

可靠组播拥塞控制最新研究进展^{*})

Overview of Reliable Multicast Congestion Control(RMCC)

任立勇 卢显良

(电子科技大学计算机学院 成都610054)

Abstract Congestion control, which is one of the important research points of reliable multicast protocol, has become a active field in the computer network and communication areas. In the paper the principal problems of congestion control for reliable multicast are analyzed first, and then merits and limitation of methods for addressing these problems are demonstrated detailedly. At last the future work is pointed out.

Keywords Reliable multicast, Congestion control, Scalability, TCP-friendly

1. 引言

可靠组播是一种为优化使用网络资源而产生的技术,通常使用在一对多工作方式下的应用中,如软件分发、视频会议、交互式仿真等。近年来,随着这些应用的开展,对可靠组播协议的研究已成为网络技术领域的研究热点。

可靠组播的引入虽然节约了网络资源,减少了网络管理费用,但同时也给 Internet 带来了许多潜在的问题^[1,2],如:组播安全问题、拥塞控制问题、组播路由问题、错误恢复问题等,而其中可靠组播的拥塞控制是所有问题中最重要的、最难解决的、最令人关注的问题之一。如果可靠组播应用不能对网络拥塞作出正确响应,将会给 Internet 带来比单点投递(Unicast)应用产生的拥塞更为严重的影响。这主要因为:(1)可靠组播投递流可能沿着其多点投递树广泛分布于整个 Internet;(2)可靠组播流的接收者具有异构性,每个接受者的处理能力不同,其分组投递路径也可能有不同的带宽和差错特性。(3)可靠组播发送方需处理比单点投递多得多的反馈报文,对这些报文如果不加以处理,不仅可能淹没发送方,而且它们本身也是对网络资源的一种极大的开销。

由于不同的组播应用对服务质量(QoS)有不同要求,试图设计一种类 TCP(TCP-Like)的、通用的拥塞控制算法显然是不可能的。为此,近年来出现了大量的针对不同应用的组播拥塞控制机制与算法,这些算法的出发点与解决问题的重点不同,方法也就不同,它们往往相互对立甚至是矛盾的。这就需要在适当的时候对这些方法加以总结与回顾,指出各自的优缺点,并提出以后应重点解决的问题与研究方向。尽管已经有文献做了这部分工作^[1-3],但它们完成的时间较早,不能充分反映组播拥塞控制这两年的最新研究成果。本文正是基于上述原因而提出的。

文章首先分析了可靠组播拥塞控制中应重点解决的问题,然后详细论述了解决这些问题的方案及已有的各种算法优缺点,最后对这些方案进行了总结,并展望了可靠组播拥塞控制领域未来的研究方向。需要指出的是,本文无意罗列可靠组播拥塞控制领域的所有文献,仅对其中几种较为典型的方法进行了剖析。

2. 组播拥塞控制的主要问题

绝大多数可靠组播协议是建立在 IP 层的 Multicast 之上,而 IP 层的组播通信只提供尽力型服务,不保证报文的可靠传输。可靠组播拥塞控制通过接收者或其它网络设备提供的显示或隐式信息来推断网络状态、调节发送速率、重传丢失分组,以达到报文的可靠投递。与单点投递应用不同的是,不同的组播应用对组播拥塞控制的要求不同,组播接收者一般数量巨大,网络特性也千差万别,这些都给设计组播拥塞控制算法带来许多困难^[4],主要表现在以下几个方面。

·可伸缩性 是组播拥塞控制最核心的问题。组播用户往往数量巨大,而且可能遍布于整个 Internet,因此可靠组播需处理较多数量的反馈信息,如确认报文(ACK)、否定报文(NAK)、状态报文等,如果不对这些报文进行正确处理,将导致两种经典的问题:归零(Drop-to-Zero)与 ACK 风暴(ACK Implosion)(此处的 ACK 包括肯定确认(ACK)和否定确认(NAK))。如图1所示,网络拥塞一般表现为三种情况^[5]:独立型拥塞,共享型拥塞,混合型拥塞。图中 S 表示发送方, R 表示接收者,虚线表示链路拥塞。当发生独立型拥塞时,每个因拥塞而丢失报文的接收者将向发送方反馈不同的拥塞信息(EI),发送方将对每个拥塞信息作出响应,降低发送速率。当独立型拥塞较多时,最终将导致发送速率归零(Drop-to-Zero)。当发生共享型拥塞时,拥塞链路下游的每个接收者都将向发送方反馈丢失信息(NAK),如果下游接受者数目较多时,此时将发生 ACK 风暴(ACK Implosion)。当发生混合型拥塞时,上述两种现象均会发生。为避免归零与 ACK 风暴,现有的组播拥塞控制一般采用了两种方法:ACK 聚集(ACK Aggregation)与 ACK 抑制(ACK Suppression)。

