P2P 流媒体系统中基于网络坐标的拓扑优化研究

林予松[1] 崔 勇[2] 王宗敏[1]
[1](河南省信息网络重点学科开放实验室 河南 郑州 450052)
[2](郑州大学信息工程学院 河南 郑州 450052)

摘 要 近年来,基于 P2P 的大规模流媒体直播系统得到了广泛应用,但是应用层覆盖网与底层物理网络存在失配问题。针对该问题,提出了一种基于 Vivaldi 网络坐标算法的流媒体系统拓扑优化机制——NCSTO(Network Coordinate System in P2P Streaming Topology Optimization),通过采用双重采样和样本过滤器,能够有效地针对覆盖网进行拓扑优化,减少网络失配,提高系统运行效率,降低带宽浪费。

关键词 P2P 流媒体 拓扑优化 网络坐标 双重采样

ON NETWORK COORDINATE-BASED TOPOLOGY OPTIMISATION IN P2P STREAMING MEDIA SYSTEM

Lin Yusong[1] Cui Yong[2] Wang Zongmin[1]
[1](*Henan Provincial Key Lab on Information Networking, Zhengzhou 450052, Henan, China*)
[2](*School of Information Engineering, Zhengzhou University, Zhengzhou 450052, Henan, China*)

Abstract With the wide deployment of P2P-based live streaming media system in recent years, the mismatch does exist between overlay network in application layer and physical network in base layer. To address this issue, in this paper, the NCSTO mechanism, a topology optimisation mechanism for streaming media system based on Vivaldi network coordinate algorithm, is proposed. By using dual sampling and sample filter, the NCSTO can effectively optimise the topology aiming at overlay network, and decrease the network mismatch, improve system performance efficiency as well as save the network bandwidth.

Keywords P2P streaming media Topology optimisation Network coordinate Dual sampling

0 引 言

当前,随着 P2P 技术、应用层组播技术的不断发展[1-4],大规模流媒体直播系统已经成为互联网上的一项主要应用。近几年出现了很多基于数据驱动 P2P 流媒体系统,其中 DONet/Coolstreaming[5,6] 首先提出基于数据驱动的方式构造覆盖网,它不组建和维护一个传输数据的明显拓扑结构,而是用数据的可用性去引导数据流,节点之间自由组织成 Mesh[7] 网,具有很强的动态性和抗抖动性,提高了用户的体验质量,促进了网络视频的发展,对后续的研究影响很大。

基于数据驱动的系统虽然提高了系统吞吐量,健壮性好,适合动态网络环境,但构造的覆盖网与底层网络存在拓扑失配问题[8],流量越多失配越严重,导致底层物理网络压力大,给骨干网带来不必要的负载,同时使 Mesh 网效率低下,影响播放质量。

针对拓扑失配问题,一些传统的拓扑优化方法主要依靠获取底层网络信息来实现,比如根据 IP 地址匹配、DNS 或 BGP 信息[9],2-hop 邻居信息[10] 或者直接定期进行时延测量除去慢连接[11],这些方法都有局限性,或者不能准确地进行拓扑发现,或者代价过高,不适合大规模动态环境。最近,有学者提出利用网络坐标来优化 Overlay 拓扑[12],主要根据时延信息把节点映射到一个欧几里德坐标系[13] 中,通过获取部分其他节点的时延和坐标信息(样本),对坐标不断的更新使其精确化(采样),使用坐标距离来预测实际时延,大大提高了拓扑发现的效率和准确性。当前网络坐标系统主要有两种机制,一种基于 Landmark,代表是 GNP[14] 和 PIC[15],另一种基于模拟物理系统,代表是 Vivaldi[16] 和 Big band[17]。其中 Vivaldi 使用模拟弹簧势能衰弱的机制来构造坐标,将要测量的节点间延时建模为两个质点间弹簧的伸长量,当弹簧系统势能达到最低状态时整个网络的测量误差就达到最小。它摆脱了 Landmark 的设置,简单有效,完全分布,更加精确并且能及时反应网络的变化[18]。开源的 P2P 文件共享系统 Azureus[19,20] 就使用由哈佛大学网络坐标小组[21] 提供的 Vivaldi 算法模块进行 DHT 路由遍历优化,使得 DHT 查找速度得到明显提高。虽然基于 Vivaldi 的网络坐标算法呈现出在延时测量和拓扑发现方面的优势,但其运行效率依赖于应用系统[22],在 P2P 流媒体系统上还没有提出使用网络坐标的有效机制。另外,虽然一些 P2P 流媒体系统中提出了拓扑优化的方案,比如 AnySee 中使用了类似 LTM[8] 的机制使用测量延时来去除慢连接,但是他们的运行负载较高,不具有分布性、动态性。

