性，实际上是每流控制的普遍问题，基于速率的方法首先需要维护每流的速率(或速率加权)信息，同时必须根据每个(活动)流的速率信息计算每个包的截止时间(deadline)，最后的调度决策必须在所有的流的第一个包中比较截止时间。因此，如果流的数量很多时，在时间和空间上的消耗都很大；2)速率依赖性，基于速率的方法在

性能保证上依赖于预留的速率，而不是流的服务质量要求，则两个因素可能冲突，特别是对发送速率比较小的流，其按发送速率得到的性能保证往往不能满足它的服务质量要求。为了满足服务质量要求，应用将不得不提高预留的速率，造成资源的浪费和有效利用率的降低。为了解决扩展性问题，也有一些基于传统的调度方法的改进。例如，SCRQ 使用了自时钟机制解决 WFQ 中需要在任意时刻跟踪活动流的数量来计算加权的速率值引入的复杂性。其基本假设是系统的虚拟时钟可以根据当前正在传送的包的虚拟服务时间来估计。基于 RDD 的方法可以部分解决维护每流状态带来的扩展性问题，但是这些方法的改进算法都不能从根本上解决扩展性问题。

与基于速率的方法不同的是，基于延迟的方法，即 EDF (Earliest Deadline First)调度及其变体，直接给每个包赋予一个截止时间而不是依靠速率计算出延迟界限，然后按照截止时间的顺序调度每个包。相比基于速率的机制，按截止时间调度给与了包调度的灵活性。它表现在调度可以按照每个流实际的服务质量要求进行，而不必受预留速率的限制。通过指定截止时间指派的策略，可以实现不同的调度策略。

基于延迟的调度可以实现和基于速率的方法同样的性能。文[15]指出，辅之以流量整形，EDF 在端到端延迟，缓冲区需求和实现简单性方面优于 GPS。文[16]指出了 EDF 是优化的，即在满足可调度的条件下，EDF 能够保证不会违反延迟约束(delay bound)。EDF 的一个问题是需要满足可调度性条件。这方面有一些工作讨论。主要集中于采用服务曲线分析。另外，为了满足可调度性，引入一定的优先级控制是必要的[17](或者它事实上不得不按优先级来调度)。从扩展性的角度来看，为了提供对每个流的延迟保证，基于延迟的调度也需要每流的知识，而且基于每包的状态比较也是必不可少的。

4. 无状态调度

基于速率的每流控制，如前所述，一个主要的问题在于每个节点需要维护每流的状态，而且流状态的更新依赖于每个包的状态。无状态调度的基本思想是通过提前计算把基于速率的服务质量需求转化成每个节点的截止时间。

最有代表性的无状态调度是所谓的动态包交换(DPS，Dynamic Packet State)[18]方法。它通过在包中携带调度信息来实现无状态的调度。DPS 的实现基于一个类似于区分服务域的域模型 SCORE[18]，其基本方法是在一个域中区分边界节点和核心节点，在边界节点初始化每个包中携带的用于动态计算每个包的合法时间(eligible time)和截止时间的必要的信息。在域的内部(核心)节点根据这些值进行每包的调度。同时根据调度的状态更新这些值。

DPS 的核心是 CJVC (Core-Jitter-VC)，它是一种消去了每流状态依赖的延迟抖动虚拟时钟调度算法(Jitter-VC)[19,20]，Jitter-VC 是一种非持续工作(non-work-conserving)的调度算法，它同时使用了一个延迟抖动控制器(delay-jitter-controller)和一个虚拟时钟调度器。为了消除每流状态，CJVC 必须要消除延迟计算中对前面的包的依赖，这可以分成两步实现：首先需要提前计算出每个包穿越整个域需要的时间，这可以在边界节点完成，因为边界节点维护每流的信息而且具体地知道流中包序列的每个包，同时，边界还知道穿越整个域需要经历的跳数。第二步计算出的相邻包的延迟差，并把这个差均匀地分配到每个跳中。这样，在中间节点，包的截至时间可以通过简单地增加一个固定的值即可。把这个值携带在包中，就可以实现无状态的调度。图 2 显示了无状态 CJVC 调度和 JVC 调度方式在计算截至时间上的变化。

