

63-66, 71

计算机网络

拥塞控制

分组交换

资源共享

(16)

计算机科学1999Vol. 26No. 9

计算机网络中的拥塞控制技术研究^{*}

Survey of Congestion Control Techniques for Computer Networks

朱利^{1,2} 周俊辉² 郑守淇¹

TP393

(西安交通大学计算机系 西安 710048)¹ (西北核技术研究所 西安 710024)²

Abstract In this paper the concept and principle of congestion and congestion control in computer networks are specified and a taxonomy for congestion control algorithms is proposed. From the point of network and application, some important algorithms of congestion control are introduced according to this classification. In the meantime the recommendation standards for designing congestion control mechanisms are presented here.

Keywords Congestion, Congestion control, Bottleneck, Quality of service

1 引言

当前的计算机网络一般使用分组交换,动态分配网络资源。分组交换的优点是所有的源结点统计共享网络资源,但当源数目不断增加时会加重业务流的不可预测性和突发性,在瓶颈处引起资源冲突,从而导致网络拥塞。拥塞发生时,网络资源处于饱和状态,信息传送时延变长,丢失率上升,系统吞吐率急剧下降,严重的可能导致网络崩溃。Internet上最早的拥塞发生于1986年10月,当时的网络吞吐率从32Kbps下降到40bps^[1]。随着网络互连和应用特别是连续媒体应用的增长,网络拥塞问题会越来越引起人们的关注。

网络拥塞源于资源共享。在分组交换网中,交换处理机速度、通信信道和缓存空间通常既是所有主机共享的网络资源也是网络系统潜在的瓶颈。与网络拥塞密切相关的因素是系统存在瓶颈和源端无限制的业务发送。前者是由网络系统的结构决定的,并随网络结构的发展而逐步改善。然而,不管是Internet,还是B-ISDN,其物理资源相对于用户需求总是不足的。因而,要解决拥塞问题必须在共享资源管理的基础上控制源端的数据发送率,合理使用瓶颈处的资源,避免网络拥塞或将网络从拥塞状态恢复至正常工作状态,这就是网络拥塞控制的基本思想。经过二十多年的研究与发展,人们提出了多种拥塞控

制方法,如“窗口式流控制^[2]”、“源抑制^[3]”、“慢启动^[4]”、“二元反馈^[5]”、“基于速率的控制^[6]”及“自适应拥塞控制^[6]”等,其中部分控制机制已成功地在网络中。

尽管拥塞控制已有了相当的发展,由于网络规模扩大速度很快,网络拥塞仍然是一个十分严峻的问题。特殊的网络应用也需要不同寻常的拥塞控制技术支持,因而拥塞控制技术研究对于将来的网络设计、协议设计及网络应用均有着十分重要的意义。

2 拥塞及拥塞控制的基本原理

根据网络的性能特征,我们定量地说明拥塞和拥塞控制的概念及内涵。图1是根据文[7]综合而成,它表明了分组交换网络的性能(功率、来回时间、吞吐率)负荷关系。

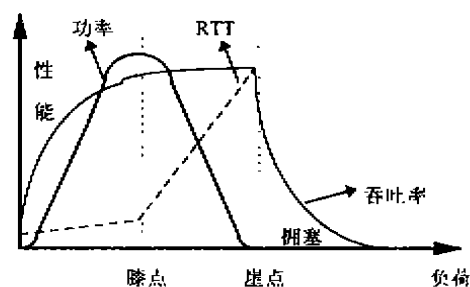


图1 网络性能-负荷关系

^{*} 本文受“863计划项目”资助。

图1中资源功率(Resource Power)=吞吐率/响应时间;膝点(Knee)指资源功率到达最大值时的负荷量;崖点(Cliff)指资源功率下降至最小值且开始丢包时的负荷量。

