组播拥塞控制综述*

石 锋, 吴建平

(清华大学 计算机科学与技术系 网络技术研究所,北京 100084)
E-mail: shf@csnet1.cs.tsinghua.edu.cn
http://netlab.cs.tsinghua.edu.cn

摘要: 在组播获得广泛应用之前,必须解决拥塞控制问题.组播拥塞控制有两个重要的评价目标:可扩展性和TCP友好(TCP-friendly).围绕这两个评价目标,介绍组播拥塞控制的研究现状,从不同角度对组播拥塞控制算法进行分析,并讨论最近的组播拥塞控制协议,最后指出今后的研究方向.

关 键 词: 组播;拥塞控制;可扩展性;TCP-友好
中图法分类号: TP393 文献标识码: A

在单点到多点的通信中,组播是一种有效的数据传输方式.然而至今组播无法得到 ISP(Internet service provider)的广泛应用,一个重要的原因在于组播没有提供合适的拥塞控制.近来,组播拥塞控制得到了广泛的关注,它已成为互联网研究领域中的一个热点课题.

在组播拥塞控制中有两个重要的评价目标:可扩展性和TCP友好.可扩展性是指随着组规模的增大,拥塞控制协议不会造成组播性能的下降.TCP友好则要求组播流量和TCP流量能公平地竞争带宽.这两个评价目标在一定程度上是对立的,组播拥塞控制协议需要根据实际需求在两者间作出权衡.

本文围绕两个评价目标,对各种组播拥塞控制算法进行分析,同时讨论最近的一些组播拥塞控制协议.第1节介绍组播与拥塞控制的基本概念,包括Internet服务模型和拥塞控制、IP组播和拥塞控制.第2节介绍组播拥塞控制的两个评价目标.第3节讨论几种组播拥塞控制算法分类.第4节将讨论一些具体的组播拥塞控制协议,分析它们的优缺点.第5节指出未来研究中需要解决的问题和研究方向.第6节对全文进行总结.

1 组播与拥塞控制

1.1 Best-Effort服务模型和拥塞控制

传统的Internet使用Best-Effort服务模型,所有分组受到同等对待,网络尽力发送每个进入网络的分组,不保证服务质量(吞吐量、端到端延迟、丢失率等).由于Internet没有提供接入控制(admission control),用户获得的服务质量不仅取决于网络自身,也取决于其他用户在网络中产生的负载,这使得整个网络完全缺少隔离和保护.

拥塞是一种持续的网络超负荷状态[1].当用户需求大于网络供给时,网络就会发生拥塞.拥塞与Best-Effort服务模型有紧密的联系,在Internet中,用户无法协作共享资源.多个用户可能对同一网络资源提出请求,从而导致拥塞.

文献[2]使用图1描述拥塞.网络负载较小时,吞吐量与网络负载基本保持线性关系,网络延迟缓慢增加;网络负载超过第1个关键点Knee后,网络吞吐量增长变慢,网络延迟增长变快;网络负载超过第2个关键点Cliff后,网络吞吐量急剧下降,网络延迟迅速增加.拥塞控制的目标就是使网络在关键点Knee附近工作.

* 收稿日期: 2002-01-29; 修改日期: 2002-06-25
基金项目: 国家自然科学基金资助项目(90104002,69725003);国家高技术研究发展计划资助项目(2001AA121013)
作者简介: 石锋(1976－),男,湖北沙市人,博士生,主要研究领域为计算机网络体系结构;吴建平(1953－),男,山东巨野人,博士,教授,博士生导师,主要研究领域为计算机网络体系结构,协议工程学,互联网络.

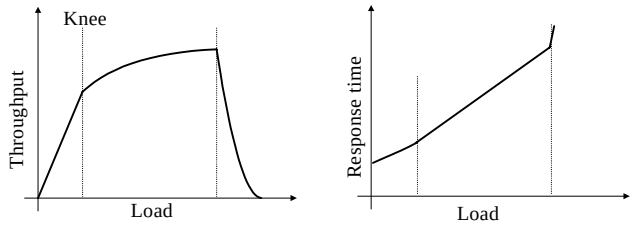

Fig.1
图 1

另外,拥塞控制需要保证流量间的公平性.公平性问题与拥塞密切相关:在低负载情况下,每个用户对资源的请求都能得到满足,这时不需要考虑资源分配的公平性问题;当资源竞争出现时,公平性才成为需要考虑的问题.

1.2 IP组播中的拥塞控制

Stephen 在文献[3]中描述了基于 IP 的标准组播模型.IP 组播通过组播路由协议在网络中建立组播转发树,转发树负责将发送端数据转发到接收端.然而与传统 IP 相同,IP 组播不提供速率控制.因此,组播流量可能耗光所有的网络资源,导致网络拥塞.为 IP 组播设计合适的拥塞控制成为迫切需要解决的问题.