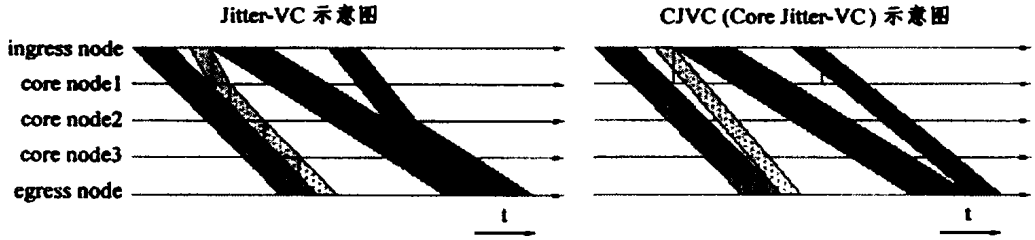

图 2　JVC-CJVC 对截至时间的计算

图 2 中，四边形和线段的交点代表每个包在该线段代表的节点的合法时间和截至时间。四边形的宽度代表该包的最长等待时间(从合法时间开始计算)，它实质上是包长和速率的比。注意，忽略链路传输延迟，前一个节点的截至时间不大于下一个节点的合法时间。所以，包长越大的包，在图上的斜率越大。如果一个包长较小的包紧跟着一个包长很大的包，斜率也会相应变大。在 CJVC 中，每个包的斜率根据端到端的延迟而不是跳到掉的延迟计算，从图上看，在每个跳，包的延迟斜率被拉直了。它就是计算每跳截至时间的依据。

值得指出的是，由于 CJVC 模拟的是 Jitter-VC，因此，CJVC 本质上也是一种非持续工作的调度算法。从图 2 可以看到合法时间的变化。尽管非工作持续的整形对每跳的截止时间的无状态计算是必要的，因为在边界节点无法预测内部节点的动态变化。但是，非工作持续并非是无状态的必要条件，实际上，通过放宽或者取消对合法时间的限制，可以实现在调度上的非工作持续，尽管其截至时间的计算按照非工作持续方式进行。实现工作持续的优点是可以支持更好的统计复用从而更有效的利用链路带宽。

因此，一些文献提出了通用的可以基于工作持续的无状态调度，例如文[21]提出的无状态保证速率(CSGR)调度通过仅考虑某个流已到达包的最大包长避免了复杂的包长依赖计算可以实现无状态的调度，而且由于 CSGR 不对合法时间作任何限制，所以 CSGR 可以实现工作持续的方式并且可以应用于所有的基于速率保证的调度机制上。但是，CSGR 是通过放宽延迟条件而实现的无状态计算，所以没有 DPS 方法精确，而且正如前面所述的，它的截至时间的计算仍然依赖于非工作持续的形式。

文[22]提出了一个通用的基于无状态调度的模型：虚拟时间参考系统(virtual time reference system)。它使用了时间标记和每跳行为(PHB)定义的方法实现了一种通用的模型。该模型从结构上看类似于 SCORE 模型[18]，它的特点是不依赖于任何一种调度算法而实现了服务质量控制和调度机制的分离。调度算法的抽象使得虚拟时间参考系统可以支持有状态的，无状态的速率保证和延迟保证的多种调度方法而采用

一种统一的控制手段。尽管虚拟时间参考系统实现了通用的无状态调度模型，它的无状态调度的核心本质上仍然是 DPS 的方法。

各种无状态的调度都可以提供和 WFQ 同样的服务保证。包括基于速率的最坏延迟保证和公平性，但是考虑到这些方法在计算截止时间时采用的非工作持续性质，因此，其端到端的平均延迟性能会受到影响。

无状态的方法和每流控制的方法相比，其扩展性优势是很明显的：它消除了核心节点维护每流状态的复杂性，对于骨干网络的路由器设计而言，这是一个很关键的因素。无状态调度的复杂性主要体现在每包操作上，包括计算和更新包状态，比较包状态（截止时间）进行调度决策。二者都和当前缓冲区中包的数量有关。对于后者，如果仅对截止时间排序的话，每个包到来时需要的复杂度为：$\log(n)$，n 为包的个数。如果再考虑包的合法时间，复杂度会更高。降低每包操作复杂度的一个方法是采用日历法[22]，或者多级 FIFO 队列。它们以牺牲精确性或链路利用率为代价。无状态调度的另一个折中是需要增加包的负荷，包括流的信息（例如速率），包的信息（例如截止时间或时间戳），调度相关信息（例如提前发送的时间）等等，这些信息的携带，编码对于在传统的 IP 协议上实现是很困难的。