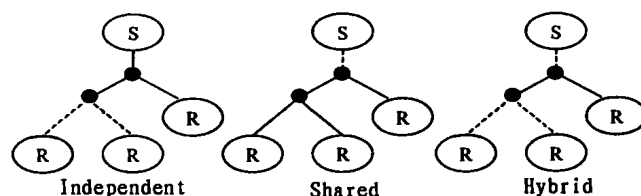


图1 独立拥塞与共享拥塞

^{*})本文得到国家九五科技攻关项目基金资助,信息产业部生产发展基金资助,任立勇 博士研究生,主要研究方向为网络服务质量、新型网络协议、分布式操作系统等,卢显良 教授,博士生导师,主要研究方向为操作系统与网络应用技术。

·异构性 主要表现在组播的接收者处理能力不同与报文到达不同接收者经历的链路的网络特性(如带宽、延迟等)不同,更为重要的是,链路的异构性在组播会话期间常常是动态变化的,这就更增加了拥塞控制的难度。为保证报文的可靠投递,发送方需要及时地获取每个接收者或其它网络设备的反馈信息,调整发送速率、重传丢失报文。因此,可靠组播拥塞控制往往选择最慢速接收者的接收速率来发送报文,这在很大程度上浪费了网络资源,降低了组播通信的效率。为解决这类问题,现有的拥塞控制一般采用了两种方法:层次化传输与结构化组播成员,第4节我们会详细介绍这两种方法的适用范围与优缺点。

·公平性 是组播拥塞控制机制能否得到广泛应用的基础,也是每个可靠组播拥塞控制算法所追求的设计目标之一^[5]。可靠组播拥塞控制的公平性主要表现在以下两方面。一是协议间公平性(Inter-Protocol Fairness),主要是指当可靠组播流与 TCP 流共享瓶颈链路时,能公平占用网络带宽,即可靠组播流是 TCP 友好的(TCP-Friendly)。Internet 之所以能健康稳定地运行,很大一部分应归功于 TCP 拥塞控制对网络状态的自适应性与公平性,并且,目前的 Internet 中超过 80% 的数据流使用 TCP 作为可靠传输协议。如果可靠组播拥塞控制对拥塞指示(EI)的响应机制与 TCP 不同,可能导致它们不能公平共享网络带宽,而使 TCP 流饿死,最终导致整个 Internet 崩溃。可靠组播拥塞控制公平性另一方面是协议内公平(Intra-Protocol Fairness)。如前所述,在可靠组播通信中,发送方的发送速率一般是由最慢速接收者的接收速率决定的,这显然对具有大处理能力与高链路带宽的接收者是不公平的。但幸运的是,可靠组播用户一般对投递差错要求较高,而对数据延迟则没有严格的要求。在处理上述两种公平问题时,组播拥塞控制一般采用类 TCP(TCP-like)响应机制来提供协议间公平,具体主要分为两种:基于速率响应机制与基于窗口响应机制。而采用基于代理(Agent-Based)与前向纠错(FEC)等其它方式来提供协议内公平。

除以上提到的三个主要问题外,设计一个有效的组播拥塞控制算法,还需要考虑以下问题:拥塞度量问题、拥塞响应度问题、层次传输时层数选择问题等。TCP 将拥塞控制与错误恢复耦合在一起,将报文丢失作为网络拥塞的标志之一,而在组播通信中则不能将每个报文丢失作为网络拥塞指示,否则极易造成发送速率振荡。可靠组播拥塞控制通常将“一段”时间内的报文丢失率作为网络拥塞度的参数之一。TCP 拥塞控制能在一个 RTT 或更短的时间内对网络拥塞状况作出响应,而可靠组播拥塞控制由于接收者数量大、范围广,因此常常需要几个甚至十几个 RTT 才能对拥塞作出响应,因此,设计算法时常在拥塞响应度与伸缩性间作折衷选择。若采用多层速率传输数据,则必须根据具体应用类型决定层数。

