第 35 卷 第 2 期 计 算 机 工 程 2009 年 1 月
Vol.35 No.2 Computer Engineering January 2009
·网络与通信· 文章编号：1000—3428(2009)02—0086—03 文献标识码：A 中图分类号：TP301.6

基于可用带宽测量的应用层组播算法

杨 珊，黄东军，周 伟
(中南大学信息科学与工程学院，长沙 410083)

摘 要：针对组播分发树建立过程的特性和需求，提出一种基于可用带宽测量的应用层组播算法。该算法以组播数据作为测试源，建立输入数据率和单向时延的关系模型，融合可用带宽测量与组播分发树的建立，以降低测量开销和对网络的影响，仿真实验表明，生成的组播树具有高吞吐量和低链路压力的特点。

关键词：应用层组播；带宽测量；算法融合；组播树构建

Application Level Multicast Algorithm Based on Available Bandwidth Measurement

YANG Shan, HUANG Dong-jun, ZHOU Wei
(School of Computer Science and Engineering, Central South University, Changsha 410083)

【Abstract】 According to the characteristics and special requirements of constructing Application Level Multicast(ALM) tree, this paper proposes a new ALM algorithm based on available bandwidth measurement. Depending on the relation between the input data rate and the one-way delay, it uses multicast data as the probing source, and calculates the available bandwidth. The method helps the application reduce probing cost and influence on the network. Simulation demonstrates the effectiveness of the algorithm in terms of throughput and average link stress.

【Key words】 Application Level Multicast(ALM); Bandwidth measurement; algorithm integration; multicast tree construction

1 概述

与 IP 组播相比，应用层组播(Application Level Multicast, ALM)具有更强的灵活性，便于针对特定应用进行优化。然而，由于应用层组播以端系统之间的连接作为组播树的分支，而不管它们是否共享物理链路或路由器，所构建的组播树常常带有较大的折返，因此大大降低了组播性能。为了显式地避开折返，优化组播树的结构，ALM 算法需要网络的拓扑结构、QoS 状态等信息。但是网络是分层的，应用系统通常不能直接获得网络状态信息。所以，把网络看成“黑盒”，通过测量技术感知底层状态是一种切实可行的技术路线。

传统测量技术采用独立的数据进行网络探测[1-5]，不仅会增加网络负担，也没有考虑应用数据本身的潜力，如果希望在组播中利用组播数据完成探测，就需要考虑测量与组播的结合。因此，本文探讨了一种新的基于可用带宽测量技术的ALM算法，由组播的上游节点承担测量任务以适应组播传输的实际需要，同时基于探测速率模型，利用组播源数据本身(设为MPEG视频流)的变化特性，建立输入速率和单向时延的关系模型，在组播算法中实现测量，从而更加有效地支持节点连接到组播树上。

2 概念和相关工作

定义 1 带宽容量：在没有背景流量的条件下，一条端到端路径能够给数据流提供的最大 IP 层吞吐量。

定义 2 可用带宽：在不影响现有背景流量数据传输速率的前提下，当前路径上所有链路可用带宽的最小值。

目前，用测量技术感知IP网的可用带宽仍在研究中。以往的研究大致分为 2 类：(1)基于探测间隔模型的算法[1-2]，即首先估计出网络路径中瓶颈链路的容量，该瓶颈容量的估计是否准确直接影响到可用带宽测量的准确性。(2)基于自导拥塞概念发展而来的探测速率模型。该模型基于以下简单的启发式原理：如果发送探测包流的速率低于被测路径上的可用带宽，则在接收端探测包的到达速率将与发送端的相匹配；反之，会在网络中的瓶颈链路上出现排队现象，探测包流被延迟，导致探测包在接收端的速率低于发送端的速率。典型的算法有TOPP[3]和SloPS[4]。TOPP以递增的速率向目的主机发送包对组成的包列，根据不同包对输入输出速率之间的关系来判断可用带宽。SloPS算法思想与TOPP非常接近，只不过它用单向时延的变化趋势判定下一次的发送速率。由于用了二叉查找的办法，因此提高了算法收敛速度。pathload[5]是基于SloPS思想实现的可用带宽测量工具。