总之，从扩展性的角度来看，无状态调度把复杂性由流操作变成了包操作，把核心节点的复杂性推向了边界节点。这一点和区分服务模型比较类似，但是无状态调度可以提供更好的服务质量保证。由于无状态调度的这些特点，我们认为它在和区分服务结合提供保证的服务，例如区分服务加速转发（EF PHB）上实现是一个比较有吸引力的选择。

包携带信息的思想还可以用在其它领域，例如对 TCP 流实现无状态的公平调度[23,24]，通过在包中携带基于测量的速率信息，可以为中间节点的公平速率分配提供依据。因此，无状态的思想对于提供适应性速率控制协议（如 TCP 协议）也有一定的帮助。

5. 汇集调度

由于每流调度的扩展性问题主要体现在维护每流的状态，因此提高扩展性的另一个方法是通过汇集（aggregation）来减少维护的状态。流汇集把具有某种相同或相似的特征的流汇集起来作为一个宏流处理。所以，流汇集可以大大减少网络设备需要维护和管理的流的数量，降低网络资源的消耗和处理的复杂性，如减少维护的流的状态，简化调度和缓冲区管理，快速重路由等等，提高了系统的扩展性和效率。对于流量汇集的性能，文[25]提出了支持服务质量的流汇集的基本要求：1）汇集前的流状态在汇集后是不可见的（unawareness）。2）流的汇集不会导致流之间的互相影响（保持隔离）。3）保证每个流满足服务质量要求（QoS）。4）流的汇集不影响单个流的预留。

通常，流汇集方式可以以两种方式进行：一种是基于行为的汇集，一种是基于路径的汇集。基于行为的汇集把具有相同和相近的服务质量要求的流汇集在一起，本质上，基于行为的汇集是一种分类的方法。区分服务（Diffserv）模型是采用基于行为的汇集的典型例子。在区分服务中为不同的服务质量需求定义了不同的每跳行为（PHB：per hop behavior），具有相同的 PHB 标记的包在核心节点会被同样的处理。基于路径的汇集则是把在一定粒度定义下经历相同的路径的流汇集在一起。这里，路径可以表示为流经历的节点（node/hop）的序列。粒度表示路径定义的程度。最细的路径粒度是端到端的路径，粗一些的路径粒度可以是源网络出口和目的网络的入口，更粗一点的路径粒度是一个管理域（如区分服务域或 MPLS 域）的入口节点到出口节点。显然，路径粒度越大，允许被汇集的流的数量越多。ATM 的 VC/VP 可以看作一个路径汇集的例子。

我们首先来分析区分服务中的流汇集调度。目前，区分服务模型支持服务保证的加速转发每跳行为（EF PHB）通常都采用简单的 FIFO 调度来实现。为了保证 EF PHB 定义的性能，必须保证转发引擎保持足够小的缓冲区，这通常需要使用一定的优先级策略（如绝对优先级）和过量提供的服务率来实现。一些观点认为，只要链路的利用率足够低（例如 50%），就可以保证足够小的最坏延迟。然而，文[26]指出，即使在汇集的输出整形的情况下，最坏延迟仍然有可能是没有上界的。文[27]的分析进一步给出了延迟有限的条件，即仅当链路的利用率小于 $1/(H^*-1)$ 的情况下，网络的最坏延迟才是有限的，H^* 是网络的直径。实际上，使用汇集的 FIFO 调度的情况下，网络的延迟性能与大量的因素有关。包括网络的拓扑结构，流量的特征，数量和抵达模式等等。