3. 拥塞控制模型:基于发送方与基于接收方的控制

可靠组播拥塞控制模型主要分为两种:基于发送方的拥塞控制与基于接收方的拥塞控制。

在基于发送方的可靠组播拥塞控制^[4,6,7]中,每个接收者向发送方发送反馈信息,这些反馈信息包括 ACK、NAK 或状态报文。发送方根据反馈信息,重传丢失报文,并确定组播树中的最慢速接收者速率,以该速率调节发送速率。采用这种模型的好处有:(1)发送方能全面了解整个组播树的网络状态,以便准确地控制数据的可靠投递。(2)对于以多层次方式传输

数据的应用中,发送方能根据网络状况及时、动态地调整传输数据的层次,有效地利用了网络带宽。(3)在开环(Opened-Loop)控制中,发送方能根据网络状况及时调整编码参数(如前向纠错 FEC)。但基于发送方的模型有一个致命的缺点,即反馈风暴(Feedback-Impllosion)。随着组播接收者的规模变大,发送方接收到的反馈报文急剧增加,从而导致发送方处理开销急剧增加,最终崩溃。为解决这个问题,现有的基于发送方可靠组播拥塞控制一般采用两种方法:反馈聚集与反馈抑制,但这不可避免地会增加算法的复杂性与额外的网络设备投入。

在基于接收方的可靠组播拥塞控制^[8~10]中,每个接收者根据接收报文情况推断网络状态,然后将反馈报文发送回发送方和上一级接收者(或代理)。现有的基于接收方的可靠组播拥塞控制算法,一般应用在连续媒体流和大批量数据分发的层次传输中。采用基于接收方的主要原因有^[8]:(1)在组播会话中,随着用户的增加,速率调节与维护组播成员的相关算法的复杂性与开销急剧增加,如果这些任务都由发送方来完成,必然要求发送方具有较强的处理能力与快速发应能力,这在很大程度上降低了拥塞控制的伸缩性;(2)拥塞控制决策不一定需要完全由发送方来完成,如接收者可以根据网络情况调节自身接收速率,加入或离开某个组,甚至退出会话;(3)可靠组播的报文重传可能在本地域完成,而不需要发送方参与;但采用基于接收方模型的可靠组播模型也有许多缺点,如:(1)大多数基于接收方的协议主要通过 NAKs 来确定网络拥塞度,如果发送方没有收到反馈 NAKs,则有可能错误地推断网络无拥塞和拥塞已清除。例如,有研究表明大多数拥塞将持续几秒到几分钟,在这段时间内,接收者无法接收到任何报文,也就不向发送方发送反馈报文;(2)基于接收方的协议中采用的随机 NAK 抑制技术要求每个接收者执行复杂度为 $O(n^2)$ 的链路往返时间计算开销,这在一定程度上限制了协议的伸缩性;(3)大多数基于接收方的分层传输要求原始数据具有易分层编码的自然属性,同时,在接收方要求预留大量的缓存用来同步接收的多层数据。

4. 可靠组播成员的逻辑关系

在可靠组播会话中,发送方和接收方的逻辑关系与拥塞控制的效率紧密相关。通常可将可靠组播协议分为以下两类:平面型与基于结构型,如图2所示。

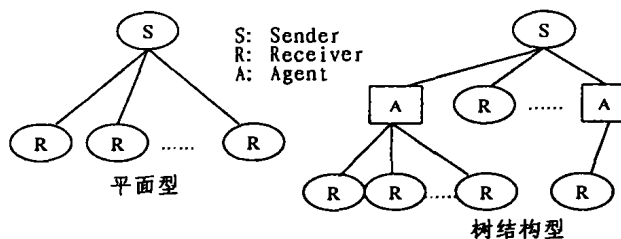


图2 可靠组播发送方与接收方逻辑关系

平面型 这种类型中接收方以单点投递和组播的方式将反馈信息发送到发送方和其他接收者。其好处主要有:(1)协议简单,不需其他网络设备(如路由器)支持;(2)由于反馈不需中间处理,发送方能快速了解网络拥塞状况,并作出响应。但平面型可靠组播有一个致命的缺陷,即很容易在发送方形成反馈风暴,协议伸缩性差。为防止反馈风暴,通常采用的方案有:

(1) 基于定时器方案^[11]。每个接收者需要等待一段随机时间(直至定时器超时),才能将其反馈信息发送给发送方,并同时以组播的方式发送给其他接收者。在等待期间,若一个接收者收到一个从其他接收者来的对同一报文的反馈信息,则取消发送自己的反馈报文。