由图1可以看出,当负荷较小时,网络吞吐率和资源功率随负荷的增加而以指数级增长,RTT(来回时间)随负荷的增长略有上升。当负荷到达膝点时,功率达到最大值,从此点开始,吞吐率的增长远远慢于负荷的增加,RTT直线上升,功率急剧下降。此时若负荷继续增加,接近或达到系统的容量时,交换结点处的队列变得很长,潜在地导致丢包丢失。负荷到达崖点时,吞吐率最终达到最大值,功率到达最小值。此后,RTT以指数级增长,吞吐率急剧下降至一个非常小的值(有可能为0),系统处于拥塞状态。由此可见,膝点是系统最为理想的工作点。拥塞控制就是采用某种策略或机制,保持网络工作在正常的状态下,也就是使网络经常工作在崖点左侧的区域内。若一种控制机制使得网络工作在膝点附近,该方法称之为拥塞避免(Congestion Avoidance);若一种控制机制使得工作在崖点或崖点以后的网络回复至膝点前后,该方法称之为拥塞恢复(Congestion Recovery)。拥塞避免策略是预防性的,目的是保持网络工作在最大功率点附近,不发生网络拥塞;而拥塞恢复策略是救治性的,当拥塞发生后,使得网络复原至正常的工作状态。没有拥塞恢复机制,当拥塞发生时网络有可能崩溃。因此,即便使用了拥塞避免策略,仍然需要拥塞恢复技术来保持吞吐率,因为网络中源结点发送数据的起始时间、数据率及大小具有不可预测性和突发性,有可能会引起网络拥塞。可见,拥塞控制应包括拥塞避免和拥塞恢复两者。

3 拥塞控制机制

从整体上看,拥塞控制可分为两类,一类是“基于网络系统的拥塞控制”,另一类是“基于应用的拥塞控制”。早期的拥塞控制算法大多是基于系统的,如“位轮流公平排队”算法^[1]、“基于服务的控制”方法^[9]及前面所述的几种算法等。最近几年出现了面向具体应用的拥塞控制算法,集中于音、视频通信控制方面^[10],我们将其归之于第二类。

3.1 基于网络系统的拥塞控制

这类拥塞控制是网络系统居主动地位的控制方式。在这种方式下,网络动态地管理其资源,如带宽、缓冲区、队列长度、时钟等,根据资源的占用情况标识拥塞并将拥塞信息反馈至源端,驱动源端调节输

入业务流。控制系统的模型如图2所示。其中,时间片 t 内第 i 个用户的负荷为 $x_i(t)$,瓶颈处的总负荷就为 $\sum x_i(t)$,系统的状态可用 n 维向量来表示: $x(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}$,其中, $x_i(t)$ 表示第 i 个用户的需求及系统资源的分配。在此时间间隔内系统根据负荷级发送反馈信息 $y(t)$ 。整个控制过程涉及到四个控制因素:包调度、缓存管理、信息反馈及端调节。

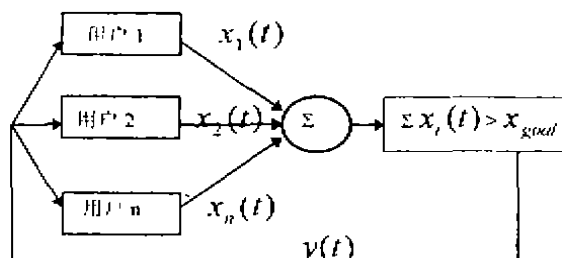


图2 n 用户共享网络的控制系统模型

·包调度。包调度即控制数据包在网络中的服务顺序和包数量的分配等。两个比较流行的包调度算法是FCFS和FQ(公平排队)算法。FQ将带宽平均分配给当前的流,交换结点为每个源分别维持一个队列,轮流传送各队列中的数据包,和FCFS相比,其公正性要好得多。