在这些分析的基础上，许多研究希望能够设计出在保证性能和扩展性的基础上提高链路利用率和实现简单的接纳控制的方法。文[28]对延迟根据流的流量特征划分了类，然后根据网络的拓扑（包括每条链路的带宽，连接的情况等）建立了最坏延迟的矩阵。通过求解矩阵可以得到每个节点的最坏延迟。文[28]采用了绝对优先级算法，它的优点是可以根据具体的网络拓扑得到更精确的延迟上限，而且具体的值可以在已知网络拓扑的情况下离线地计算出来，同时接纳控制也可以自然的导出。文[28]通过在 EF PHB 流量内部划分绝对优先级满足不同流对服务质量的不同需求，实现了总体利用率的提高和灵活的服务质量提供。在实现上，绝对优先的方法需要区分优先级和流量类型，并且依据当前的网络拓扑和流量矩阵计算，引入了一些复杂性，而且绝对优先的方法在资源的分配上缺乏灵活性，因此它本质上是性能和扩展性上的一个折中。为了提高链路的利用率，另一些调度方法加入了更多的控制。例如，文[29]提出了 SETF/DETF（static/dynamic earliest time first）的方法。该方法给每个进入域的包打上时间戳，然后根据这个时间戳进行包一级的调度。其中，SETF 不再对时间戳进行修改，而 DETF 需要在中间节点对这个时间戳进行更新。两种方法都可以提高利用率水平，但是后一种方法更复杂，同时其延迟性能和链路利用率水平更高。SETF/DETF 方法虽然可以提供比较好的延迟和利用率性能，但是它一方面需要在边界进行时间标记，这增加了包的负载，同时需要整个域的时间同步；另一方面，内部节点提取时间戳信息增加了对包进行处理的复杂度，同时基于时间戳的转发增加了调度的复杂度。因此，SETF/DETF 在扩展性上的折中较大。上面的分析表明，在满足服务质量保证的情况下，调度实现的复杂度和资源的利用率水平存在一个明显的折中，其尺度依赖于具体的网络环境和策略。

与基于行为汇集的区分服务方法不同，基于路径的汇集则可以看作是集成服务（Intserv）方法的一个扩展。因此，基于路径的汇集本质上是一种基于速率的方法，它可以提供速率保证。根据路径汇集的粒度划分，路径汇集可以分成两类，一种是树形的路径汇集，即路径粒度是多点到点的，汇集可能发生在路径树的任何分支上，从汇集的角度来看，这种汇集是层次式的。文[30,31]指出在包交换网络中基于路径的流汇集也可能支持比较好的服务质量。在文[31]中，作者分析了层次流汇集的延迟，实际上，层次流汇集的延迟来自两个方面，一个

是汇集流转发的延迟，这个延迟由于汇集流的服务率大与其组成流的速率而降低，另一个是汇集时的附加延迟。作者的分析表明，在公平汇集的条件下，汇集时的附加延迟是有限的。因此，流汇集在一定条件下可以减小端到端最坏延迟。另一种路径汇集是线形的路径汇集，路径的粒度是点到点的，汇集在汇集路径（线段）的起始点进行并终止于路径的终点。通常基于线形的路径按照域来定义，例如某个管理域的入口节点到出口节点。这个域从管理上看可以是相对独立的 ISP 运行网络或者从网络结构上看是一个 MPLS 域或者 Diffserv 域（我们在后面将会讨论路径汇集以及 MPLS 和 Diffserv 的关系与互操作性）。通常认为，由于路径汇集增加了控制的粒度，所以服务质量会有所下降，然而，许多分析表明，流汇集反而可能改善服务质量性能，特别是延迟性能。比较树形路径而言，线形的路径汇集对延迟性能的改善可能更大。因为在线形汇集中，只有一个汇集点，进行一次汇集操作，因此，汇集带来的附加延迟很小。文[32]给出了在公平汇集下，比较每流调度的情况，路径汇集提高最坏延迟的条件。另外，基于线形的路径汇集还可以从一定程度上解决前面提到的基于速率的调度和服务质量要求不匹配的问题。文[33]讨论了用汇集集成服务的保证服务流降低总服务率。在文[33]中，汇集的集成服务本质上就是一种线形的路径汇集。在集成服务中，为了保证服务质量，需要比较根据流量速率（包括平均速率和峰值速率）和延迟要求来确定服务率，如果在每流调度下按照速率预留资源不能满足服务质量要求就必须增加预留的资源，即提高服务率。这会影响到链路的利用率。采用汇集的集成服务方法，由于汇集流的总速率比每个组成流而言大大高了，就会缓解这种速率和服务质量要求不一致的情况。文[33]的证明，汇集会有效的降低总的服务率，无论汇集的流在流量的特征上是否一致。它进一步指出，汇集的这个优点主要是因为汇集避免了多个流同时考虑调度负担的情况（pay scheduling errors only once）。

从服务质量的角度来看，基于行为的汇集和基于路径的汇集在提供服务质量的方式和能力方面有很大的区别。基于行为的汇集具有良好的扩展性，因为它仅支持少量的流量类型，服务质量保证通过边界的流量控制，如流量整形和服务极协议（SLA）以及核心的优先级调度来实现。但是，基于行为的汇集要实现服务保证必须依赖于资源的过量提供和优先级控制。基于路径的汇集可以看作是基于每流控制的集成服务模型的扩展，其服务质量保证依靠基于速率的调度。即通过对汇集流的速率保证实现服务质量。基于路径的汇集在扩展性上依赖于路径粒度的定义，路径粒度越大，扩展性越好。