(2) 基于轮询(Poll)方案^[12]。发送方定时向接收者轮询网络拥塞状况,收到轮询的接收者有权限向发送方反馈信息。

(3) 基于概率方案。接收方发送反馈信息前,首先计算能发送反馈的概率,若大于某个指定的阈值,则发送反馈。

(4) 基于代表的方案^[13]。假设有 N 个接收者,指定 K ($K \ll N$) 个符合一定条件的接收者作为代表,只有这 K 个代表才能发送反馈信息。

上述方案虽然能在一定程度上防止反馈风暴,但并没有完全解决拥塞控制的伸缩性问题^[1],同时(2)、(3)、(4)方案又带来了另一个问题,即反馈信息不能完整、准确地反映网络目前的真实状况。

基于结构型 这种类型的组播协议将接收者组成一种逻辑结构,两种较为普遍的结构是:环结构与树结构。

(1) 文[14]中,将可靠组播成员构建成若干个逻辑环,每个环中有一个令牌绕环传播,只有获得令牌的接收者才能发送反馈信息。

(2) 另一种更常见的结构是树结构^[4,8,9,15]。这种方案将接收者按照一定的条件(如位置、带宽等)组织成树状结构,叶子节点是接收者,根节点是发送方,中间节点可以是接收者,也可以是其他网络设备(如路由器等)。每个节点将它的反馈报文发送给父节点,由父节点对反馈信息进行聚集,再发送给它上一级节点。

树结构方案采用得最广,其原因主要有以下几点:a)由于中间节点能对反馈进行聚集,因此树结构能有效地避免反馈风暴;b)中间节点(代理, Agent)缓存未被确认报文,以便为其子孙节点重传丢失报文。因此,局部的网络拥塞不会影响到整个组播域,提高了可靠组播的效率;c)构造组播树的过程可由 IP Multicast 捎带完成,而无需增加另外的协议。但树结构的可靠组播拥塞控制也带来了一些负面影响:a)由于中间节点对反馈进行处理的开销,增加了反馈报文到达发送方的延迟,从而使发送方不能及时对网络拥塞状况作出响应;b)由于采用了丢失报本地重传,因此有可能导致发送方不能掌握整个组播会话组的完整情况,从而作出错误的速率调节决策;c)大多数树结构方案中采用固定位置、固定数目的代理,虽然这降低了协议配置与拥塞控制的复杂度,但不能动态地、自适应地随接收者位置、数目及网络拥塞状况变化而变化。S. Shi^[16]等人提出了动态确定最差链路并选择代理的方法,在解决上述第三个问题方面迈出了重要的一步。

5. 反馈控制机制

可靠组播拥塞控制可分为闭环控制与开环控制。发送方从接收方接收反馈信息以进行拥塞控制,称为闭环控制,反之则称为开环控制。开环控制一般应用多媒体数据组播应用中,如采用 FEC 编码并传输数据^[17],在发送方将每 k 个原始数据报文编码加上 h 个校验报文进行投递,接收者只要收到 $k+h$ 个报文中的任意 k 个报文就能正确恢复原始数据。

目前大多数可靠组播拥塞控制采用闭环控制^[6,17],接收方向发送方发送反馈报文,反馈报文的类型有:肯定确认报文(ACKs)、否定确认报文(NAKs)、选择确认报文(SACKs)。无

论采用哪种类型,都需要考虑反馈风暴问题。第4节已经介绍了采用反馈抑制和反馈聚集的方法来避免反馈风暴问题。除此以外,在反馈控制方面,部分可靠拥塞组播算法还采用了以下两种策略,如图3所示。

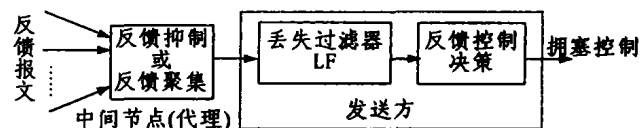


图3 可靠组播反馈控制机制

(1) 当可靠组播用户数特别大时,经过中间节点处理后到达发送方的反馈报文数目仍然有可能淹没发送方。P. Thapliyal 等人^[18]提出在接收方添加一组丢失过滤器,按照一定策略,将一些重复的、无意义的反馈报文丢掉,同时,建议在接收方用丢失事件(Loss-Events, LEs)取代丢失指示(Loss-Indications, LIs)表示报文丢失率,从而极大地减少反馈报文的数目。

