

P2P 流媒体系统服务节点选择机制研究与仿真^①

王 晴, 张晋豫, 宋晓莉

(北京交通大学 计算机与信息技术学院, 北京 100044)

摘 要: 根据 IP 地址的特性, 提出一种节点间静态测距法. 同时考虑节点内容的相似性、网络节点位置、稳定性、传输能力等因素, 根据节点数据的传输时延的特点, 提出一种选择服务节点的算法. 经实验和仿真证明, 文中所提的节点选择机制能较大提高 P2P 流媒体系统性能, 优化系统路由.

关键词: IP 地址; 静态测距; 服务端节点; 节点选择

Research and Simulation for Service-Peer Selection Mechanism on P2P Streaming System

WANG Qing, ZHANG Jin-Yu, SONG Xiao-Li

(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Based on the characteristics of IP address, this paper proposes an algorithm which can statically measure inter-peer distance. And considering these factors such as coherence of peer contents, the locality of peer, stability and transmission capacity, we propose an algorithm of choosing service node based on the characteristic of data transmission delay of inter-peer. Simulation results show that our algorithm can enhance performance of P2P system and optimize the routing of P2P system.

Key words: IP address; statically measure distance; service-peer; peer selection

1 引言

随着网络用户的爆炸式增长, 网络视频用户数量呈现指数级的上涨趋势. 并且, 随着网络用户对多媒体的需求日益增加, 以 P2P(Peer-to-Peer)技术为核心的流媒体系统也得到了广泛的应用^[1]. 在 P2P 流媒体系统中, 新加入节点所需内容均从其他节点获取, 为保证流媒体播放的流畅性, 保证流媒体系统的服务质量, 需要选取合适的服务节点. 因此服务节点是决定用户的媒体服务质量和网络业务量状态的关键因素, 节点选择是获取视频流的重要前提, 如何在 P2P 流媒体系统中选取合适的服务节点是国内外一个研究热点.

2 节点选择机制研究现状

P2P 流媒体系统中的节点选择机制是指当有一个节点进入系统时, 系统中已存在多个节点, 且存在一组节点具备该节点所需的内容, 系统将根据某种决定策略从那组节点中为该节点选择邻居节点. 该节点进

一步与所选的邻居节点建立连接, 获取所需内容, 这种决定策略即为节点选择机制^[2].

目前 P2P 流媒体的主流节点选择机制分为以下三种类型:

(1) 随机节点选择机制: Coolstreaming 系统、Peercast 系统均使用的是此机制. 该机制在选择服务节点时, 从候选节点集列表中随机选择一定数量的节点作为服务节点. 其优点是可以平衡网络中的负载, 但缺点是不考虑节点的其他能力因素, 选择的节点往往不是最优节点.

(2) 带宽优先选择机制: Promise 等系统使用的是此选择机制, 带宽优先选择机制也是目前 P2P 流媒体系统中使用最多的节点选择机制. 这种机制综合考虑节点能力, 如带宽、时延、丢包率等网络参数, 其优点是可以提升用户体验, 但其缺点是等待延迟过长、覆盖网的压力过大^[5]等.

(3) 网络拓扑聚集性选择机制: 华中科技大学开发

^① 收稿时间:2016-05-09;收到修改稿时间:2016-06-06 [doi:10.15888/j.cnki.csa.005570]

的 Any See 视频直播系统^[5]选用的此选择机制. 这种机制可以解决 P2P 网络中的拓扑失配问题, 但其缺点是流服务质量较差.

本文通过研究分析多种节点选择机制, 发现传统的随机选择机制和网络拓扑聚集性机制未将节点的综合能力纳入考虑范围, 这两种选择机制所获得的流媒体视频质量较差. 带宽优先选择机制则没有考虑节点的位置信息, 当请求节点距离较远时, 跨越骨干网时会产生较多的流量, 并且网络时延相对较大. 同时还发现, 现有的机制未能解决上层逻辑结构与底层物理拓扑结构不匹配的问题. 针对发现的问题, 本文提出一种基于 IP 地址和节点综合能力的节点选择机制, 优先选择与请求节点距离近、能力强的节点. 该机制可以较好地优化系统路由, 提升系统整体性能, 优化用户体验.