·缓存管理。即如何分配缓冲区给不同的流,当超载时选择丢弃哪些包。当用户可用的有效带宽减少时,系统必须提供充分的缓冲机制以使用户有时间减少其负荷。因此,缓存管理是拥塞控制的一个重要组成部分。两类不同的缓存管理方法是共享缓冲池方法和按流分配方法。前者将所有的流所需的缓冲区汇聚成一个大的缓存池,按先来先使用方式服务(即FCFS),当共享池的占用量超过某一设定值时丢弃后到的包;与此相反,按流分配法则为每个流分别分配一缓存,根据每个缓存的占用情况丢弃不能缓冲的数据包。“输入缓存限制”算法^[2]采用的就是共享缓冲池方法,在拥塞情况下,输入业务流被终止,输出业务流使用全部的缓冲池。

·反馈机制。即网络提供给源端状态信息,用以调节源的业务输入流。反馈既可以是前向的也可以是后向的。在前向反馈中,拥塞标志在跳结点上被标记后发往目的结点,由目的结点再返回源结点;而后向反馈中的拥塞标志信息则直接由中间跳结点按流的逆向返回至源结点。“二元反馈方法”使用的即是前向反馈。在该算法中,每个数据包的包头中均含有

一位拥塞指示位(为1时表示拥塞,为0时表示非拥塞),当某个路由器的平均队列长度大于或等于1时,路由器在网络层将该位置1,继续向前传送。数据包到达目的站后,拥塞指示位被复制到传输层的反馈包包头中,然后发回源站。源站根据拥塞指示位的统计数字来决定发送窗口的大小,当50%的位被设置时,发送窗口减少为当前窗口的87.5%,否则窗口增1。也就是说,算法采用的是“加性增/乘性减”策略。在“二元反馈方法”的基础上进一步发展的算法有“可选择二元反馈算法^[11]”、“Q位方法^[12]”等,前者将统计平均排队时间间隔内每个用户的数据包并保存下来,只拥塞那些过量发送数据的源标识,要求其减少负荷;后者的每个数据包中除了拥塞位外,增加了排队位,以标识交换结点中数据包排队情况,当一个包被迫在队列中等待时,Q位置位,源结点根据拥塞和排队信息来调节输入的流率。

除了使用队列长度和排队时间作为拥塞度量对象外,还可以使用其它对象如资源利用率、丢包率、缓存占用量、业务流量(输入和输出流量)等。在“BBN”算法^[13]中,每个交换结点计算出其资源利用率,加到路由器的更新包内以扩散方式传播至整个网络,各个源端点据此调节自己的流率;在“丢失负荷曲线算法^[14]”中,每个交换结点监视局部的业务负荷,提供一个丢失负荷曲线,反馈至源端,源端根据此曲线导出大致的吞吐率曲线,然后对吞吐率和丢包率进行权衡折衷,决定自己的传送率。著名的“TCP慢启动”算法以源结点到目的结点的RTT作为测试对象进行拥塞决策,反馈至源结点的拥塞信息决定发送窗口的增减(线性增/乘性减)。

•端调节。端系统调节是对伺服控制环的响应,端调节算法和反馈形式及网络中的测试对象密不可分。不管采用什么反馈机制,最终都要通过端调节来实现流控的目标。“反馈控制”中已说明了一些端调节的具体算法。

3.2 基于应用的拥塞控制

基于应用的拥塞控制是针对特殊应用的,和具体的应用程序结合起来并包含于应用程序中,这类控制机制一般不关心网络的具体结构,而是将整个网络看成一个黑匣子,由端结点通过对网络状态参数的测试来获取拥塞信息,再根据拥塞信息按某种策略控制端结点的输入流。最常用的测试参量一般是RTT和丢包率。基于应用的拥塞控制模型可表示成图3形式。其中,S表示源结点集;D表示目的结点集; $\lambda(t) = \{\lambda_1(t), \lambda_2(t), \dots, \lambda_n(t)\}$,表示不同源的