随着 MPLS（多协议标记交换）正成为实现 IP 网络的主流的技术，在 MPLS 上提供服务质量支持成为一个重要的课题。目前，已经有大量的研究讨论了如何在 MPLS 上实现区分服务模型。但是，与基于行为汇集的区分服务模型相比，我们认为：基于路径的汇集在实现 MPLS 服务质量提供方面有许多的优势，包括：

1）它可以提供类似每流控制的速率保证，因为从实现上看，路径汇集本质上还是一种集成服务模型的扩展。

2）通过定义不同的路径粒度支持可伸缩的扩展性。如通过定义不同的 MPLS 转发等效类（FEC：Forward Equivalence Class）可以实现汇集粒度的弹性控制。

3）与 MPLS 的基于 LSP（Label Switching Path）的转发技术有更好的集成性。例如，可以利用 LSP 的标记实现汇集分类和队列调度。

4）借助 MPLS 的通道（trunk）管理，可以降低基于路径的汇集的管理复杂性。包括对资源的汇集预留，汇集流速率的增加和减少，接纳控制等。

值得指出的是，基于行为的汇集和基于路径的汇集讨论了流汇集的不同层面，因此可以综合起来使用。

6. 支持可扩展调度的资源管理

可扩展的调度方法需要相应的可扩展的资源管理的支持，例如资源的预留、分配、维护和撤销等。本节针对扩展调度的两种基本方法：无状态和汇集方法讨论相应的资源管理机制。资源管理的关键问题有几个方面：1）资源预留，通常对于提供服务保证的多媒体流采用 RSVP 进行资源预留。但是 RSVP 是面向每跳每连接的，在无状态或汇集的情况下，RSVP 需要作一定的修改，因为在缺乏每流状态的情况下，资源的预留必须对重复预留，预留信息丢失等特殊的情况提出解决方法，另外它还需要适应包含边界-核心节点的域模型。2）资源分配和接纳控制，在无状态或汇集的情况下，核心节点往往缺乏对流状态的动态变化的准确了解，因此资源的分配和接纳控制必须建立在对流量的变化进行预测的基础上并且需要有一定的机制避免资源的过渡预留或者是过低的利用率。3）资源的维护，因为缺乏每流状态，资源维护通常只能对总的资源占有或者是汇集流的状态进行维护，另外，因为缺乏有效的每流状态刷新机制，节点不能准确的跟踪流的状态变化，例如变为 idle，甚至退出。因此，有效的资源维护必须具有一定测量流量变化并对测量的结果的累计误差进行纠正的机制，资源的维护是资源分配的依据。

目前提出了许多方法解决无状态或汇集条件下的资源管理。文[18]为其提出的 SCORE 体系结构的实现提供了相应的资源管理方法。包括：采用一个 RSVP 的变体进行资源的预留，其主要的预留过程在入口节点和出口节点之间处理，因为入口节点必须了解一些相关的信息，例如穿越整个域的跳数等等。资源的分配和接纳控制在每个核心节点完成，核心节点维护总的资源利用状态，一方面它根据接纳的流和退出的流的速率的修改得到一个总速率的计算值，另一方面，它通过在边界的携带的状态信息的方法得到一个实际流量的测量值。结合这两个值可以对当前的资源预留情况做一个预测，并把它作为接纳控制的基础。结合计算和测量方法的优点是它一方面保证了对流状态变化的正确响应，令一方面可以消除累计误差。

文[34]提出了 DiffRes 方法，它也采用了类似 RSVP 的机制进行基于无状态的域的资源管理。基本思想是策略流量和收集每流的信息在边界节点处理，在核心节点维护临时的每流状态信息（在建立和撤销的时候）。这样，结合一定的超时机制，DiffRes 可以处理控制消息的错误，包括控制消息的丢失，重复的预留等等。DiffRes 有更强的无状态资源管理功能。但是它采用了太多的 RSVP 功能，复杂性比较高。