IETF 将组播分为两种模式[4]:单点对多点、多点对多点.后者在组播树生成和组管理等方面存在较大的难度,现阶段在 Internet 上推广的希望不大,而且它可以通过多组单点对多点通信来实现[5].本文的讨论仅限于单点对多点模式.

2 组播拥塞控制协议的评价目标

组播拥塞控制协议具有很强的针对性,大部分协议都是针对某些特定问题提出的[5].需求多样性导致了组播拥塞控制协议的评价目标多样化,我们认为其中的两个评价目标对组播的发展尤为重要:可扩展性和 TCP-友好.

2.1 组播拥塞控制协议的可扩展性

组播拥塞控制协议的可扩展性是指协议在性能(包括吞吐率、延时)下降前可以支持多少用户.它受到 4 方面因素的限制:

(1) 任务复杂性:当组成员的数量变得越来越大时,拥塞控制任务的复杂性会急剧上升,从而限制协议的可扩展性.可以通过在发送端和接收端之间进行合适的分工来解决这个问题.

(2) 反馈爆炸问题[6]:拥塞控制需要考虑所有组成员的拥塞状况,随着组的规模增加,大量的反馈可能湮没发送端.可以通过反馈聚合或反馈抑制机制来解决这个问题.

(3) LPM 问题[7]:指出,随着组规模的增加,组播树的数据丢失路径会随之增加,从而导致大多数分组至少会经历一次丢失,如果发送端对每一次丢失作出响应,组播吞吐量可能下降为 0(drop-to-zero),这就是组播中的丢失路径多样性(loss path multiplicity,简称 LPM)问题.适当的反馈聚合和反馈抑制可以减轻 LPM 问题对组播组性能的影响.

(4) 网络随机延迟的影响[8]:指出,即使在非常理想的网络环境中(网络中无分组丢失,路由器缓存无限大),随着组规模的增加,网络中随机分布的队列延迟(路由器的服务延迟)也会给组播组的性能造成影响.在大的组播组中,多速率组播可能是更好的选择.

2.2 组播拥塞控制协议的TCP-Friendly

TCP 在现在的 Internet 中占据了统治地位,然而随着网络应用的发展,组播流量可能在未来的网络中变得越来越重要.组播拥塞控制协议应该保证组播流量与 TCP 流量之间公平的竞争资源,即 TCP-友好(TCP-friendly).

2.2.1 TCP 流量模型

TCP 的吞吐量与下面的参数有关:往返时间 t_{RTT},重传超时值 t_{RTO},分组大小 s,丢失率 p.式(1)[9]给出一个简化模型,近似模型 TCP 在稳定状态下的吞吐量.该简化模型没有考虑 TCP 超时.

式(2)[10]给出了一个更复杂的模型:b 是每个 ACK 应答的分组数量,W_m 是拥塞窗口的最大值,与式(1)不同,它考虑了 TCP 超时.在高丢失率网络中,式(2)更精确地模型化了 TCP 的行为.

$$T(t_{RTT}, s, p) = \frac{c \cdot s}{t_{RTT} \cdot \sqrt{p}} \quad c\text{是常量,通常取}1.5\sqrt{2/3}\ , \tag{1}$$

$$T(t_{RTT}, s, p) = \min\left(\frac{W_m \cdot s}{t_{RTT}}, \frac{s}{t_{RTT}\sqrt{\frac{2bp}{3}} + t_{RTO}\min(1, 3\sqrt{\frac{3bp}{8}})\, p(1+32p^2)}\right). \tag{2}$$

2.2.2 组播 TCP-Friendly 的定义

组播 TCP-Friendly 存在多种定义.文献[11]给出的定义是“非 TCP 流量的长期吞吐量不超过相同情况下 TCP 流量的吞吐量”.文献[12]对 TCP-Friendly 的定义更加严格:

(1) 单播 TCP-Friendly 的定义:在相同网络条件下,如果一个单播流量对其他并存 TCP 流量的长期吞吐量的影响(减少)不大于另外一个 TCP 流量对后者的影响,此单播流量被认为是 TCP-Friendly.

(2) 组播 TCP-Friendly:在发送端与每个接收端之间,如果流量具有单播流量 TCP-Friendly 的特性,此组播流量被认为是 TCP-Friendly.

对组播 TCP-Friendly 的定义还存在争论,某些定义提出应该允许组播流量使用比单播流量稍多的带宽,因为组播流量为多个接收端提供服务.文献[13]引入下面的公式来定义组播的 TCP-Friendly:

$$\alpha \cdot r_{TCP} \le r \le b \cdot r_{TCP}\ , \tag{3}$$

其中,r 表示瓶颈链路上的组播流的速率,r_{TCP} 表示相同情况下 TCP 流的速率,α 和 b 是流接收端数目的函数.当 $b=1$ 时,两种定义是相同的.

