第34卷 第22期 计 算 机 工 程 2008年11月
Vol.34 No.22 Computer Engineering November 2008
·网络与通信· 文章编号：1000—3428(2008)22—0103—03 文献标识码：A 中图分类号：TP393.02

面向实时交互的 P2P Streaming 模型

许 成，蔡鸿明，姜丽红
(上海交通大学计算机科学与工程系，上海 200240)

摘 要：实施大规模分布式可视化操纵系统的一个重要瓶颈在于节点间海量数据的传输。P2P Streaming 技术逐渐成为解决该瓶颈的可行方案之一。主流 P2P Streaming 模型由于实时性上的缺陷并不适于此类系统。该文通过引入服务质量的概念并采用启发式的算法，设计一种改良的 P2P Streaming 模型。仿真实验结果表明，新模型在平均延迟、首包到达时间等指标上优于现有模型。

关键词：P2P Streaming 技术；实时交互；服务质量

P2P Streaming Model Oriented to Real-time Interactivity

XU Cheng, CAI Hong-ming, JIANG Li-hong
(Department of Computer Science and Technology, Shanghai Jiaotong University, Shanghai 200240)

【Abstract】Massive data exchange among nodes is a bottleneck of deploying distributed visual steering systems. P2P Streaming becomes one of the solutions. Current P2P Streaming models are not suitable for such applications because they are short on real-time interactivity. Typical models and design improved model are analyzed by applying QoS concept and heuristic algorithm for P2P streaming. It is indicated by the result of simulations that new model is better than typical ones in metrics such as average delay and time to first package.

【Key words】 P2P streaming; real-time interactivity; QoS

1 概述

随着 P2P 技术的发展，人们开始在应用层上通过 P2P 网络组播这些流数据。由于 P2P 网络本身的可扩展性，很好地解决了传统客户/服务器模式下海量数据传输的带宽不足问题。此类技术被称作 P2P Streaming。将 P2P Streaming 技术应用到分布式的可视化操纵系统中，如何维持系统的实时性和交互性都是值得考虑的。目前 P2P Streaming 的研究主要针对媒体流组播的应用，这些模型往往着眼于如何平衡网络负载，维持数据流的连续传递，在优化传输延迟及其传输性能的稳定性上存在着欠缺。总之，现有模型并不太适合应用于对实时交互性要求较高的分布式系统。

2 现有 P2P Streaming 模型的分析

2.1 P2P Steaming

在P2P Streaming系统中，众多的对等节点充当了数据流转发的角色。这种模式充分利用了P2P系统共享带宽的优势，从而缓解客户/服务器模式下数据源端的带宽不足。当有节点加入或者旧节点离开时，系统都需要重构数据流传播的逻辑路由。因此，针对如何组织和维持P2P方式的流数据传递就有了各种研究及模型。这些模型有着各自不同的重叠网络的拓扑结构。将其分为 2 大类[1]：源驱动模型和数据驱动模型。

源驱动模型往往由数据源端来控制节点动态出入的控制策略，拓扑结构为树型结构，源节点就是组播树的根节点。典型的例子有PeerCast[2], Narada, Zigzag等。

2.2 PeerCast

以经典的 PeerCast 为例，其数据流由数据通道和控制通道组成。前者用于传输流数据，后者用于控制节点的传输策略。数据流在以数据源为根节点的生成树上传递，每个节点向子节点传输数据和控制信息，并负责动态地维护子节点所组成的组播树。

树型的数据传播和控制管理结构简洁、易于实现，但是会对树的上层节点造成很大的负担。一旦节点离开，对后继节点的影响很大。此外，集中式的控制管理模式使得它响应网络变化(节点的动态加入、离开及失效)的速度较慢。一些改良的 PeerCast 模型采用网孔(Mesh)的模式进行资源发现。虽然组播树的高度、逻辑路由的跳数得到了平衡和优化，但是数据组播的延迟依然很大，对于要求延迟为毫秒级的实时交互应用，此类模型仍有需要改进的地方。

2.3 DONet

典型的数据驱动模型DONet[3]采用泛洪算法的思想，节点将新信息发给一组随机选择的节点，这些节点会再把信息发送给别的随机选择的节点，直到网络中所有节点都收到信息。此类算法的随机选择性加强了系统应对网络变化的弹性，具有良好的容错性。但由于大量的冗余数据占用带宽，因此这类非结构化的P2P系统的使用有很大的局限性。尽管DONet设计了一套启发式的调度算法降低了冗余的数据量，但是当系统规模扩大时，节点间交换的生存信息量仍会呈几何级的增长。此外，由于DONet组播的随机性，组播的性能指标并不稳定，因此对于实时交互而言也是相当糟糕的。