综上所述,在现有的基于数据驱动的 P2P 流媒体系统中还没有找到拓扑优化的有效机制。针对此问题,本文提出了一种

收稿日期:2010-03-03。国家高技术研究发展计划项目(2008AA01A315)。林予松,副教授,主研领域:下一代互联网,应用层组播。

适用于大规模流媒体系统的拓扑优化机制 NCSTO,其中的“双重采样”和“样本过滤器”能够将 Vivaldi 坐标算法“低载高效”地运用在应用系统中。

1　设计方案

使用网络坐标对基于数据驱动的 P2P 流媒体系统进行拓扑优化,必须解决两个问题: 一个是如何有效地建立坐标,更新坐标; 另一个是如何提高网络坐标在系统中的作用力,既保证效率又不会给系统带来多余负载。在 NCSTO 中,针对上述第一个问题,提出“样本过滤器”SF(Sample Filter) 来为坐标更新选择有效样本; 针对第二个问题,提出了“双重采样”机制 DS(Dual Sampling) ,结合 SF,前期使用主动采样/快速更新算法 AFS(Active Fast Sampling) ,主动采样使得坐标快速收敛; 后期使用被动采样/轻量更新 PLS(Passive Light Sampling) 算法,将坐标更新过程附带在应用流量中,降低系统负载。

整个方案构造了一个适用于基于数据驱动 P2P 流媒体系统的网络坐标系统,对覆盖网进行拓扑优化。下面对样本过滤器 SF 和双重采样机制 DS 进行具体介绍。

2　SF 的设计和实现

Vivaldi 虽然在实验中证明了样本选择对于坐标收敛的重要影响[16] ,即选择精度较高的远近结合的样本能够提高采样效率和准确性,但由于 Vivaldi 算法采样过程依赖于应用系统,没有显式提出选取样本的方法。本文提出了一个样本过滤器 SF,通过计算样本的样本级别 SG(Sample Grade) 值选择有效样本集。

2.1　SG 值算法设计与实现

算法根据样本信息,包括距离 d、准确度 e 和坐标寿命 age 来计算 SG 值。SG 值越大说明样本越符合精度和距离优质的条件。

SG 算法伪代码描述如下:

```
//对于样本 j,i 对其计算坐标距离 d = ‖x_i − x_j‖ ,c_SG为调整常
//数。输入: 样本 sample_j; 输出: j 的 SG 值 ComputeSG( sample_j)
{
//计算距离权重 dw,根据 d 计算远近权值,其中 RT 为一个距
//离阈值,超过该值计为远节点。ε 为一小正数。
dw = √((d − RT)^2 + ε) % (RT^2 + ε)
//根据 e,age 和 dw 对于样本的影响计算 SG
SG = c_SG × dw^2 / (e(e + dw)) × age
Return SG
}
```

2.2　SF 算法的实现

SF 算法根据上小节计算出样本的 SG 值,对需要过滤的样本进行样本筛选,为坐标更新提供指定数量的有效样本集。

SF 算法伪代码描述如下:

```
//输入: 样本集 SampleSet,样本个数为 N,
//DisSum: 样本集中样本距离和; n 为筛选的样本数
SF ( SampleSet, n)
{
DisSum = GetSumDistance( SampleSet)
//取距离平均值为距离阈值
RT = DisSum / N
foreach sample in SampleSet
{
If ( d = 0)
//坐标相同指数为 0
SG = 0
Continue
ComputeSG( sample)
}
SortBySGDec( SampleSet)
Return SampleSet( n)
}
```

3　DS 的设计和实现

目前 Vivaldi 算法获取样本信息的过程依赖于应用系统,虽然这种被动的方式可以达到低负载(比如,在 Azureus 中的实现[23]) ,却降低了坐标收敛速度和准确度。本文提出“双重采样”机制——DS,来提高坐标系统使用效率,实现“低载高效”。DS 把流媒体系统运行过程分为前期,即播放频道前和后期,即播放频道中。前期使用 AFS 算法,主动获取样本,快速对坐标进行更新。后期使用 PLS 算法,在应用流量中被动获取样本进行坐标更新。其中都结合 SF 进行样本选择。

3.1　AFS 算法

AFS 中,节点根据独立于播放频道的全局节点 RC (Remote Contact) 主动进行样本选择和坐标更新的迭代过程,使坐标快速收敛,目的使节点在选择频道播放之前就已经获得坐标定位,尽可能在运行数据请求调度时有较准确的坐标。

算法简单描述:

新节点加入后立即向列表服务器请求返回 K 个节点资源,对这 K 个节点进行采样并获得他们的 RC_k,计算出初始粗略坐标 C_{np}和初始误差 E_{np}。然后用 C_{np}从 K 个节点中根据 $SF(RC_k, m)$ 选出 m 个节点,使用他们的 RC_m,根据 $SF(RC_m, m)$ 选出 m 个节点进行采样更新 C_{np},依次迭代下去,直到迭代次数大于阈值 W。

算法伪代码描述:

```
//StepNum 为迭代次数,SL 为样本列表
//输入: K, m
ASF ( K, m)
{
//初始化节点坐标和精度
InitCnpEnp( C_np, E_np)
RC_k = BootFromListServer( K)
SL = GetSampleInfo( RC_k)
UpdateCoordinate( C_np, E_np, SL)
CandidateNodes = SF ( RC_k, m)
StepNum = 0
do
{
    StepNum ++
    foreach node in CandidateNodes
    {
    NodesThisStep += GetRC( node)
}
```

```
CandidateNodes + = NodesThisStep
NodesForSample = SF ( CandidateNodes, m)
SL + = GetSampleInfo( NodesForSample)
UpdateCoordinate( C_np, E_np, SL)
RC = SF ( RC + NodesThisStep, MAX_NRC)
CandidateNodes = NodesForSample
} while ( StepNum < W )
}
```

3.2 PLS 算法

在数据驱动的 P2P 流媒体系统中,节点在播放过程中会不定期的进行"邻居重选"PR(Parter Reselection) [6],在此过程中从邻居列表 mCache 中建立新的连接关系 Partership。PLS 就在该过程获取样本信息进行采样,减少系统负载,同时坐标更新可以针对本频道进行优化。

算法简单描述:

启动过程 播放节点选择频道,根据 RC 中的频道号 *NCid* 找出该频道资源节点并保存在 *mCache* 中。请求节点在 *mCache* 中根据坐标距离选择 *r* 个邻近节点对其连接建立 *Partnership*,并根据数据资源表 *BufferMap* 的调度算法选取父节点获取数据块,播放流程开始执行。

播放期间 在每次进行 PR 时选择 *r* 个新节点连接 *New-Connects*,使用 SF(*NewConnects*,*m*) 选择 *m* 个样本采样,以此来进行坐标被动更新。

算法伪代码描述:

```
//输入: r ( PR 中要进行邻居重选的节点数)
//I 为初始化节点数的最小值
PLS( r)
{
    mCache = GetSourcePeers ( RC)
    if( mCache. num < I)
    mCache = mCache + GetSourcePeers( ListServer)
    NearCacheList = GetNearPeerByCoordinate( C_np, mCache)
    ParterList = CreatePartnership( NearCacheList)
    BufferMapDispatch( ParterList)

    While ( DataStreaming( ParterList) )
    {
        If ( NeedParterReselection( ) )
        {
            NewConnects = DoParterReselection( r)
            ConnectsForSample = SF ( NewConnects, m)
            ParterList = CreatePartnership( NewConnects)
            Neighbours + = GetSampleInfo( ConnectsForSample)
            UpdateCoordinate( C_np, E_np, SL)
            BufferMapDispatch( ParterList)
        }
    }
}
```

4 性能测试与分析

本文使用. NET 平台建立了一个基于数据驱动的 P2P 覆盖网仿真平台——NcstoSim,在该平台上实现了 NCSTO 机制。通过 Coordinate、Vec、Sample、SamplemMgr、VivaldiClient、NcstoClient 等几个核心类模拟了网络坐标系统在应用系统中的运行过程。仿真试验分为三个部分,第一部分测试 SF 的性能; 第二部分测试 DS 的性能; 第三部分测试 NCSTO 进行拓扑优化的性能。在三个试验中,Vivaldi 算法中的常数 ce 为 0.5,cc 为 0.25,SG 值算法中的 csg 为 0.25,ε 为 0.01,样本规模分为六种类型: 50、100、200、500、1000、1500,试验统计次数为 50 次。

在实验中发现,坐标更新最后都会收敛到一定精度(稳定状态) ,此时精度在 0.3 ~ 0.4 内,以下的实验结论主要是在达到稳定状态时取得。

4.1 SF 性能评估

本实验对比 SF 采样更新与随机采样更新的效率,实验中选择样本的数目 FilterNum 值分为三种类型: 5、8、10,测量指标为采样更新次数。

如图 1 所示,SF 采样比随机采样效率提高较大,平均更新速度提高近 30%,特别是在样本数目多并且 FilterNum 值大的情况下效果更明显,例如在 500 个节点,FilterNum 为 10 的情况,提高了近 50%。