3 组播拥塞控制算法分类

组播拥塞控制算法存在多种分类方法[12].本节将讨论几种可能的分类方法,同时分析它们在实现可扩展性和 TCP-Friendly 上的优势和缺陷.

3.1 基于窗口与基于速率

对组播拥塞控制算法的一种可能的分类是依据拥塞控制的控制参数:窗口或速率.

3.1.1 基于窗口

基于窗口的拥塞控制算法在端系统维护拥塞窗口,通过拥塞窗口来控制未应答分组的数量.与 TCP 类似,发送一个数据分组会占用拥塞窗口的一部分,收到一个分组的应答后会释放占用的部分;拥塞窗口有空闲时,发送端可以继续发送数据.没有拥塞时,增加拥塞窗口;拥塞发生时减小拥塞窗口.

3.1.2 基于速率

基于速率的拥塞控制算法根据网络拥塞动态调整发送速率,分为两种:

(1) 简单的 AIMD 拥塞控制:这种算法模仿 TCP 的 AIMD 行为.它实现简单,但是容易导致速率在短期内出现与 TCP 类似的锯齿形.

(2) 基于模型的拥塞控制(equation-based congestion control):这种拥塞控制方式的主要目的是在对拥塞有反应的前提下,保持平滑的速率变化.基于模型的拥塞控制根据 TCP 流量模型(式(1)和式(2))调整速率,因此可以产生非常平滑的速率,适用于多媒体应用.基于模型的拥塞控制需要在拥塞反应速度和速率振荡之间作出权衡.

3.1.3 两种算法的比较

文献[14]指出,将拥塞窗口机制从单播扩展到组播时,为了最大化组播组吞吐量,发送端需要为每个接收端维护一个独立的拥塞窗口.另外一个值得注意的结论是:基于窗口的算法不需要估计接收端和发送端之间的

RTT,就能保证组播流量的 TCP-Friendly;而对于基于速率的算法,RTT 信息是必须的.

对基于窗口的算法,随着组规模的增大,为每个接收端维护拥塞窗口会导致发送端的拥塞控制任务变得非常复杂,从而降低了可扩展性.另外,发送端如果接收所有接收端的反馈,它将遭遇反馈爆炸.基于窗口的算法只需要采取与 TCP 类似的 AIMD 行为,就能比较容易地保证 TCP-Friendly.

基于速率的算法不需要维护拥塞窗口,但是发送端也需要从接收端收集的控制参数,例如分组丢失率和 RTT,如果不提供适当的反馈抑制机制,发送端也将遭遇反馈爆炸问题.基于速率的算法在实现 TCP-Friendly 时,需要获得 RTT 信息,然而在组播环境中,对 RTT 进行大规模测量非常困难.

3.2 单速率和多速率

组播拥塞控制的另一种分类依据是数据的发送方式:单速率和多速率.

3.2.1 单速率算法

在单速率算法中,发送端只用一种速率发送数据.整个组播组的吞吐量受瓶颈接收端(拥塞最严重的接收端)的限制.

3.2.2 多速率算法

多速率算法允许发送端使用多种发送速率,从而在接收端之间产生更灵活的带宽分配.

一种典型的实现方法是分层组播(layered multicast):数据被分为多个层,分别使用不同的组播组进行发送.接收端根据拥塞选择订阅适当的层.订阅的层越多,数据质量越高.以视频传输为例,增加层的数量可以提高视频质量;而在块数据的可靠传输中,订阅额外的层可以减少传输时间.

分层组播通过组管理和路由机制间接实现拥塞控制.为了增加拥塞控制的效率,接收端之间需要协同工作,尤其是拥有相同拥塞瓶颈的接收端.如果某些接收端检测到拥塞,取消当前订阅的最高层,但同一瓶颈后的其他接收端没有取消订阅,组播转发树不会进行剪枝,拥塞依然存在;如果接收端没有对加入操作进行同步,某些组成员则可能无法充分利用带宽.离开延迟是另外一个需要考虑的问题:从接收端发出离开消息到组播转发树完成剪枝,可能花费相当长的时间,ICMP-v1[15]可能需要 3 分钟,ICMP-v2[16]有所改善,也需要几秒钟的时间.

3.2.3 两种算法的比较

在单速率算法中,组播的吞吐量受瓶颈接收端的限制,限制了协议的可扩展性.文献[9]的研究表明,即使在网络情况比较理想时,随着组规模的增加,单速率组播拥塞控制也可能严重影响组播组的性能.单速率算法的优势是实现相对简单,不需要考虑对数据分层编码、决策同步等问题.