综上，针对现有模型的这些问题，有必要设计一套改良的、面向实时交互的 P2P Streaming 模型及算法，以适合大规模分布式的可视化操纵应用。

3 新模型的设计

3.1 问题描述

将P2P系统的物理网络层记作$G=(V, L)$。其中，集合V代

基金项目：国家自然科学基金资助项目(60603080, 70471024)
作者简介：许 成(1983－)，男，硕士，主研方向：智能信息处理，分布式系统；蔡鸿明，讲师、博士；姜丽红，副教授、博士
收稿日期：2008-03-27 **E-mail：**koalaxu@sjtu.edu.cn

表了网络中所有节点；$\mathcal{L}$代表了所有物理链路。在覆盖网络层，将系统所需组播的流数据集合记作M。对于每个传输会话$m\in M$，数据源节点记作s_m，所有参与传输会话的节点记为集合R_m。则该会话的覆盖网络图可以记作$G_m=(V_m,\varepsilon_m)$，其中，$V_m=\{s_m\}\cup R_m$；ε_m代表了覆盖网络层节点间的有向连接。例如，$(v_s, v_t)\in\varepsilon_m$表示数据流由节点$v_s$传向节点$v_t$，而这一连接可能经过了若干物理的路由，$(v_s, v_t)=L(s, t)$。

根据上述定义，获得最小平均延迟的问题就可以描述为寻找一个使得平均延迟函数

$$average\_delay(G_m)=\frac{1}{|R_m|}D$$

$$D=\sum_{v_i\in R_m}\ \sum_{(v_s,v_i)\in\varepsilon_m(s_m,v_i)}\ \sum_{l_k\in L(s,t)} delay(l_k)$$

取得最小值的$G_m^{optimal}$。由此可见，此问题的求解是非常复杂的。更重要的是，对于建立在应用层的P2P系统而言，网络层的拓扑结构及其连接线路的实际性能往往是不可知的或部分可知的。因此，设计一套启发式的算法来优化这种P2P Streaming网络的构建是合理的选择。

3.2 设计思想

鉴于P2P系统对于实际网络拓扑结构及其性能的不可知性，希望节点能够动态地收集和反馈数据流传输的服务质量信息，从而为系统选择逻辑链路提供依据。节点自行统计所收到数据流的数据丢失率、传输延迟等重要的QoS参数。根据实际应用的不同，系统可以设定特定的QoS函数，由这些性能参数计算出的QoS函数值作为构建组播树时节点选择的重要根据。对于节点v_i而言，QoS(v_i)反映了其能否胜任成为数据转发节点的参考标准。对应于某个具体的数据接收节点r_i，由于两者网络物理链路情况的未知，有必要发送测试数据流来测量两者之间数据传输的服务质量，再根据系统设定的QoS函数来做计算，将其记作$f(v_i, r_i)$。

笔者设计的P2P Streaming模型拟采取生成树结构。对于传输会话$m\in M$，参与其组播的每个节点$v_i\in V_m$，需要维护子节点列表Child$_m$(v_i)、候选节点列表Candidate$_m$(v_i)、备用父节点列表Backup$_m$(v_i)以及祖先节点列表Ancestor$_m$(v_i)。节点向自己的子节点转发由父节点接收来的数据流。如是，数据流由上而下经由整个生成树以完成向所有节点的组播。

4 新模型的算法实现

4.1 节点加入

当节点v_{new}请求进入传输会话m时，会先和数据源节点s_m进行连接。s_m返回一个候选父节点的列表Candidate$_m$(s_m)。v_{new}同时向每个候选节点$c_i\in$Candidate$_m$(s_m)发出请求信息尝试进行连接，测试与它们之间的传输性能。v_{new}根据反馈的传输性能信息，选出最优的节点c_{max}作为它的父节点，即$f(c_{max}, v_{new})=\max\{f(c_i, v_{new})\}$。此外，$v_{new}$还将次优的若干节点尝试加入自己的备用父节点的队列中。

4.2 节点退出

当节点v_{leave}离开传输会话m时，向所有的子节点Child$_m$(v_{leave})发出控制信息表明自己即将离开。它的每个子节点$c_i\in$Child$_m$(v_{leave})，对其备用父节点队列中的每个节点$b_j\in$Backup$_m$(c_i)发出请求信息，尝试进行连接。与节点加入的处理方式类似，c_i选取传输性能最优的备用父节点b_{max}作为新的父节点，即$f(b_{max}, c_i)=\max\{f(b_j, c_i)\}$。完成以后，$c_i$向$v_{leave}$发送一个控制信息表明已经找到新的父节点，当收到所有节点的反馈后，v_{leave}向自己的父节点发送消息，断开连接，退出系统。