3 基于IP地址的节点选择机制

本文所提出的节点选择机制运用在以 Mesh 技术为基础的 P2P 流媒体系统上, 综合考虑 IP 地址的特性和评估节点的能力值, 选取最优节点. 基于 IP 地址的节点选择机制包含三个部分: 服务节点位置选择策

$$d_j^i = \begin{cases} \Delta D_G(i, j) + k(\sum_x^{i,j} (\sum_{y=1}^{4(16)} (|ip_x^p(y) - IP_x^{Gate}(y)|) * 256^{y-1})), & \Delta D_G(i, j) \neq 0 \\ k(\sum_{y=1}^{4(16)} (|ip_i^p(y) - IP_j^{Gate}(y)|) * 256^{y-1}), & \Delta D_G(i, j) = 0 \end{cases} \quad (1)$$

式中 d_j^i 表示节点 i 和节点 j 之间的距离, 其结果用时延表示; $IP_x^{Gate}(y)$ 表示节点 x 在其所在运营商网络中网关 IP 中的第 y 位; $ip_x^p(y)$ 表示节点 x 所在公网 IP 的第 y 段, 其中 IP 地址从最右端开始计算, IP_{v_4} 地址字段为 4, IP_{v_6} 为 16; k 表示 IP 地址差值所对应的时延.

约束条件 $\nabla D_G(i, j)$ 表示节点 i 和节点 j 所在的运营商网络的网关之间的距离, 其结果用时延表示. 当 $\nabla D_G(i, j) \neq 0$ 时, 表示节点 i 和节点 j 处于不同的运营商网络, 它们之间的距离等于运营商网络网关之间的距离与两个节点到其网关的加和距离之和. 当 $\nabla D_G(i, j) = 0$, 表示节点 i 和节点 j 处在同一运营商网络, 则它们之间的距离等于节点到网关距离之差.

当需计算处于局域网中的节点之间的距离时, 由于互联网间的路由机制, 节点的内网地址 NAT(Network Address Transfer)被网关、防火墙屏蔽, 因此节点对外显示的公网 IP 总是相同的, 所以应按照式(1)的第 2 种情况计算, 即处于同一网关的两个节点

略、服务节点综合能力值评估策略以及节点任务量分配策略.

3.1 服务节点位置选择策略

解决节点间拓扑不匹配的关键问题是节点构建拓扑时加入距离元素, 然而由于 Internet 在应用层屏蔽了物理链路的细节, 节点间的物理距离无法从用户终端获得; 同时, 又由于 IP 网络属于自治域网络, 缺乏全局的管控机制, 节点间的物理距离也无法从全局得到. 因此, 本文根据 IP 分配时的规律性, 静态测量节点间的距离.

由于互联网的开放性, 运营商拥有的网段和网关 IP 地址均对外界公开. 运营商之间的网络通过网关进行连接, 通常情况下, 采取测试或者查找等手段, 可以获得运营商网关之间的距离. 运营商网络中 IP 地址的分配一般按照地域进行规划, 具有显著的位置特征, 即给不同的区域分配不同的网段. 因此, 一般情况下, 若两个节点之间 IP 地址的区别越大, 则两点间的距离就越大.

基于以上规则, 本文提出一个节点间物理距离的估计公式, 并通过它来表示请求节点 i 和服务节点 j 之间的传输难易程度, 如式(1):

之间距离为 0.

3.2 服务节点综合能力值评估策略

3.2.1 轮询机制

本文中流媒体系统分成若干个轮询周期, 每个周期包含若干数据包, 系统中所有节点均采用轮询机制请求资源, 节点的数据传输过程如图 1 所示.

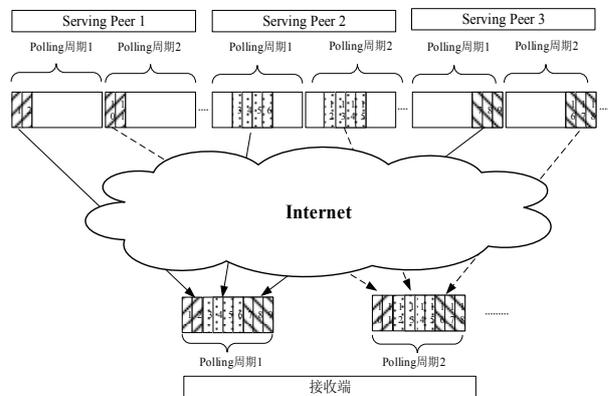


图 1 节点资源传输示意图

对于一个请求节点 i , 轮询周期包含两个参数: T_i 和 L_i . T_i 表示一个周期的时间, L_i 表示一个周期所包含的数据包数. 在使用轮询方式的 P2P 流媒体系统中, T_i 的大小受网络随机业务量的影响是不确定的, 而 L_i 的大小则是确定的.

由于流媒体系统一边下载一边播放, 请求节点使用环形缓存存储服务节点传送的信息. 如果下载速度过分大于播放速度, 则下载点可能会超过播放点一周而覆盖没有播放的内容, 破坏播放的连续性; 如果下载速度过小, 则播放点可能会追上下载点, 造成播放停顿. 因此, 为了实现流畅播放, 本文假定下载速度等于播放速度.