从 ALM 算法的角度来看，还存在可用带宽测量技术如何与 ALM 算法融合的问题。最早注意到在 ALM 算法中应用网络状态信息的是 Nadara 组播协议，但该协议没有给出获得网络信息的具体方法。稍后的 NICE 协议提出由加入节点探测到源端的可用带宽，同时探测到了潜在父节点(除源端以外的其他在树节点)的可用带宽，从中选择较为有利的路径连接到组播树上。显然，NICE 的设计只适用于无向网络，实际网络环境多是有向的，从加入节点到源端的可用带宽不等于反向路径的可用带宽，而组播关心的是从源端到接收端的可用带宽，这样势必会降低组播算法对网络变化的反应性。而独立的探测数据又会占用额外的网络资源，加重数据的传输负担。

作者简介：杨 珊(1983－)，女，硕士研究生，主研方向：网络与多媒体技术；黄东军，教授；周 伟，硕士研究生

收稿日期：2008-07-30 **E-mail：**pure_2001@126.com

3 嵌入可用带宽测量的组播算法

3.1 测量原理

在网络计算中，一个连接的可用带宽是特定应用程序提供必要服务质量的基础。一个连接的延时由 4 个部分构成：传播延时，传输延时(由可用带宽和需要传输的数据量决定)，排队延时和处理延时。其中，传播延时对于一条路径来说是常量；处理延时在当前高速路由、交换设备的条件下可以忽略不计；传输和排队延时是变量。设 T_q 为排队延时，T_r 为传播延时，T为测量到的总延时，A为一个链接的可用带宽。本文借助MPEG视频流数据随时间做周期性变化这一特性来实现对可用带宽A的测量。在MPEG数据流中，I帧被压缩为单个图像，P帧和B帧的压缩仅使用图像的差值。这样，在传输具有依次递减数据率特点的I, P, B帧时，测量到的传输延时会依次递增，即得到不同的传输时间T_1, T_2, T_3：

$$T_1=\frac{I}{A}+T_q+T_r,\ T_2=\frac{P}{A}+T_q+T_r,\ T_3=\frac{B}{A}+T_q+T_r$$

但上式并不能直接用来计算 A，尽管 T_r 对于确定路径是常数，但因为 T_q 仍然是变量，所以无法从 T 中分离出 T_q 和 T_r。

由 PRM 模型分析可知，当传输的数据小于 A 时，T 与传输数据的增加呈线性关系；而当传输数据大于 A 时，T 与传输数据的增加呈非线性关系，因此，可以通过计算 T 的增长趋势来逼近 A。为此，需要使测量数据从小于 A 增加到大于 A，这可以通过在探测数据中增加一个额外增量 Δ 来实现。虽然用到独立探测数据，但 Δ 只占总探测数据 $(I,P,B)+\Delta$ 的一部分，达到了降低额外探测数据的目的。A 的测量问题由此转化为传输时间变化趋势的测量问题，当逐步增加 $(I,P,B)+\Delta$ 直到超过 A 时，传输时间的增量变化就会由线性增长变为非线性增长，一旦发现这个转变，即认为 $(I,P,B)+\Delta$ 等于 A。由于带宽实际上是一个速率值，因此 $(I,P,B)+\Delta$ 应准确定义为单位时间内(1 s)传送的数据量。

3.2 测量算法的实现

在实际的测量过程中，由于一开始并不知道初始输入速率和可用带宽的大小关系，因此有一个传输时间变化的观察过程，用以正确调整发送速率逼近可用带宽大小，基本过程如下：

```
Begin
    Ri sends stream(T1) to Rj in ascending order;   //Ri为探测发送
//端，Rj为接收端
    If (Ri检测到Tn呈线性增长趋势) Do
      stream(Tn)=stream(Tn-1)+; // Tn表示发送第n个变比特码流//
时的时延，stream(Tn)为发送第n次探测流的大小
      Ri send stream(Tn) to Rj;
      Rj compute Tn;
      While (Tn is non-linearly increased)
    Endif
    A= stream(Tn);
    If (Rj检测到T呈非线性增长趋势)   Do
     stream(Tn)=stream(Tn-1) - ;
      Ri send stream(Tn) to Rj;
      Rj compute Tn;
      While(Tn is linearly increased);
   Endif
    A= stream(Tn);
    Ri forward A to Rj;
End
```