4.3 节点失效

当节点v_{fail}失效时，不会向任何节点发出控制信息。因此，系统需要靠心跳机制去感知节点的失效。失效节点的父节点parent$_m$(v_{fail})发现其失效后，直接将其从自己的子节点列表中删除。失效节点的子节点Child$_m$(v_{fail})，类似于节点退出情况的处理方式，通过备用父节点机制快速重构覆盖网络。

4.4 节点候选机制

加入传输会话m的节点v_i会周期性地计算自己收到数据流的QoS函数，记作QoS$_m$(v_i)；同时还会向父节点反馈一个根据性能优劣排序的候选节点列表Candidate$_m$(v_i)，表明v_i的后继节点中胜任转发工作的节点。因此对于某个节点v_i而言，它的每个子节点$c_j\in$Child$_m$(v_i)都会反馈一个候选节点列表，它只需对这些列表Candidate$_m$(c_j)以及$\{v_i\}$一起做多路归并排序，即可得到有序的候选节点列表Candidate$_m$(v_i)。如是自底向上地归并，数据源节点可以不用探询整个网络，而找到合适的候选节点列表。

4.5 节点备用机制

加入传输会话m的节点v_i需要维护备用父节点列表Backup$_m$(v_i)，周期性地尝试与每个备用父节点进行连接，测试传输性能。如果备用父节点离开了系统或是性能恶化，v_i会向源节点s_m索取候选节点列表Candidate$_m$(s_m)，通过类似节点加入的操作，尝试选择新的备用父节点。

对尝试加入备用父节点列表的节点v_{backup}，如果parent$_m$(v_i)$\in$Ancestor$_m$(v_{backup})，则加入失败；否则加入成功。因为将自己父节点的后代节点作为自己的备用父节点没有意义。

5 实验

5.1 实验参数

实验采用美国波士顿大学的BRITE项目[4]来构建仿真的P2P环境，以评估改良后的模型及其算法的性能。笔者随机构建了一个基于Router的有1 000个节点的Waxman型虚拟网络环境。节点的带宽随机地服从10 MB/s~1 024 MB/s的均匀分布。实验中开启2个流量为2 MB/s的数据流传输服务，节点每秒都有0.2%的概率加入某传输会话，并持续200 s~1 000 s(随机地服从指数分布)后离开该会话。同时，在线节点每秒都有0.05%的概率发生失效。

在上述的网络环境下，分别对客户-服务器模型、PeerCast模型以及本文的新模型(记作Q-P2PS)进行1 000 s的实验。

5.2 实验结果及比较

首包到达时间(T2FP)反映了构建覆盖网络结构的效率。在图1中，C-S结构简单，T2FP指标低且稳定。PeerCast和T2FP指标随着时间的增长而增长。PeerCast的组播树随着节点增多而增大，寻找合适转发节点的时间也越来越长；而新模型始终仅需从有限候选节点中寻找，因此要优于PeerCast。

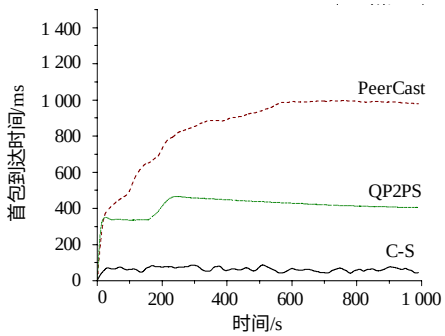

图1 T2FP的对比图

在图 2 中，C-S 模式下平均延迟非常低趋于直线；随着时间的推移，PeerCast 组播树的高度不断增加，平均延迟也越来越高；而新模型的平均延迟时间始终处于较低的水平，可见引入基于 QoS 的启发式算法很有必要。

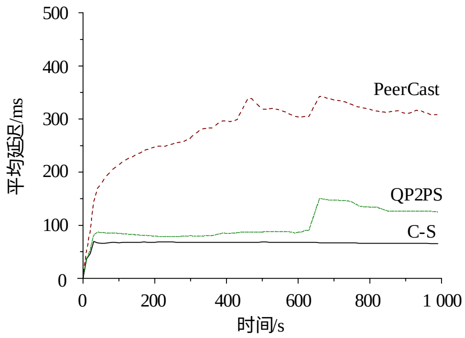

图 2　平均延迟的对比图

在图 3 中，C-S 模型在 2 min 后丢包率急剧上升。P2P Streaming 技术很好地平衡了传输负载，PeerCast 和新模型的丢包率都处于较低的水平。新模型的丢包率还要略低于 PeerCast，是因为新模型能够更快地重构组播树，从而减少节点失效导致的丢包。