业务率; $f(t+\tau)$ 表示D所发送的反馈信息矢量, τ 表示来回时延;实箭头表示正常业务流的传输路径和方向;虚箭头表示反馈信息的传输路径和方向。

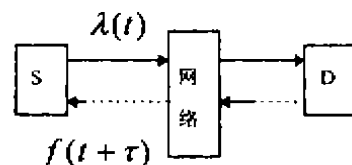


图3 基于应用的拥塞控制模型

和具体应用结合起来的拥塞控制算法一定程度上继承了“基于网络系统的拥塞控制算法”的思想,通过对RTT测试、估算或丢包率反馈来估计当前网络的状态。根据当前网络状态,采用发送窗口限制或发送速率限制等源自约束规则,调节进入网络的数据包量,所有这些操作均是在应用层进行。

•单播实时视频通信中的拥塞控制

视频通信中拥塞的判别基于目的端回至源端的关于视频质量的反馈信息,包括所接收的最大包长、丢包数、估计的包抖动间隔、延迟等。可以简单地只使用丢包率来判定网络状态,如当丢包率大于所设定的阈值时拥塞标志置位,低于某个阈值时欠负荷标志置位;也可以使用综合信息来识别网络是否拥塞。拥塞发生时需要降低编码器的输出码率,如果接收端对图象质量要求不高,可采用PR(速率优先)方式增大量化系数、运动检测阈值或减少色差分量降低编码器的输出率;若对图象质量要求较高,可采用PQ(质量优先)方式降低视频帧的采集速率来降低编码器的输出率。一种拥塞控制算法设计如下:

```
if (CONGESTION)
    Current_Rate = max(Current_Rate * μ, MIN_RATE);
else if (NOCONGESTION)
    Current_Rate = min(Current_Rate + δ, MAX_RATE);
```

算法采用了“加性增/乘性减”策略,其中 $0 < \mu \leq 1$;为了保证图象的最小质量和避免过大的流注入网络,对Current_Rate作了限制:

$$\text{Current_Rate} \in [\text{MIN_RATE}, \text{MAX_RATE}].$$

•组播实时通信中的拥塞控制

组播(Multicast)方式下,网络的异构性决定了其拥塞控制不同于单播方式。一个接收者通过反馈向源报告了丢包信息,源如果简单地限制业务流就会影响整个组内的那些未拥塞结点的业务质量,当拥塞发生在源端或靠近源端时,组内所有接收者检测到拥塞并向源反馈拥塞报告包,这样有可能在源

或瓶颈结点处发生“信息内爆”。因此,组播下的拥塞控制必须很好地解决这些问题。

一种方法是将网络状态分成“欠负荷”、“满负荷”和“拥塞”三类,由接收者根据丢包率来决定其属于哪一类并将状态信息反馈到源端,源端对来自不同接收者的状态信息进行统计,根据统计值调节发送的码率。算法如下。

①对丢包率作低通滤波处理,以避免 QoS 振荡,低通滤波器设计为: $\rho = (1 - \alpha)\rho + \alpha b$, 其中 b 为刚计算出的丢包率; ρ 为平滑后的丢包率; $0 \leq \alpha \leq 1$; 如图4所示,定义两个阈值: ρ_c, ρ_u , 当 $0 \leq \rho \leq \rho_u$ 时,网络状态为欠负荷; 当 $\rho_u < \rho \leq \rho_c$ 时,网络状态为满负荷; $\rho \geq \rho_c$ 时,网络状态为拥塞; ρ_c, ρ_u 可通过实验进行调整。