3.3 判断转折点

算法的另一个难点是如何尽快搜索出转换带宽。首先将传输时间增量变化的转变点定义为可用带宽捕获点。一般的探测算法都是按照发送包速率增大(减小)的方向寻找可用带宽——正(反)向点的。一旦检测到时延增长趋势有明显变化，即获得可用带宽，程序就可以停止了。然而，这一转折点与 Δ 有关，Δ 越大，则该正(反)向点能越早地找到，但是测量值越偏离实际的可用带宽值；Δ 越小，意味着接收端需要缩短反馈的周期，报文开销越大。为此，本文探测算法采用一种回归搜索策略，以获得比较准确的测量结果。具体做法是，设初始输入速率小于可用带宽，之后程序增大通信量，当程序搜索到正向点时，继续增大一定数量的包，然后减小探测包的速率，使队列逐渐脱离拥塞状态，探测包的延时又随之减小，当再次探测到时延趋势明显变化时，即可找到该反向点。由于这种方法是在找到正向转折点的基础上找到的，因此根据误差出现的互补性，测量结果会更准确。

3.4 组播算法描述

基于上文给出的原理，组播算法需要集成探测进程和传输时间增量变化捕获进程。

(1)由接收节点发起加入组播的进程。一个新节点通过目录机制获得当前在树节点信息(地址)，通过向在树节点发送请求，激活上游节点的探测进程。

如图 2 所示，节点 P 为要加入组播树的新主机。节点 P 首先联系目录服务器 DS，获取当前系统的主机信息。这些主机信息由目录服务器随机产生，目的是使节点加入的通信开销对所有主机来说是分布的。图 2 中 DS 返回组播树的拓扑结构信息以及随机产生的在树节点 B, D, H, I。然后节点 P 依次向以上节点发送 Join_Probe 报文，收到报文后，节点 B, D, H, I 发送 $(I,P,B)+\Delta$，开始探测过程。

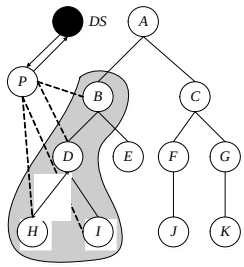

图 1　新节点激活在树节点发起探测进程示意图

(2)视频码流测量法要求发送端与接收端协同工作。发送端负责探测报文的传送，计算可用带宽。接收端在探测报文到达后将到达的时间戳回送给发送端。在不考虑探测报文丢失的前提下，发送端能够根据报文的发送及接收时间，计算出从发送端到接收端的单向延时。

(3)发送节点在传送 $(I,P,B)+\Delta$ 过程中通过观察传输时间增量的变化捕获进程，得到当前可用带宽 A。

(4)接收节点在收到多个上游节点告知的可用带宽 A_i 后，选择其中最大的可用带宽路径连接到组播树，完成加入过程。

4 仿真实验

4.1 异构负载下的测量结果

首先应用 NS2 验证本文端到端可用带宽测量算法的准确性，具体配置见图 2。默认测试路径各链路的带宽为 P={100, 10, 4, 5, 100} Mb/s，因此，路径容量为 4 Mb/s。本文选取同样基于 PRM 模型的 pathload 作为参考对象，仿真设置与文

献[5]完全相同：测试路径为 5 跳，瓶颈链路(同时也是可用带宽最少的链路)出现在第 3 跳，其利用率为 u。每一跳的背景流由 10 个 Pareto 源产生，α =1.5。

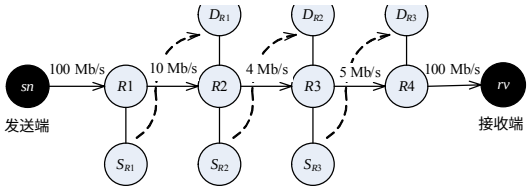

图 2　实验拓扑图

链路利用率 u 分别取值 20%, 40%, 75%。对于测量工具的每个链路负载，各测试 40 组实验数据。需要注意的是，pathload 的测量仅返回一个区间，因此，图 3 给出的是 40 次测量区间的平均值(即由 40 次测量区间上限的平均值和 40 次测量区间下限的平均值所构成)。本文测量方法每次测量的返回结果是一个值，即转折点的数值，因此，图 3 给出的是 40 次测量值的分布区间。

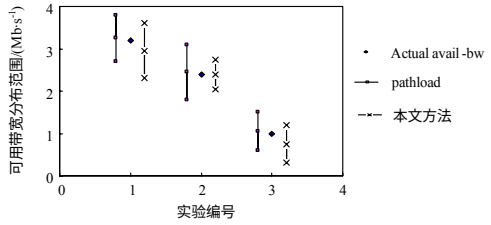

图 3　基于视频码流的可用带宽测量结果

可以看出，本方法在测试准确度上与 pathload 基本相同。但是，平均独立探测量(迭代增量 Δ)仅为 36.2 KB，而 pathload 中每个流的默认大小为 80 KB，pathload 的每次测量通常需要 12 个流。本方法的测量消耗较低，因为承担主要测量任务的是视频码流即组播数据本身，所以节省了探测资源，减小了注入网络的干扰流量。这对音视频传输系统缓解拥塞、维护媒体传输质量有着积极的意义。