(2) 反馈控制决策是发送方根据丢失指示形成拥塞指示(Congestion Indications, CIs)的策略问题。反馈控制决策影响着拥塞控制机制对网络拥塞状态的响应速度,并最终决定着对拥塞响应的准确度。可靠组播拥塞控制中主要有两种比较典型的反馈控制决策:a)基于报文丢失率阈值检查的反馈控制决策,即当丢失报文数目大于指定阈值时,则认为网络拥塞,否则认为网络正常;b)基于代表的反馈控制决策,即从 N 个丢失指示中选择 K 个作为拥塞指示。

6. 速率调节算法

拥塞控制的最终形式体现在发送方根据网络的拥塞状态调节其发送速率。可靠组播拥塞控制的速率调节主要分为两种:基于速率的调节与基于窗口的调节。

基于速率的调节^[4,10,12] 为实现与 TCP 公平共享网络带宽,基于速率的可靠组播拥塞控制调节算法一般采用稳定状态下 TCP 速率响应函数作为其响应函数。(1)、(2)式分别为两种 TCP 响应函数:

$$T \leq \frac{1.5\sqrt{2/3} * M}{RTT * \sqrt{p}} \approx \frac{1.22 * M}{RTT * \sqrt{p}} \quad (1)$$

$$T = \frac{M}{RTT \sqrt{2p/3} + T_{\text{pro}}(3\sqrt{3p/8})p(1+32p^2)} \quad (2)$$

在基于接收方的模型中,每个接收者分别测量路径 RTT 与分组丢失率 p ,然后根据响应函数计算吞吐量,将其作为反馈发送回发送方,发送方综合“所有”接收者的反馈报文,选择“最佳”速率作为下一步发送速率。响应函数中 RTT 与 p 的测量精确性决定着速率调节的准确度。文[19]给出了一种测量 RTT 的有效方法,而文[18]则给出了一种精确表达报文丢失率的方法。

D. Sisalem 等人采用了另外一种测量瓶颈链路带宽的方法^[20],如(3)式所示,发送方定时向接收者发送两个连续的探测报文,接收者根据两个探测报文到达的时间差计算瓶颈链路的带宽。发送方根据“每个”接收者反馈报文中的瓶颈带宽确定下一步的发送速率。

$$R = \frac{\text{探测报文大小}}{\text{两个探测报文间的时间差}} \quad (3)$$

S. J. Golestani 给出了一种通用的计算接收者接收速率的公式,如(4)式所示。

$$r_j = \begin{cases} r_j + a_j(r_j, \tau_j) & \text{successful packet delivery to } j \\ r_j - b_j(r_j, \tau_j) & \text{packet loss observed by } j \end{cases} \quad (4)$$

式中 r_j 为第 j 个接收者接收速率, τ_j 为计算速率的时间间隔, $a_j(r_j, \tau_j), b_j(r_j, \tau_j)$ 分别为速率增加函数与减少函数, 并且一般必须满足条件: $a_j(r_j, \tau_j) \geq 0, b_j(r_j, \tau_j) \geq 0, b_j(r_j, \tau_j)/a_j(r_j, \tau_j)$ 为 r_j 的递减函数。

基于速率的调节算法由于不需要维护太多的状态参数, 并且算法也比较简单, 因此, 一般被广泛使用, 但该方法三个缺陷: a) 每个接收者需精确测算 RTT 与报文丢失率 p , 目前对这两个参数的测量尚无特别有效的算法; b) 由于大多数情况下丢失率 p 的测量在几个 RTT 内完成, 因此算法对拥塞响应速度较慢; c) (1)、(2) 式表示的响应函数的正确性及有效性尚没有得到广泛的验证。

基于窗口的调节^[8,20,15] 该方法采用了类 TCP(TCP-like) 速率调节算法, 即在发送方保证未被所有接收者确认的报文个数小于窗口大小。基于窗口的速率调节主要分为两类:

(1) 单窗口、单速率调节(Singlerate)^[15]。在发送方维护一个发送窗口, 然后由具有最长往返时延的接收者决定发送速率: $r = w/t$ 。如文[15], 每个接收者和中间节点分别维护两个窗口(最大接收窗口、未确认窗口), 由中间节点对下游子节点(或接收者)的窗口进行聚集, 再发送到上一级中间节点或发送方。