与单速率算法相比,分层组播的可扩展性较好,组的吞吐量受瓶颈接收端的限制较小,接收端可以选择合适的接收速率.分层组播的缺点是协议复杂,因为它利用底层路由机制来间接实现拥塞控制,频繁的加入/离开可能对路由协议造成较大的负担.为了提高拥塞控制的性能,需要解决接收端的决策同步问题.另外,考虑到组的数量增大会带来管理问题,分层组播不可能使用太多的组,从而造成层间的速率变化较大,拥塞控制的粒度比较粗糙,一方面降低了接收端对带宽的利用率,另一方面也为保证 TCP-Friendly 增加了难度.

4 组播拥塞控制协议

本节将介绍一些最近的组播拥塞控制协议.这些协议被分为单速率和多速率两大类,每一类又分成两个子类:基于窗口和基于速率.我们观察的重点是它们如何在可扩展性和 TCP-Friendly 之间作出权衡.

4.1 单速率组播拥塞控制协议

4.1.1 基于速率

很多基于速率的拥塞控制协议仅仅简单的模仿 TCP 的 AIMD 行为,而另外一些协议则根据 TCP 流量模型显式或隐式地调整发送速率.基于速率的协议在保证可扩展性上主要面临两个难题:LPM 问题,防止反馈爆炸;为了保证 TCP-friendly,协议需要解决大规模 RTT 测量问题.

TRAM(tree-based reliable multicast protocol)[17]是针对可靠组播设计的拥塞控制协议.协议预设两个变量:最小速率和最大速率,允许发送速率在这两个预设值之间平滑变化.TRAM 使用修复树(repair tree)限制数据重

传范围,修复树是一种动态的树形结构,用来收集接收端到发送端的反馈.修复树包括两类节点:一类是中间节点,也被称为修复节点(repair head),位于树的内部,它们的任务是缓存数据,聚合底层节点的拥塞信息,进行数据重传,当缓存数据超出一定限度时,它们将向上层节点发送拥塞信息.另一类节点是叶子节点,即接收端,如果在接收到一定数量的分组(ACK 窗口)之前经历了分组丢失,接收端向上游修复节点发送拥塞消息,位于相同 ACK 窗口的拥塞消息在上游修复节点被聚合.一旦收到拥塞消息,发送端将当前速率减半;如果没有收到拥塞消息,增加发送速率,增加量与当前速率和预设最大速率之间差值相关.

修复树避免了反馈爆炸问题,但 TRAM 中仍然存在 LPM 问题.另外,TRAM 的 TCP-Friendly 依赖于参数设定,例如最小、最大速率和 ACK 窗口.在某些网络情况下,这些预设值可能是不正确的.

TFMCC(TCP-friendly multicast congestion control)[18]是对 TFRC 协议[19]的扩展.TFRC 是一个针对多媒体应用的单播拥塞控制协议,TFMCC 将它扩展到组播中.与 TFRC 相同,TFMCC 也基于复杂的 TCP 流量模型[12]调整速率,接收端使用平均丢失间隔算法(average loss interval method)计算丢失率.协议还使用额外的机制防止接收端对单个丢失事件反应过于强烈,同时保证丢失率能迅速适应长期无分组丢失的情况.启动后,发送端立刻进入慢启动阶段,通过快速地增加速率获得公平共享带宽.第 1 个丢失事件发生后,TFMCC 终止慢启动.

TFMCC 给出了一种大规模测量 RTT 的方法:发送端在数据分组中携带时间标记(timestamp),接收端根据时间标记计算 RTT.接收端根据 RTT、丢失率和 TCP 流量模型,计算出速率并反馈给发送端.发送端从中选出一个瓶颈接收端作为代表,TFMCC 称之为 CLR(current limiting receiver),此后仅根据 CLR 的反馈来调整速率.CLR 减轻了 LPM 问题带来的影响.另外,为了防止反馈爆炸,协议使用指数加权随机时钟来抑制接收端的反馈数量.TFMCC 的主要缺点是,在起始阶段,接收端的 RTT 初始化需要耗费较长的时间.另外,如果选择了错误的代表,可能导致组播流量的 TCP-Unfriendly 行为.

4.1.2 基于窗口

基于窗口的单速率组播拥塞协议的主要评价目标是可扩展性,它面临几个主要的问题:首先,协议需要解决 LPM 问题;另外,协议需要合理地利用拥塞窗口机制,提高协议的性能;发送端也可能遭遇反馈爆炸问题.

文献[14]提出一种基于窗口的组播拥塞控制算法,每个接收端维护一个拥塞窗口,采用与 TCP 类似的方法调整窗口.接收端根据窗口大小和已接收分组的数量,计算出可接收分组的最大序列号.接收端将自己的最大序列号通过树形结构聚合(上游节点从中挑出最小值),反馈给发送端,从而避免了反馈爆炸.当聚合后的信息到达发送端后,发送端就得到了发送分组的最大序列号.通过这些机制:在每个接收端维护拥塞窗口,使用树形结构对拥塞信息进行聚合;协议避免了 LPM 问题.