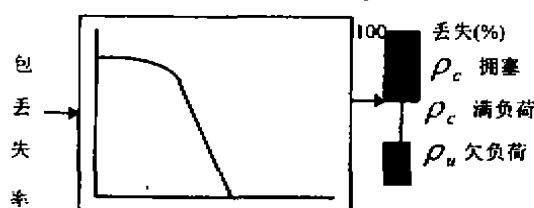


图4 低通滤波器

②用 n_c 表示报告拥塞的目的站数; n_l 表示满负荷的目的站数; n 表示组内所有目的站数。定义拥塞的阈值为 N_c , 满负荷的阈值为 N_l ; r_n 为当前发送要采用的码率, r_{n-1} 为本次控制执行前的发送码率。MIN-RATE 为保证最小业务质量的发送码率; MAX-RATE 为源可用带宽的上限; $\mu \in [0, 1]$, 为乘性减因子; δ 为加性增因子; $r_n, r_{n-1} \in [\text{MIN-RATE}, \text{MAX-RATE}]$ 。

```
if ( $n_c/n > N_c$ ) // 拥塞
     $r_n = \max(r_{n-1} * \mu, \text{MIN-RATE});$ 
else if ( $n_l/n > N_l$ ) // 满负荷
     $r_n = r_{n-1};$ 
else // 欠负荷
     $r_n = \min(r_{n-1} + \delta, \text{MAX-RATE});$ 
```

在此算法中当 N_c, N_l 选值较小时,体现减优先特性;例如, $N_c = 0.1, N_l = 0.1$ 时,只有当 80% 的接收者处于欠负荷状态时才增加码率。反之, N_c, N_l 选值较大时,算法体现增优先特性。

另一种方法是应用分层,即将应用分成多个层次,不同的层对应不同的业务质量。拥塞支路上的接收者对应于拥塞组,只接收基本层数据,当拥塞恢复后加入到非拥塞组,除接收基本层数据外还接收增强层数据,以提高业务质量。

4 拥塞控制算法设计标准

拥塞控制算法设计的一般原则是使得资源功率最大,但还要考虑其它因素。具体地说,控制算法设计时要考虑这样几个主要因素:高效性、公正性、分布性和收敛性。

·**高效性**。资源利用的效率可通过总负荷接近膝点的程度来定义。若 x_{goal} 表示膝点处所期望的负荷级,那么 $x(t) = \sum x_i(t) \approx x_{\text{goal}}$ 时资源利用率最高。高效性只和总资源分配有关,两种不同的资源分配只要总资源分配接近目标值可能都会产生高效性,但二者的公正性不会相同。

·**公正性**。一般来说,若资源是系统的主要瓶颈,多用户平分资源时公正性最好。即对于所有共享瓶颈的业务流 $x_i(t) = x_j(t)$ 时系统公正地工作,量化公正性的函数为:

$$F(X) = (\sum x_i(t))^2 / (n \sum x_i^2(t))$$

$F(X)$ 有下列主要特性:

■ $F(X) \in (0, 1)$, 完全公正时, $F(X) = 1$; 完全不公正时, $F(X) = 1/n$;

■ 公正性和测量单位无关;

■ 公正性函数是连续函数,分配中的微小变化均可在公正性上反映出来;

■ 若 k 个用户均享资源 ($k < n$), 其余 $n - k$ 个用户未接收任何资源,公正性为 k/n 。

·**分布性**。集中式方法需要系统的完全信息(如每个用户的资源需求或总量),传送此类信息至用户会带来相当大的开销,特别是当一个用户同时使用几种资源时。因此,拥塞控制宜采用反馈信息量需要较少的分布式方法。在分布式方式下,反馈信息可只包含欠负荷和超负荷二元反馈位,其它信息如 x_{goal} 及共享资源的用户数可不必传送给用户。

·**收敛性**。拥塞控制方法必须是收敛的,收敛性可通过系统从任一起始状态到达目的状态的速度或时间来测量。对于二元反馈,系统一般不会收敛于一稳态,而在平衡点上下振荡。系统到达平衡点的时间和振荡幅度决定着控制算法的收敛性。其中,时间决定着控制的响应性,幅度决定着控制的平滑性。理想的控制算法其响应时间和振荡幅度均应很小。

“加性增/乘性减”控制策略满足了上述要求^[15],这就是为什么目前的拥塞控制算法多采用此策略的原因。

(下转第71页)