4.2　组播树性能分析

在探测算法得到有效验证后，采用 NS2 模拟仿真器将其嵌入到组播树算法的构建中。

仿真实验主要从吞吐量和链路压力比较了 Narada 算法和按照视频码流测量算法建立的生成树性能。其中，吞吐量反映了各个节点获得数据的能力。在分别用 2 种不同的组播算法构建组播节点数为 40 的组播树后，从根节点发送 10 Mb 数据，记录下每个节点的吞吐量情况，如图 4 所示。

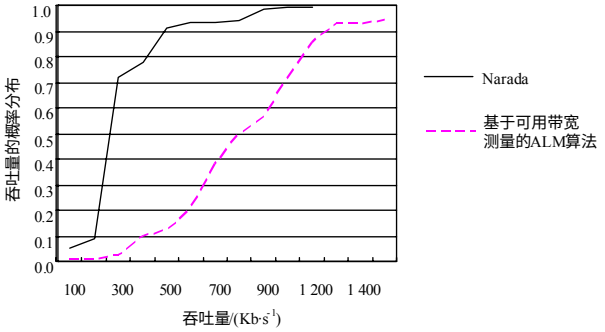

图 4　吞吐量分布

结果显示，本文算法不仅有最大的平均吞吐量，而且在回避窄带宽链路方面明显优于 Narada。这是因为基于可用带宽测量的组播算法以节点可接收数据能力为优化目标，而 Narada 以覆盖拓扑或 mesh 拓扑作为其组播树的计算基础，无法保证组播树的物理连接质量。

链路压力是指组播算法在物理拓扑的每条链路上发送同一数据包的数量，其值越小越好。图 5 中的平均链路压力定义为

$$ALS = \frac{\sum_{l \in L} S_l}{|L|}$$

其中，S_l为某条链路压力；$|L|$是物理拓扑中的链路数目。由于探测了时延信息和可用带宽，因此在建树过程中有效避开了紧链路。Narada算法的逻辑拓扑结构和底层物理网络之间存在较大差异，应用层覆盖网中的通信量给底层物理网络带来较大的压力。实验数据表明，在不同节点数的环境下，本文算法的链路压力相比Narada平均降低了 18%。

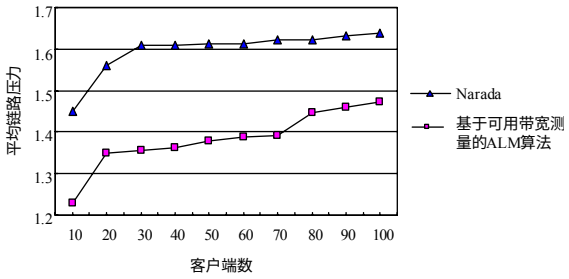

图 5　平均链路压力

5　结束语

本文利用组播数据流完成测量工作，并根据时延增长特点提出了基于视频码流的可用带宽测试方法。在建立组播分发树时，提出“边测量边组播”的思想，将带宽测量融合到组播算法中。实验结果证明，根据源数据流特点进行探测是可行的，与同样基于探测速率模型的方法相比有较大的优势，在主观上削减了主动探测对媒体传输质量的影响，降低了网络消耗和周期性探测量，建立的组播树有较高的吞吐量，能够有效避免路径折返，具有良好的应用前景。

参考文献

[1] Stauss J. A Measurement Study of Available Bandwidth Estimation Tools[C]//Proceedings of ACM SIGCOMM Conference on Internet Measurement. Miami, USA: ACM Press, 2003.

[2] Hu Ningning, Steenkiste P. Evaluation and Characterization of Available Bandwidth Probing Techniques[J]. IEEE Journal on Selected Areas in Communications, 2003, 21(6): 879-894.

[3] Melander B, Bjorkman M, Gunningberg P. A New End-to-end Probing and Analysis Method for Estimating Bandwidth Bottlenecks[C]//Proc. of the Global Internet Symp.. San Francisco, USA: IEEE Press, 2000.

[4] Jain M, Dovrolis C. End-to-end Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput[J]. IEEE/ACM Trans. on Networking, 2003, 11(4): 537-549.

[5] Jain M, Dovrolis C. Pathload: A Measurement Tool for End-to-end Available Bandwidth[C]//Proceedings of Passive and Active Measurements Workshop. Fort Collins, USA: [s. n.], 2002.