文献[14]的研究为基于窗口的组播拥塞控制提出了理论背景.下面我们将介绍一些具体的协议.

RLA(random listening algorithm)[14]引进了一些增强措施,将 TCP SACK 扩展到组播中.发送端为每个接收端保存 RTT,并计算它们的拥塞概率.发送端通过不连续的确认或超时来探测分组丢失,并记录拥塞概率较高的接收端的数目 n.探测到拥塞后,如果满足下列两种情况中之一,发送端窗口减半:(1) 距离上一次窗口减半超过一段时间;(2) 正态分布的随机数 π 小于或等于 $1/n$.如果一个分组得到所有接收端的确认,发送端增加拥塞窗口.RLA 还包含一个类似 TCP 快速恢复的重传方案.

RLA 的关键思想在于使用随机监听(random listening),发送端随机地对来自接收端的拥塞信号产生反应,通过这种机制,RLA 避免了 LPM 问题.另外,RLA 的窗口调整机制保证协议能获得统计上的长期公平性.RLA 的缺陷是没有提供反馈抑制机制,限制了协议的可扩展性.

MTCP(multicast TCP)[20]是一个针对可靠组播的拥塞控制协议.协议生成一棵多级的逻辑树,逻辑树的树根是发送端,树的其他节点是接收端.在逻辑树中,父节点负责处理子节点的反馈和丢失分组的重传.子节点向父节点发送两种反馈:ACK 或 NACK.每个父节点维护两个窗口:拥塞窗口和传输窗口.对拥塞窗口的管理与 TCP 类似;传输窗口负责跟踪还没有被子节点确认的分组.

每收到一个 ACK,节点向父节点发送一个拥塞摘要(congestion summary).拥塞摘要包括最小拥塞窗口和最大传输窗口.发送端可以发送最小拥塞窗口和最大传输窗口之间的数据.

MTCP 使用逻辑树进行丢失重传和反馈聚合,从而避免了 LPM 和反馈爆炸问题.每个节点转发其子节点瓶

颈链路信息到自己的父节点.因此,接收端将收到全部瓶颈链路信息,不仅仅是无关的分组丢失信息.通过采用与 TCP 类似的拥塞窗口控制机制,MTCP 可以较好地保证 TCP-Friendly.MTCP 的主要缺点是协议复杂,需要建立逻辑树,每个节点都需要执行缓存、修复和拥塞监控等功能.而且 MTCP 要求所有接收端对每个分组进行确认,限制了协议的可扩展性.

与 TFMCC 类似,PGMCC(pragmatic general multicast congestion control)[21]也是一种基于代表的拥塞控制协议.协议从组成员中选择一个瓶颈接收端作为代表,负责对每个接收到的分组进行确认,这种机制允许在发送端和代表之间使用与 TCP 类似的拥塞控制.在基于代表的算法中,拥塞控制和分组修复彼此之间是互相独立的,因此这种方法不仅适用于不可靠传输,也可以与可靠性机制结合起来.

基于代表的最大问题是如何挑选出合适的代表.在 PGMCC 中,每个接收端使用 TCP 流量模型计算出期望的速率,由 NACK 将速率期望值携带到发送端;发送端将拥有最低速率期望的接收端选为代表.

基于代表的技术使 PGMCC 避免了 LPM 问题和反馈爆炸问题.如果选择了正确的代表,PGMCC 可以很好地保证 TCP-Friendly.问题在于 PGMCC 仅仅根据接收端粗略的速率估计来选择代表,如果选择了错误的代表,可能导致相对 TCP 的不友好.

4.2 多速率组播拥塞控制协议

下面介绍比较典型的多速率组播拥塞控制协议.

4.2.1 基于速率

基于速率的多速率组播拥塞控制有一种典型的实现:分层组播.与单速率协议相比,分层组播拥有较好的可扩展性.

RLM(receiver-driven layered multicast)[22]是在分层组播方面最早的工作之一.RLM 是为视频传输设计的,协议使用接收端驱动(receiver-driven)机制提高协议的可扩展性.发送端将视频数据分为多个层,每层使用独立的组播组发送,接收端订阅第 1 层,开始接收数据.一段时间后,如果没有经历分组丢失,它通过周期性地加入试验(join experiment)订阅下一层;如果经历分组丢失,接收端取消最新订阅的层.

RLM 存在很多问题.它没有考虑 TCP-Friendly,也没有考虑接收端之间的加入/离开同步问题.另外,加入试验失败可能增加其他接收端的拥塞.

Vicisano 提出了 RLC(receiver-driven layered congestion control)[23],对 RLM 作出了很大的改进.