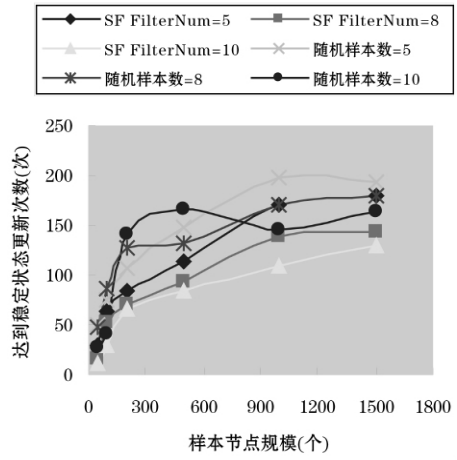

图 1 SF 采样与随机采样对比

4.2 DS 性能评估

本实验对比 ASF + PLS 综合运行,即 DS 方式和 PLS 单独运行的性能。实验中 SF 的 FilterNum 值为 8,ASF 迭代阈值 W 为 10,PLS 邻居重选数 r 为 15,测量指标为达到稳定状态的时间。如图 2 所示,使用 ASF 主动采样对坐标更新效率提高有很大影响,当样本数目多时更加明显。可以明显看出 DS 较 PLS 单独运行提高坐标更新速度约两倍。

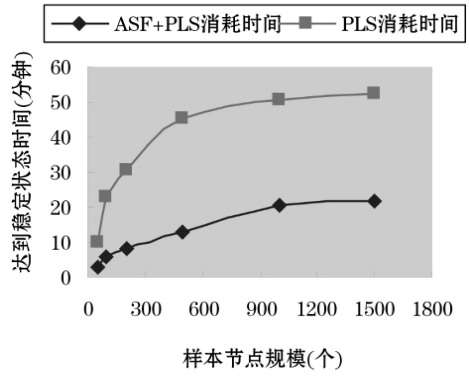

图 2 ASF + PLS 与 PLS 更新速度对比

4.3 NCSTO 拓扑优化性能评估

本实验对比使用 NCSTO 和随机方式进行邻居节点选择的

性能。实验中选择的邻居节点数目 *SourcePeerNum* 值分为两种类型: 4、8,测量指标为选择出节点的 RTT 值。

如图 3 所示,使用 NCSTO 的坐标距离选择出的节点比随机选择的节点的邻近性高很多,RTT 距离降低了近 40%,这说明 NCSTO 能有效地优化覆盖网拓扑。

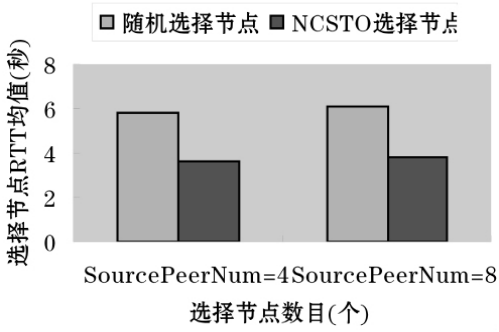

图 3　NCSTO 节点选择与随机节点选择对比

5　结　语

本文提出了一种适用于基于数据驱动的 P2P 流媒体系统的拓扑优化机制——NCSTO,其中的"双重采样"和"样本过滤器"将 Vivaldi 坐标算法"低载高效"地运用在应用系统中。通过仿真实验表明,NCSTO 能显著提高网络坐标在基于数据驱动的 P2P 流媒体系统中的利用效率,能有效地针对覆盖网的构造进行拓扑优化,提高系统运行效率,降低带宽浪费。

参　考　文　献

[1] Chu Y H, Rao S G, Zhang H. A case for end system multicast [C] // Proc. of ACM SIG-METRICS, (Santa Clara, CA), June, 2000, 20: 1456 – 1471.

[2] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast [C] //Proc. of ACM SIGCOMM02, August 2002: 205 – 217.

[3] Duc A, Tran K A H, Do T ZIGZAG. An effcient peer-to-peer scheme for media streaming [C] //Proc. Of Infocom'03, 2003, 2: 1283 – 1292.

[4] Lin Y S, Wang B Q, Wang Z M. MixCast: A New Group Communication Model in Large-scale Network [C] //Proc. Of the IEEE 19th International Conference on Advanced Information Networking and Applications. March, 2005, 2: 307 – 310.

[5] Zhang X, Liu J, Li B. DONet/Coolstreaming: A datadriven overlay network for live media streaming [C] //Proc. Of IEEE Infocom'05. Mar, 2005, 3: 2102 – 2111.