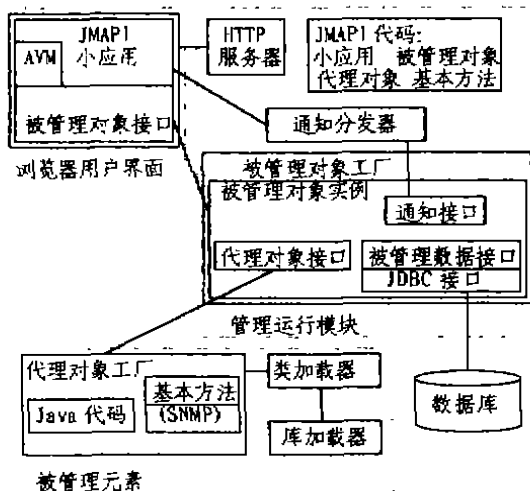


图7 JMAPI 管理体系结构

结束语 本文综述了网络管理体系结构的研究,在分析集中方式的缺陷的基础上,从 DME、Delegation 和 CORBA 三个方面讨论了分布方式体系结构,Internet 技术的发展对网络管理体系结构的研究起一定的推动作用。

参考文献

- 1 Divakara K U. Network Management Systems Essentials. McGraw-Hill. 1995
- 2 William S. SNMP and SNMPv2: The Infrastructure for Network Management. IEEE Communications Magazine, 1998(Mar.)
- 3 Ray H. SNMP, SNMPv2 and CMIP—the Technologies for Multivendor Network Management. Computer Communications, 1997, 20(2)
- 4 Lakshmi R. OSI Systems and Network Management. IEEE Communications Magazine, 1998(Mar.)
- 5 Mohsen k, et al. Decentralised Approaches for Network Management. Computer Communication Review, 1997, 27(3)
- 6 Open Software Foundation. DME. Available at: <http://web1.osf.org/comm/lit/lit-dme.html>
- 7 Schonwalder J. Network management by delegation—From research prototypes towards standards. Computer Network and ISDN Systems, 1997, 29(11)
- 8 German G, Yechiam Y. Delegated Agents for Network Management. IEEE Communications Magazine, 1998 (Mar.)
- 9 Steve V. CORBA: Integrating Diverse Application Within Distributed Heterogeneous Enviroments. IEEE Communication Magazine, 1997(Feb.)
- 10 Alexander K. Service-based Systems Management, Using CORBA as a Middleware for Intelligent Agents. In: Proc. of the IFIP/IEEE Interl. Workshop on Distributed Systems, Operations & Management. L' Aquila, Italy, 1996
- 11 Luca D, et al. Static vs. Dynamic CMIP/SNMP Network Management Using CORBA. In: Proc. IS & N' 97. Como, Italy, 1997
- 12 Joint Inter-Domain Working Group, X/Open and Network Management Forum. Joint Inter-Domain Management Specifications: Specification Translation. April 1995
- 13 Juan P, et al. CORBA for Network and Service Management in the TINA Framework. IEEE Communication Magazine, 1998(Mar.)
- 14 OMG. CORBAServices: Common Object Services Specification. OMG Document, November 1997
- 15 Won-ki H J, et al. Web-Based Intranet Services and Network Management. IEEE Communication Magazine, 1997(Oct.)
- 16 Patrick T J. Web-based Enterprise Management Architecture. IEEE Communication Magazine, 1998(Mar.)
- 17 Raouf B, et al. An Outlook on Intranet Management. IEEE Communications Magazine, 1997(Oct.)
- 18 Available at: <http://wbem.freerange.com>. WBEM Document
- 19 Sun Microsystems, Inc. Java Management API Architecture. June 1996

(上接第66页)

结束语 拥塞及其控制是计算机网络设计和网络应用必须面对的问题。本文提出了一种新的分类方法,将计算机网络中的拥塞控制机制分为“基于网络系统”和“基于网络应用”两大类,并从此角度出发,较系统地说明了拥塞与拥塞控制的概念、本质、具体方

法和设计标准。随着连续媒体在网络上越来越多的应用,拥塞控制越来越显现出其重要性,我们希望本文的研究能对将来的网络设计和网络应用提供一些帮助。

(参考文献共15篇,略)