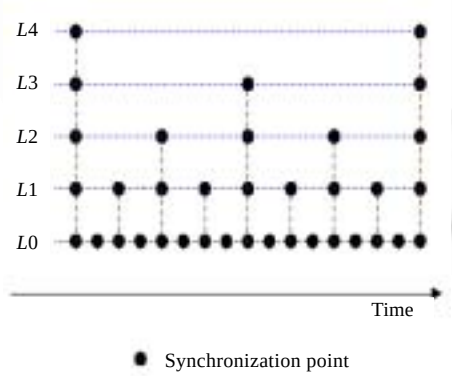

Fig.2 SPs in RLC

图 2 RLC 中的同步点

为了减少编码复杂度和层的数量,RLC 提出一种层分配方案:按指数递增分配每层的带宽,加入层的等待时间也呈指数增长.一旦发生分组丢失,接收端立即取消最新订阅的层,从而使接收速率减半;在没有分组丢失的情况下,接收速率随加入新层的等待时间成比例增加.通过这种分层方案和等待时间机制,RLM 模仿了 TCP 的 AIMD 行为.

为了改善接收端之间的同步问题,RLC 要求接收端只能在同步点(synchronization point,简称 SP)加入新层(如图 2 所示).每层的 SP 数量呈指数递减.这样,刚加入的接收端可以在一段时间后赶上订阅了较高层的接收端,从而使相同瓶颈后的接收端可以同步地做出加入/离开决策.RLC 在相邻 SP 之间定义了一段突发周期,在这段时间内,发送端将每层的发送速率加倍.接收端探测到拥塞,就不会进行加入试验,减轻了加入试验失败产生的额外的拥塞.

RLC 仍然存在缺陷.首先,它的速率调整粒度比较粗糙,指数分层只允许接收速率加倍和减半,使接收端无法充分地利用带宽,同时也可能导致 RLC 流量间的不公平行为.发送端在决定速率时没有考虑 RTT,在高 RTT 情况下可能出现相对 TCP 的不友好竞争.另外,RLC 要求数据支持分层.

为了改善 RLC 的缺陷,Byers 等人提出了 FLID-DL (fair layered increase/decrease with dynamic layering)[24].

协议在发送端使用 Digital Fountain[25]编码,接收端接收到一定数量的独立分组后就能将其还原成原始数据.因为无须保证特定包的可靠传输,分层方案可以非常灵活.

FLID-DL 引入了动态分层(dynamic layering)方案,减少了加入和离开延迟.在动态分层中,层占用的带宽随时间减少.接收端必须定期加入新层来维持接收速率.只要接收端不加入新层就可以减少接收速率;但是为了增加接收速率,接收端需要加入多个新层.经过一段时间后,某层的速率可能降为 0,FLID-DL 定义了一个层的静止期,经过静止期后,速率降为 0 的层将被重用.从而减少了层的数量,减轻了组管理和路由协议的压力.

FLID-DL 还使用公平分层增加/减少(fair Layered increase/decrease)方案对动态分层进行补充,这种方案使用 TCP 流量模型[11],使 FLID-DL 能保证 TCP-Friendly.协议保留了 RLC 的 SP 机制,但是它没有采用突发周期来探测带宽.它使用了概率增加信号(probabilistic increase signals)机制,概率较大时,接收端才会订阅新层.概率选择的依据是 TCP 流量模型,从而保证协议的 TCP-Friendly.

FLID-DL 和 RLC 相比,有了相当大的改进:它避开了离开延迟过长的问题,在带宽分配上也非常灵活.但是与 RLC 相同,FLID-DL 没有考虑 RTT,在某些网络情况下可能表现为 TCP-Unfriendly.另外,FLID-DL 的加入/离开操作发生得更加频繁,给组播路由协议造成很大的开销.

以上的方案有一个共同点,即通过分组丢失率检测带宽.在具有极长队列丢弃缓冲管理(longest queue drop buffer management)的公平调度器(fair scheduler)网络中,可以通过分组对(packet-pair)来探测带宽,根据两个相邻分组的接收时间间隔可以计算出流的可用带宽.PLM(packet-pair receiver-driven cumulative layered multicast protocol)[26]使用了这种分组对机制.协议还通过额外的时钟机制,保证接收端的加入/离开同步.

PLM 不需要层之间的特定的带宽分配.公平调度器网络有很多特征便于拥塞控制协议的设计,也可提高现有协议的性能.但要 Internet 在短期内不可能大体上实现基于公平的调度.

5 未来的研究方向

近年来,组播拥塞控制方面的研究取得了很大的进展,然而还有很多问题还没有得到解决.

一个需要进一步研究的方向是对 TCP 流量模型的改善.现有的模型对网络提出一些假设,在某些网络环境下可能无法保证 TCP-Friendly.例如,有的模型假设速率对 RTT 和丢失率不会产生影响.这种模型在 Internet 中可能工作得很好,如果将模型用于少数流量共享瓶颈链路,这种情况下,速率的变化很容易改变瓶颈链路的使用情况,影响模型中的参数,反过来又对速率产生影响.这种反馈环可能导致模型的结果与真实情况不符.