文[35]的机制和文[18]类似，即不在核心节点维护流的状态信息，也没有跳到跳的信息来处理信令包的丢失以及节点失效等问题，而通过一定周期的调整来对预留发生的错误或误差进行修正，这样减少了核心节点了负担，代价是降低了资源利用率，同时需要一定的周期刷新协议。文[35]使用了基于汇集流量的模型，可以简单的描述为：平均值＋周期变化＋噪声。通过对相关的三个指标进行的分析，包括过载概率、资源利用率和可用带宽。同时考虑了几个因素，包括请求的时间尺度，控制的时间尺度和请求的变化量会对系统的性能带来影响。作者表明本质的问题是寻找一个比较好的在性能和复杂性上的折中。

考虑到基于域(即划分边界节点和核心节点)的无状态模型和区分服务模型的相似性,我们可以采用区分服务模型的一些资源管理机制,例如基于 BB (bandwidth broker)的机制。BB 可以是集中式的,也可以是分布式的。集中式的 BB 的优点是他有全局知识,缺点是请求的响应时间和可能的性能瓶颈。分布式的 BB 则面临一个信息的共享和一致性问题(否则必须把带宽利用率降到更低的水平),现在仍是一个研究的内容[36]。文[37],为了解决中心化 BB 的扩展性问题,作者提出了对粗粒度的汇集流进行 BB 资源预留和分配的方案。

上述的方法大都讨论了无状态的资源管理,对于汇集流的资源管理,它在方式上和无状态的资源管理类似,同时由于它可以在核心节点保留汇集流的状态,在实现上要比无状态的方式简单。基于汇集流的资源管理的另一个问题是随着流加入和退出汇集,会对汇集流调度提供服务质量的性能有所影响。例如,如果流的退出会导致降低汇集流的服务率,就会改变汇集流的延迟保证,这一点在文[33]有详细的分析。因此,在汇集调度的情况下,资源的分配和管理需要考虑更多的因素,也需要较无状态资源管理更为精确的状态信息。

结论 本文主要从可扩展性的角度分析了包交换网络中调度方法的进展。值得指出的是近年来还提出了一些其他的调度算法来提高网络的服务质量性能,但是由于这些算法的主要目的不是针对扩展性,所以本文没有作具体的介绍。例如基于合作的调度[38,39]考虑端到端的延迟补偿和优化,基于学习的调度[40]认为网络应该根据具体的环境和应用的要求适应性的选择调度算法,因此需要具备基于学习的适应性调节能力。