[6] Xie S, Li B, Keung G. Coolstreaming: design, theory, and practice [J]. IEEE Transactions on Multimedia, 2007, 9: 1661 – 1671.

[7] Magjarei N, Rejie R. Understanding Mesh-based Peer-to-Peer Streaming [C] //Proc. of NOSSDAV, 2006.

[8] Liu Y, Xiao L, Liu X. Location Awareness in Unstructured Peer-to-Peer Systems [J]. IEEE Trans. on Parallel and Distributed Systems, 2005, 16: 163 – 174.

[9] Padmanabhan V N, Subramanian L. An Investigation of Geographic Mapping Techniques for Internet Hosts [C] //Proc. Of ACM SIGCOMM, 2001: 173 – 185.

[10] Liu Y H, Esfahanin A H, Xiao L. Approaching Optimal Peer-to-Peer Overlays [C] //Proc. Of 13th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (IEEE MASCOTS 2005), Atlanta Georgia, USA, September, 2005: 407 – 414.

[11] Liu Y H, Zhuang Z Y, Xiao L. AOTO: Adaptive Overlay Topology Optimization in Unstructured P2P Systems [C] // Proc. Of IEEE GLOBECOM, San Francisco, USA, December, 2003, 7: 4186 – 4190.

[12] Pietzuch P, Ledlie J, Mitzenmacher M. Network-Aware Overlays with Network Coordinates [C] //Proc. of ICDCSW'06, 2006: 12.

[13] EuclideanWiki [EB/OL]. 2008. http://en.wikipedia.org/wiki/Euclidean.

[14] Ng T S E, Zhang H. Predicting Internet Network Distance with Coordinates-Based Approaches [C] //Proc. of INFOCOM'02, 2002, 1: 170 – 179.

[15] Costa M, Castro M, Rowstron A. PIC: Practical Internet Coordinates for Distance Estimation [C] //Proc. of ICDCS'04, Tokyo, Japan. Mar, 2004: 178 – 187.

[16] Dabek F, Cox R, Kaashoek F. Vivaldi: A Decentralized Network Coordinate System [C] //Proc. of SIGCOMM'04, Aug, 2004, 34: 15 – 26.

[17] Shavitt Y, Tankel T. Big-Bang Simulation for Embedding Network Distances in Euclidean Space [C] //Proc. of INFOCOM'03, San Francisco, CA. Mar, 2003, 12: 993 – 1006.

[18] 史晓辉, 陈阳, 邓北星. 网络坐标计算算法的实现研究 [C] //第十三届信息论学术年会论文集, 2005: 262 – 265.

[19] Azureus [EB/OL]. 2008. http://azureus.sourceforge.net/.

[20] AzureusWiki [EB/OL]. 2008. http://azureuswiki.com/.

[21] Network Coordinate Research at Harvard [EB/OL]. 2008. http://www.eecs.harvard.edu/~syrah/nc/.

[22] Ledlie J, Gardner P, Seltzer M. Network Coordinates in the Wild [EB/OL]. 2007. http://www.usenix.org/event/nsdi07/tech/full_papers/ledlie/ledlie_html/wild-web.html.

(上接第 95 页)

应用。为了满足门户运行和维护的需求,门户系统提供了多种服务。门户中子门户众多,所需服务数量众多。而且,随着子门户数量的不断增加,服务的数量也在不断增加。因此,要制定相应的策略,实现对门户中服务的有效管理。本文提出了一种基于事件的服务管理模型,引入事件驱动机制实现对服务的动态加载和卸载,并根据该模型设计了一种基于事件的服务管理框架。

门户中间件 OncePortal 采用该框架,对门户系统中的服务进行管理,提高了服务的利用效率,节省了门户系统的资源,减轻了门户系统的负担,从而实现了对服务的高效灵活的管理。

参　考　文　献

[1] 江泓. 门户系统中的服务动态重配技术研究 [D]. 北京: 中国科学院软件研究所, 2006.

[2] 匡芳君. 基于 Web 服务的事件驱动集成模型及应用 [D]. 长沙: 长沙理工大学, 2006.

[3] 刘家红, 吴泉源. 一个基于事件驱动的面向服务计算平台 [J]. 计算机学报, 2008, 31(4): 588 – 599.

[4] JSR168: Portlet Specification1.0 [S]. Java Community Process, 2003.

[5] JSR268: Portlet Specification2.0 [S]. Java Community Process, 2008.

[6] 中国科学院软件研究所. 网驰平台门户中间件(OncePortal) [EB/OL]. 2007. http://www.once.org.cn.