一些组播拥塞控制协议使用基于树的技术,接收端被组织到一个树型结构中,例如 TRAM 协议中的修复树.为了简化协议设计和实现,这些协议使用底层组播路由协议产生的树型拓扑,但拥塞控制属于传输层协议,在很多情况下无法直接获得网络层的路由信息.而且,是否应将组播拥塞控制放到网络层,与路由结合,这是一个值得争议的问题.

前面,我们分别讨论了单速率组播拥塞控制和分层组播,分层组播可以不受瓶颈接收端的限制,在可扩展性上比单速率算法更占优势.分层组播在组播路由紧密模式(PIM-DM,DVMRP)下工作得很好,但是 Internet 的层次化使紧密模式不适合大规模应用,相比之下,稀疏模式组播路由协议(PIM-SM,CBT)更适合 Internet.但是在稀疏模式中,组播树可能由于网络的拥塞,出现共享树到源树的互相转换,在这个过程中,分层组播的频繁加入/离开操作可能造成路由协议进入不稳定状态.如何使稀疏模式与分层组播拥塞控制机制更好地结合,是一个需要研究的问题.

IP 组播技术在 Internet 中一直无法得到广泛的应用,一个主要的原因在于组播对网络的要求较高,很多技术都要求路由器的支持,在现有的网络中不容易得到普及.而且传统的 IP 组播模型比较学术化,在商业应用上遇到了很多问题,这导致新的组播模型(例如 EXPRESS[27])的出现,不少研究人员开始基于这类组播模型研究组播拥塞控制.

另外,最近提出的应用层组播也是一个很有前途的研究方向.应用层组播的基本思路是在应用层建立发送端和接收端之间的逻辑树,也就是由应用层负责组播路由.这种技术将组播的复杂性从网络转移到端系统,这种策略符合 Internet 的设计思想[28],而且它直接使用传统的单播拥塞控制,避开了组播拥塞控制的难题.应用层组

播最主要的问题是如何合理地建立逻辑树,这类似于传统的组播 QoS 路由问题.

6 结 论

在本文中,我们总结了组播拥塞控制领域的研究进展.首先介绍了组播与拥塞控制的基本概念,包括拥塞控制对 IP 组播的重要性以及组播拥塞控制协议的设计难点.一种组播拥塞控制协议不可能满足所有的应用需求,相对于需求的多样性,组播拥塞控制协议在评价目标上可能存在很大的差异.我们相信,在众多的评价目标中,协议的可扩展性和 TCP-friendly 对组播应用的发展更为重要.

考虑到组播拥塞控制算法的不同特性,存在多种可能的分类方法.本文讨论了其中的两种分类方法,围绕可扩展性和 TCP-friendly 这两个评价目标,对不同的算法进行分析和比较.通过这些比较,我们可以得出结论:两个评价目标在实现中存在一定的矛盾,组播拥塞控制协议需要根据实际情况进行权衡.

致谢 清华大学计算机科学与技术系网络技术研究所的林闯教授对本文的工作给予了细心的指导,任丰源讲师、徐恪讲师、刘莹博士后和章淼同志对本文的完成提出了很多有益的建议,在此一并表示感谢.

References:

[1] Gevros, P., Crowcroft, J., Kirstein, P., *et al*. Congestion control mechanisms and the best effort service model. IEEE Network, 2001,15(3):16~26.

[2] Jain, R., Ramakrishnan, K.K., Chiu, Dah-Ming. Congestion avoidance in computer networks with a connectionless network layer. Technical Report, DEC-TR-506, Digital Equipment Corporation, 1988. http://www.cis.ohio-state.edu/~jain.

[3] Deering, S. Multicast routing in a datagram internetwork [Ph.D. Thesis]. Stanford University, 1991.

[4] Sahasrabuddhe, L.H., Mukherjee, B. Multicast routing algorithms and protocols: a tutorial. IEEE Network, 2000,14(1):90~102.

[5] Mankin, A., Romanow, A., Brander, S. *et al*. IETF criteria for evaluating reliable multicast transport and application protocols. RFC 2357, 1998.

[6] Erramilli, A., Singh, R.P. A reliable and efficient multicast protocol for broadband broadcast networks. In: Floyd, S., ed. Proceedings of the ACM SIGCOMM. New York: ACM Press, 1987. 343~352.

[7] Bhattacharyya, S., Towsleu, D. The loss path multiplicity problem in multicast congestion control. In: Doshi, B., ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 1999. 856~863.

[8] Chaintreau, A., Baccelli, F., Diot, C. Impact of network delay variation on multicast sessions with TCP-like congestion control. In: Ammar, M., ed. Proceedings of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1133~1142.

[9] Floyd, S., Fall, K. Promoting the use of end-to-end congestion control in the internet. IEEE/ACM Transactions on Networking, 1999,7(4):458~472.