(2) 多窗口、多速率调节(Multirate)^[8]。发送方为每个接收者维护一个窗口, 即分别以不同的速率发送数据。

上述方法(1)虽然可扩展性好, 但由于以最低速率发送数据, 因此对具有较高带宽的接收者是很不公平的。相反, 方法(2)虽然能有效地利用网络资源, 但由于发送方需维护大量的窗口及不同的缓存数据, 因此扩展性较差。另外所有基于窗口的速率调节算法都将拥塞控制与数据恢复耦合在一起, 这大大地增加了算法设计的复杂度, 可配置性与可扩展性均不如基于速率的调节算法。

S. J. Golestani 给出了一种通用的计算接收者窗口的公式, 如(4)式所示。

$$w_j = \begin{cases} w_j + A_j(w_j, \tau_j) & \text{successful packet delivery to } j \\ w_j - B_j(w_j, \tau_j) & \text{packet loss observed by } j \end{cases} \quad (4)$$

式中 w_j 为第 j 个接收者的窗口, τ_j 为计算速率的时间间隔, $A_j(w_j, \tau_j), B_j(w_j, \tau_j)$ 分别为速率增加函数与减少函数, 并且一般必须满足条件: $A_j(w_j, \tau_j) \geq 0, B_j(w_j, \tau_j) \geq 0, B_j(w_j, \tau_j)/A_j(w_j, \tau_j)$ 为 w_j 的递减函数。

除上述两种调节算法外, 在基于层次的可靠组播通信中, 发送方根据接收者的反馈信息调节数据的层数与层次的速率, 并向每个接收者广播层次信息。设共有 l 层, 每层数据分别为 $L_i, i=1, 2, \dots, l$, 并且 $L_i > L_j, i < j$, 每一层分别使用不同的组播地址。接收者测量其瓶颈链路带宽(设为 r), 根据(5)式决定加入或离开最高层次的组播(这个过程可能是递归的)。

$$r \leq \sum_{i=1}^l L_i \quad (5)$$

可靠组播层次传输的优点是可以适应接收者的异构性要求并具有良好的可扩展性。但其缺点也是明显的(第3节已作过介绍)。

7. 公平性

公平性是组播拥塞控制机制能否得到广泛应用的基础, 也是每个可靠组播拥塞控制机制所追求的设计目标之一。可

靠组播拥塞控制公平性包含两个方面的内容: (1) 协议间公平, 主要指 TCP 友好公平; (2) 协议内公平, 指接收者之间的公平。关于可靠组播拥塞控制的公平性总结, 翟明玉等人已作过较为详细的描述^[2]。

为达到 TCP 友好公平, 大多数可靠组播拥塞控制算法都采用了类 TCP(TCP-like) 的拥塞控制方法, 即第6节描述的前两种方法之一: 符合公式(1)或(2)的基于速率的调节与基于窗口的调节。但大多数算法对 TCP 友好公平性只作了定性分析(如仿真试验), 而没有给出具体的公平性定义及定量描述。I. Rhee 等人^[15]给出了一种 TCP 友好公平的度量方法: 公平性指数 FI :

$$FI = [\sum_{j=0}^{N-1} T_j]^2 / N \sum_{j=0}^{N-1} T_j^2 \quad (6)$$

其中 T_j 表示第 j 个协议实例在瓶颈链路的平均吞吐量, N 为协议实例个数。公平指数 FI 的值在 $[1/N, 1]$ 之间, FI 越接近 1 表示协议越 TCP 公平。

D. Rubenstein 等人^[5]给出了多会话(Multi-Session)、多速率(Multi-Rate)下的接收者之间的最大最小公平(Max-Min Fair)定义:

定义1 当一种接收者速率分配方案满足下列条件时, 该速率分配为最大最小公平(max-min fair): (1) 接收者 $r_{i,k}$ 的速率为 $a_{i,k}$, 是可行的; (2) 对任何另外一种速率分配方案($r_{i,k}$ 的速率为 $\bar{a}_{i,k}$), 如果 $\bar{a}_{i,k} > a_{i,k}$, 则一定存在另外一个接收者 $r_{i',k}$ ($r_{i',k} \neq r_{i,k}$), 其速率分配有下式成立 $a_{i,k} \geq a_{i',k} > \bar{a}_{i',k}$ 。其中 $r_{i,k}$ 表示会话 k 中的第 i 个接收者, $a_{i,k}$ 为接收者 $r_{i,k}$ 的分配速率。