通过本文的分析,我们可以发现在包交换网络中调度的发展中,支持更好的扩展性是一个基本的方向。

参 考 文 献

1 Braden R, Clark D, Shenker S. Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994
2 Braden R, et al. Resource Reservation Protocol (RSVP)-Version 1 Function Specification, RFC 2205, Sept. 1997
3 Guérin R, Peris V. Quality-of-service in packet networks: basic mechanisms and directions. Computer Networks, 1999, 31: 169～189
4 Blake S, et al. An Architecture for Differentiated Service. RFC 2475, Dec. 1998
5 林闯,单志广,盛立杰,吴建平. Internet 区分服务及其几个热点问题研究. 计算机学报, 23(4): 419～443
6 Mathy L, Edwards C, Hutchison D. The Internet: A global Telecommunications Solution. IEEE Network, July/August 2000
7 Xiao Xipeng, Ni L M. Internet QoS: A Big Picture. IEEE Network, March/April 1999
8 Zhang H. Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks. Proceedings of the IEEE, 1995, 83(10),
9 Ferrari D, Verma D. A scheme for real-time channel establishment in wide-area networks. IEEE Journal on Selected Area in Communications, 1990, 8(3): 368～379
10 Cruz R L. A calculus for network delay, part I: Network elements in isolation. IEEE Transaction on Information Theory, 1991, 37(1): 114～131
11 Cruz R L. A Calculus for Network Delay, part II: Network Analysis. IEEE Transactions on Information Theory, 1991, 37(1): 132～141
12 Cruz R L. Quality of Service Guarantees in Virtual Circuit Switched Networks. IEEE Journal on Selected Areas in Communications, 1995, 13(6): 1048～1056
13 Parekh A. A generalized processor sharing approach to flow control in integrated service networks: [Ph. D. thesis]. Massachusetts Institute of Technology, Cambridge, MA 02139, Feb. 1992
14 Zhang L. Virtual Clock: a new traffic control algorithm for packet switching networks. In: Proc. of ACM SIGCOMM'90, Sep. 1990, 19～29
15 Georgiadis L, Guerin R, Peris V, Sivarajan K N. Efficent Network QoS Provisioning Based on Per Node Traffic Shaping. IEEE/ACM Transactions on Networking, 1996, 4(4): 482～501
16 Gorinsky S, Baruah S, Marlowe T J, Stoyenko A D. Exact and Efficient Analysis of Schedulability in Fixed-Packet Networks: A Generic Approach. In: Proc. of the INFOCOM' 97
17 Li Chengzhi, Bettati R, Zhao Wei. New Delay Analysis in High Speed Networks. In: Proc. of the 1999 Intl. Conf. on Parallel Processing (ICPP'99)
18 Stoica I, Zhang Hui. Providing Guaranteed Services Without Per Flow Management. In: Proc. of SIGCOMM'99, Boston, MA, Sept. 1999. 81～94
19 Zhang L. Virtual Clock: A new traffic control algorithm for packet switching networks. In: Proc. of ACM SIGCOMM'90, Sep. 1990
20 Xie G G, Lam S S. Delay Guarantee of Virtual Clock: [Technical Report TR-94-24]. Department of Computer Science, The university of Texas at Austin. 1994
21 Kaur J, Vin H, Core-Stateless Guaranteed Rate Scheduling Algorithms. In: Proc. of IEEE INFOCOM, March, 2001
22 Zhang Zhi-Li, Duan Zhenhai, Hou Yiwei Thomas, Virtual Time Reference System: A Unifying Scheduling Framework for Scalable Support of Guaranteed Services. Fifth INFORMS Telecommunications Conference, Florida, 2000
23 Cao Z, Wang Z, Zegura E W. Rainbow Fair Queueing: Fair Bandwidth Sharing Without Per-Flow State. INFOCOM 2000. 922～931
24 Stoica I, Shenker S, Zhang Hui. Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks, SIGCOMM'98
25 Guerin R, Peris V. Quality-of-service in packet network: basic mechanisms and directions. Computer Networks, 1999, 31: 169～189
26 Andrews M, et al. Universal stability results for greedy contention resolution protocols. In: 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS' 96), Burlington VT, Oct. 1996
27 Charny A, Boudec J-Y L. Delay bound in a network with aggregate scheduling. Proceedings of QOFIS, Oct. 2000
28 Wang Shengquan, Xuan Dong, Bettati R, Zhao Wei. Providing Absolute Differentiated Services for Real-Time Applications in Static-Priority Scheduling Networks. IEEE INFOCOM 2001
29 Zhang Zhi-Li, Duan Zhenhai, Hou Y T. Fundamental Trade-offs in Aggregate Packet Scheduling, submitted. 2001
30 Cobb J A. An In-Depth Look at Flow Aggregation for Efficient Quality of Service. ICNP'1999, Toronto, Canada
31 Cobb J A. Preserving Quality of Service Guarantees in Spite of Flow Aggregation. ICNP'98
32 Mingchuan Y, Hualin Q. MPLS-based Flow Aggregation Performance Analysis. ICYCS'2001
33 Schmitt J, Karsten M, Steinmetz R. On the aggregation of guaranteed service flows. Computer Communications, 2001, 24(1): 2～18
34 Sisalem D, Krishnamurthy S, Dao S. DiffRes: A reservation protocol for the differentiated service model: [tech. rep.]. HRL Laboratories, Malibu, USA
35 Fu Huirong, Knightly E W. Aggregation and Scalable QoS: A Performance Study. In: Proc. of IWQOS 2001
36 Khalil I, Braun T. Implementation of a Bandwidth Broker for Dynamic End-to-End Resource Reservation in Outsourced Virtual Private Networks. In: Proc. of the 25th Annual IEEE Conf. on Local Computer Networks (LCN'00)
37 Zhang Zhi-Li, Duan Zhenhai, Gao Lixin, Hou Y T. Decoupling QoS Control Form Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services. In: Proc. ACM SIGCOMM'00, Sweden, Aug. 2000
38 Li Chengzhi, Knightly E. Coordinated Network Scheduling: A Framework for End-to-End Services. In: Proc. of IEEE ICNP 2000, Osaka, Japan, Nov. 2000
39 Andrews M, Zhang L. Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule. INFOCOM '99
40 Melo Jr. A, Coello J M A. Packet Scheduling Based On Learning in the Next Generation Internet Architectures. In: Proc. of the 5th IEEE Symposium on Computer and Communications (ISCC'00)