[10] Padhye, J., Firoiu, V., Towsley, D., *et al*. Modeling TCP throughput: a simple model and its empirical validation. In: Oran, D., ed. Proceedings of the SIGCOMM. Vancouver: ACM Press, 1998. 303~314.

[11] Floyd, S., Handley, M., Padhye, J. A comparison of equation-based and AIMD congestion control. 2000. http://www.aciri.org/floyd/ papers.htmls.

[12] Widmer; J., Denda, R., Mauve, M. A survey on TCP-friendly congestion control. IEEE Network, 2001,15(3):28~37.

[13] Wang, H.A., Schwartz, M. Achieving bounded fairness for multicast and TCP traffic in the internet. In: Black, R., ed. Proceedings of the ACM SIGCOMM. Vancouver: ACM Press, 1998. 81~92.

[14] Golestani, S.J., Sabnani, K.K. Fundamental observations on multicast congestion control in the internet. In: Honeyman, P., ed. Processing of the IEEE INFOCOM. Ottawa: IEEE Communications Society, 1999. 990~1000.

[15] Deering, S. Host extensions for IP multicasting. STD 5, RFC 1112, 1989.

[16] Fenner, W. Internet group management protocol, Version 2. RFC 2236, 1997.

[17] Kadansky, M., Chiu, D., Wesley, J., *et al*. Tree-Based reliable multicast (TRAM). INTERNET DRAFT draft-kadansky-tram-02.txt, Work in Progress. 2000.

[18] Widmer, J., Handley, M. Extending equation-based congestion control to multicast applications. In: Floyd, S., ed. Proceedings of the ACM SIGCOMM. San Diego: ACM Press, 2001. 275~286.

[19] Floyd, S., Handley, M., Padhye, J., *et al*. Equation-Based congestion control for unicast applications. In: Floyd, S., ed. Proceedings of the ACM SIGCOMM. Stockholm: ACM Press, 2000. 43~56.

[20] Rhee, J., Balaguru, N., Rouskas, G. MTCP: scalable TCP-like congestion control for reliable multicast. In: Doshi, B., ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 1999. 1265~1273.

[21] Rizzo, L. PGMCC: a TCP-friendly single-rate multicast congestion control scheme. In: Floyd, S., ed. Proceedings of the ACM SIGCOMM. Stockholm: ACM Press, 2000. 17~28.

[22] McCanne, S., Jacobson, V., Vetterli, M. Receiver-Driven layered multicast. In: Deering, S., ed. Proceedings of the ACM SIGCOMM. Stanford: ACM Press, 1996. 117~130.

[23] Vicisano, L., Rizzo, L., Crowcroft, J. TCP-Like congestion control for layered multicast data transfer. In: Charny, A., ed. Proceedings of the IEEE INFOCOM. San Francisco: IEEE Communications Society, 1998. 996~1003.

[24] Byers, J., Frumin, M., Horn, G., *et al*. FLID-DL: congestion control for layered multicast. In: Fdida, S., ed. Proceedings of the 2nd International Workshop on Networked Group Communication (NGC 2000). Palo Alto: ACM Press, 2000. 71~81.

[25] Byers, J., Luby, M., Mitzenmacher, M., *et al*. A digital foundation approach to reliable distribution of bulk data. In: Oran, D., ed. Proceedings of the ACM Sigcomm. Vancouver: ACM Press, 1998. 56~67.

[26] Legout, A., Biersack, E.W. PLM: Fast convergence for cumulative layered multicast transmission schemes. In: Drushel, P., ed. Proceedings of the ACM Sigmetrics. Santa Clara, CA: ACM Press, 2000. 13~22.

[27] Holbrook, H.W., Cheriton, D.R. IP multicast channels: EXPRESS support for large-scale single-source applications. In: Chapin, L., ed. Proceedings of the ACM Sigcomm. Cambridge: ACM Press, 1999.

[28] Saltzer, J., Reed, D., Clark, D. End-to-End arguments in system design. ACM Transactions on Computer Systems, 1984,2(4):195~206.

A Survey on Multicast Congestion Control*

SHI Feng, WU Jian-ping

(*Institute of Computer Network Technology, Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

E-mail: shf@csnet1.cs.tsinghua.edu.cn

http://netlab.cs.tsinghua.edu.cn

Abstract: The problem of congestion control must be solved before the large-scale deployment of multicast. There are two important goals in multicast congestion control protocols: scalability and TCP-friendly. In this paper, the two goals are introduced, a survey on multicast congestion control is presented. Some recent protocols are discussed. The orientations of the future research are also given.

Key words: multicast; congestion control; scalability; TCP-friendly

* Received January 29, 2002; accepted June 25, 2002

Supported by the National Natural Science Foundation of China under Grant Nos.90104002, 69725003; the National High-Tech Research and Development Plan of China under Grant No.2001AA121013