结论与展望 可靠组播已被许多应用证明是一种非常有效的数据分发机制。在实际推广应用时, 可靠组播遇到的两个最大的难题是拥塞控制与数据恢复, 本文主要针对拥塞控制。尽管近几年已出现了大量关于组播拥塞控制的方案, 但这些方案或人为设定一些条件、或牺牲一方面的性能要求而获取另一方面的性能要求, 并且大多数拥塞控制算法仅仅在仿真平台上得到验证, 与真正实现并在 Internet 上运行还有很长一段距离。

本文首先指出并分析了可靠组播拥塞控制中存在的难点及基本问题, 然后详细论述目前几种较为典型的可靠组播拥塞控制机制的出发点、优势及缺陷。与此同时, 我们认为组播拥塞控制尚有以下几个方面亟待解决:

(1) 公平性问题, 其中包括 TCP 友好公平的定量研究与多协议、多速率接收者间的公平, 这些是影响 Internet 稳定性与鲁棒性的重要因素。

(2) 基于层次的媒体数据传输是近年来的研究热点, 但如何根据网络状态与接收者的能力与数目动态调节传输数据的层次与速率是研究的重点与难点。

(3) 由于可靠组播拥塞控制对拥塞的响应度是保证其公平性与网络稳定性的重要因素, 因此如何提高可靠组播对拥塞的响应速度也应是未来研究的一个重点^[21]。

参考文献

- 1 Obraczka K. Multicast Transport Protocols: A Survey and Taxonomy. IEEE Communication Magazine, 1998(Jan.): 94~102
- 2 翟明玉, 吴国新, 顾冠群. Internet 可靠多点投递拥塞控制研究进展. 计算机研究与发展, 2000, 27(1): 9~16
- 3 吴文峻, 袁树宣. 可靠组播研究综述. 计算机科学, 2001, 28(2): 58~62

(下转第47页)

```

.....
      (oil:hasOperand rdf:resource="#Qn"/>)
    (/oil:OR)
  (/oile:Conclusion)
(/oile:INF)

```

图8 基于 OIL 及其扩展的推理公式表示示例

结论 XML 的出现推动 Web-NG 的发展,语义化 Web 可以建立在 XML/XMLS 上,也可以建立在 RDF/RDFS 上,我们认为基于 RDF/RDFS 实现语义的互操作比基于 XML/XMLS 更具有优越性。关于 Web 资源的元数据处理除了需要元数据的语义表示模型和原语外,还需要先进的方法和技术,KE 在领域模型及其它方面的研究成果可以为语义化 Web 带来新的动力。

本文介绍了语义化 Web 的关于语义建模的一种主要方法和建模原语规范——RDF(S)、语义化 Web 的分层结构,并且介绍了 KE 研究中的热点技术领域模型和相应的建模方法。在此基础上介绍了 Web 技术和领域模型建模方法结合的一种产物——OIL,分析了 OIL 对 RDFS 建模原语的扩展。基于 OIL/RDFS 定义,我们面向命题逻辑公式和推理规则的表达作了进一步的扩展,通过示例表明基于这种扩展可以表示命题公式和推理规则,进而面向应用可以构成规则库。

在本文的基础上,我们将进行以下两个方面的研究:在理论上,我们可以把公理视为对象,基于 OIL/RDF(S)进行命题逻辑和谓词逻辑的公理化研究,同时表示 Horn 子句和 Datalog 等定义规则库模式;在具体应用方面,我们可以借用领域模型的方法建立基于 Web-NG 的网络管理、目录服务、智能信息集成、数据挖掘与知识发现等应用基础。

参 考 文 献

1 W3C. PICS Label Distribution Label Syntax and Communication Protocols. Version 1. 1. W3C Recommendation, Oct. 1996. <http://www.w3.org/TR/REC-PICS-labels>

(上接第54页)

4 Shi S, Waldvogel M. A Rate-Based End-to-End Multicast Congestion Control Protocol. 2000

5 Rubenstein D, Kurose J, Towsley D. The Impact of Multicast on Network Fairness. In: ACM SIGCOMM'99 8/99, MA USA

6 Delucia D, Obraczka K. A Multicast Congestion Control Mechanism for Reliable Multicast. In: Proc. IEEE INFOCOM, 1997

7 Mathis M, et al. SBMCC: A Generic Source-based Congestion control Algorithm for Reliable Multicast. Feb. 2000. Available at: <http://www.ecse.rpi.edu/Homepages/shivkuma/research/papers/sbmcc2.ps>

8 Golestani S J. Fundamental Observations on Multicast Congestion Control in the Internet. In: Proc. IEEE INFOCOM, 1999

9 Roca V. On the Use of On-Demand layer Addition (ODL) with Multi-Layer Multicast Transmission Techniques. WORK IN PROGRESS, June 2000

10 Vicisano L, Rizzo L, Crowcroft J. TCP-like Congestion Control for Layered Multicast Data Transfer. In: IEEE INFOCOM'98, Feb. 1998

11 Montgomery T A. Loss Tolerant Rate Controller for Reliable Multicast. [Tech Rep: NASA-IVV-97-011]. West Virginia Univ.