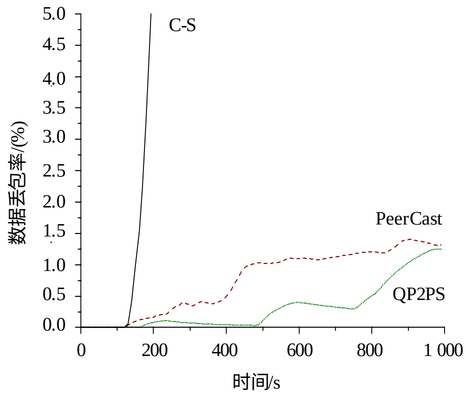

图 3　丢包率的对比图

5.3　实验结论

通过实验数据的比较，可以发现优化改良后的新模型更适合于需要实时交互的系统。新模型性能上的优化也付出了一定的代价，每个节点都必须周期性地交换一定冗余的控制信息。但笔者认为这一代价是必要的，使系统中的节点自组织地构建出更加优化的重叠网络结构，从而使 P2P Streaming 系统更加适用于不同应用的需求。

6　结束语

本文设计的 P2P Streaming 模型及其算法在一定程度上弥补了传统的 P2P Streaming 模型针对实时交互应用的不适用性，通过启发式算法把物理上接近的节点尽可能自组织地聚合在同一组播树下，使之更适用于各种需要实时传递海量数据的大规模分布式系统，提高了将 P2P Streaming 技术应用到此类系统的可行性。

当然，本文的研究工作仍较为初步，将研究成果具体应用到实时大规模分布式系统中有待进一步的研究。

参考文献

[1] Silverton T, Fourmaux O. Source vs Data-driven Approach for Live P2P Streaming[C]//Proc. of the International Conference on Mobile Communications and Learning Technologies. [S. l.]: IEEE Press, 2006: 99.

[2] Bawa M, Deshpande H, Garcia-Molina H. Transience of Peers & Streaming Media[J]. ACM SIGCOMM Computer Communication Review, 2003, 33(1): 107-112.

[3] Zhang Xinyan, Liu Jiangchuan, Li Bo, et al. Coolstreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming[C]//Proc. of the 24th Annual Joint Conference on Computer and Communications Societies. [S. l.]: IEEE Press, 2005: 2102-2111.

[4] Medina A, Lakhina A, Matta I, et al. BRITE: An Approach to Universal Topology Generation[C]//Proc. of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. [S. l.]: IEEE Press, 2001: 346-353.

(上接第 93 页)

拓扑结构进行案例的重组和修正，形成本次项目风险分析网络拓扑结构；同时为每个网络节点设定条件概率，根据软件风险评估模型的概率推理系统，计算各网络节点的发生概率，设定风险后果集，使用模糊语言对风险损失进行评估，计算风险综合影响及组合影响，根据评估结果，对风险进行解释和等级对比，定性分析项目中风险的严重程度，给出风险预防及控制措施。

实践证明，软件风险发生的概率通过贝叶斯网络拓扑结构进行推理，降低了评估的难度和主观性；使用模糊性语言评估风险后果及损失，解决了专家评估的不确定性问题；根据评估结果，提供风险预防控制措施，避免和减少了风险造成的损失；模型中具有完善的学习机制，逐步丰富软件企业的风险数据库，在不断的学习和修正过程中提高了风险的预测和应变能力，为有效地降低风险发生概率、提高软件开发成功率提供了一种新的途径，具有很好的应用价值。

参考文献

[1] 唐爱国，王如龙. 软件项目范围变更流程与过程控制研究[J]. 项目管理技术, 2006, 4(9): 71-73.

[2] Chickering D M. Learning Equivalence Classes of Bayesian Network Structures[J]. Machine Learning, 2002, 2(3): 445-498.

[3] Wang Shuangcheng, Yuan Senmiao. Research on Learning Bayesian Networks Structure with Missing Data[J]. Journal of Software, 2004, 14(7): 1042-1048.

[4] 李美华，付　宏. 软件项目风险评估模型的建立[J]. 吉林大学学报, 2005, 23(6): 696-701.