12 Sisalem D, Wolisz A. MLDA: A TCP-Friendly Congestion Control Framework for Heterogeneous Multicast Environments. In: Eighth Intl. Workshop on Quality of Service (IWQoS 2000), Pittsburgh, PA, June 2000

2 Miller E. An Introduction to the Resource Description Framework. D-Lib Magazine, May. 1998

3 Horrocks I, et al. The Ontology Inference Layer OIL. <http://www.ontoknowledge.org/oil>.

4 Bray T, et al. Extensible Markup Language (XML) 1. 0. W3C Recommendation, World Wide Web Consortium, 1998. <http://www.w3c.org/TR/REC-xml>

5 (美)Didier Martin 等. XML 高级编程. 机械工业出版社, 2001

6 Biron P V, et al. XML-Schema. W3C Recommendation, 2001. <http://TR/REC-xml-schema>

7 Lassila, Swick R R. Resource Description Framework (RDF) Model and Syntax. W3C Recommendation, World Wide Web Consortium, Feb. 1999. <http://www.w3c.org/TR/REC-rdf-syntax>

8 Brickley D, Guha R V. Resource Description Framework (RDF) Schema Specification. W3C Recommendation, World Wide Web Consortium, Mar. 2000. <http://www.w3c.org/TR/REC-rdf-schema>

9 姚绍文, 刘爱莲, 等. 基于 XML 的 Web 自动化技术. 中国 XML 技术 2000 研讨会, 北京, 2000

10 姚绍文, 周明天, 等. 下一代的 Web 技术. 计算机科学, 2001, 28 (1): 78~84

11 Berners-Lee T. Semantic web road map. World Wide Web Consortium, Sept. 1998. <http://www.w3c.org/DesignIssues/Semantic.html>

12 Studer R, et al. Situation and Perspective of Knowledge Engineering. In: Knowledge Engineering and Agent Technologies. IOS Press, 2000

13 Swartout W. Ontologies. IEEE Intelligent Systems, 1999 (Jan.): 18~19

14 Chandrasekaran B, et al. What are ontologies, and why do we need them. IEEE Intelligent System, 1999 (Jan.): 20~26

15 Lassila O. Web Metadata: A Matter of Semantics. IEEE Internet Computing, 1998 (Jul.-Aug.): 30~37

13 Lin J C. RMTP: A Reliable Multicast Transport Protocol. In: Proc. of IEEE Computer Society Press, 1996. 1414~1424

14 Whetten B, Montgomery T, Kaplan S. A High Performance Totally Ordered Multicast Protocol. In Theory and Practice in Distributed Systems. LNCS 938, Springer Verlag

15 Rhee I, Balaguru N, Rouskas G N. MTCP: Scalable TCP-like Congestion Control for Reliable Multicast. In: Proc. of the INFOCOM, March 1999

16 Donahoo M J, Ainapure S R. Scalable Multicast Representative Member Selection. In: IEEE INFOCOM 2001

17 Lacher M S, Nonnenmacher J, Biersack E W. Performance Comparison of Centralized Versus Distributed Error Recovery for Reliable Multicast. IEEE/ACM Transactions on Networking, 2000, 8 (2): 224~238

18 Thapliyal P, et al. LE-SBCC: Loss-Event Oriented Source-Based Multicast Congestion Control. submitted to NOSSDAV 2001

19 Basu A, Golestani S J. Estimation of Receiver Round Trip Times in Multicast Communications. RMRG Meeting, Dec. 1998

20 Yano K, McCanne S. A Window-Based Congestion Control for Reliable Multicast Based on TCP Dynamics. In: ACM SIGMM. Electronic Proceedings (Available from: <http://woodworm.cs.uml.edu/~rprice/ep/yano/>)

21 Legout A, Biersack E W. Fast Convergence for Cumulative Layered Multicast Transmission Schemes. In: Proc. of SIGMETR-ICS'2000, Santa Clara, CA, July 2000