为了防止播放点超过下载点, 在异常情况下, 流媒体系统能够为调整措施提供足够的反应时间, 需在下载点和播放点之间建立缓冲区, 因此本文定义下载点和播放点之间的最小间距为 C 个轮询周期. 基于以上的定义和假设, 对于一个请求节点 i , 假设节点的媒体播放速度为 v^i , p 为分组长度, 其单位为 byte, 则轮询周期 L_i 的长度为:

$$L_i = \text{int}((v^i * T_s) / (8 * p * C)) \quad (2)$$

3.2.2 节点性能参数定义

为评估一个服务节点的综合能力值, 需为每个节点维持一个性能参数统计集合. 假设存在一个服务节点 j , 则它的性能参数统计集合为:

$$\{T_0^j, N_T^j, \overline{TOA^j}, \overline{T^j}, \overline{S^j}\}$$

集合内各项性能参数的定义如下:

1) 由于服务节点缓存的资源与登录时间有关, 故用 T_0^j 表示服务节点 j 的登录时间. 为了保证流媒体播放的流畅性, 需要一定的启动时间, 本文定义启动时间为恒定值 T_s .

2) N_T^j 表示服务节点 j 的可用子流数, 即用服务节点的可用子流数表示节点在同一时刻的服务能力. 此参数随着请求任务的节点数动态变化, 当一个节点收到其他节点的资源请求任务时, 其可用子流数减一, 即 $N_T^j = N_T^j - 1$.

3) $\overline{TOA^j}$ 表示服务节点 j 上行传输的统计带宽.

4) $\overline{T^j}$ 表示服务节点 j 为其他节点服务时长的统计值. 当一个节点在系统中的时间越长, 其为其他节点提供服务的机率就越大, 即其贡献度越大.

5) 由于 P2P 网络中的节点的动态性和用户观看

流媒体视频资源的随机性, P2P 网络拓扑总是处于动态变化中, 在选取服务节点时, 选择稳定性高的节点有利于稳定获取资源. 故用 S^j 表示服务节点 j 的稳定度.

节点的上传带宽越大, 服务能力越强; 节点越稳定, 提供的资源越多; 服务节点与请求节点的距离越近, 网络时延越小. 因此基于以上提出的性能参数, 本文提出一个计算上传带宽、稳定性和距离三个性能参数的节点能力值公式, 如式(3)所示:

$$P_i^j = \alpha_1 \frac{\overline{TOA^j}}{\text{Max}\{\forall x \in [1, n]: \overline{TOA^x}\}} - \alpha_2 \frac{d_i^j}{\text{Max}\{\forall x \in [1, n]: d_i^x\}} + \alpha_3 \frac{\overline{S^j}}{\text{Max}\{\forall x \in [1, n]: \overline{S^x}\}} \quad (3)$$

其中, $\alpha_1, \alpha_2, \alpha_3$ 为调节系数.

$\alpha_1 \in [0, 1], \alpha_2 \in [0, 1], \alpha_3 \in [0, 1], \alpha_1 + \alpha_2 + \alpha_3 = 1$, n 表示除请求节点外, 系统中存在 n 个其他节点.

3.2.3 选取过程描述

综上所述, 根据节点的综合能力值选取服务节点的过程具体描述如下:

(1) 服务节点的可用子流数必须大于零, $N_T^j > 0$. 一个服务节点在同一时间只能为有限个节点提供资源, 在选取服务节点时, 需对可用子流数进行判断, 仅选择可用子流数大于零的节点.

(2) 选取请求内容相近的服务节点. 由于用户观看视频流资源的随机性, 每个节点缓存的内容具有异性和短暂性, 在选取服务节点时, 需选择缓存内容中与请求资源相近的节点. 在选择内容相近的节点时, 服务节点 j 的登录时间 T_0^j 需满足:

$$T_0^j > T_s \text{ 或 } T_0^j \leq \frac{C * L * p * 8}{v^j}$$

即请求节点 i 的下载点 D 不落后于服务节点 j 的播放点, 或者在服务节点 j 的下载点与播放点之间, 则称服务节点 j 和请求节点 i 内容相似.

确定服务节点与请求节点的内容相似后, 将服务节点 j 加入请求节点 i 的候选节点列表 Ω_i 中.

(3) 重复过程(1)(2), 直至将所有满足内容相似的服务节点选择完毕.

(4) 根据公式(3), 将请求节点 i 的候选列表 Ω_i 中所有候选服务节点按照综合能力大小进行从大到小排序. 当 $\sum_{j=1}^m \overline{TOA^j} \cdot \text{length} \geq L$ 时, 即当前 m 个服务节点的

TOA^i 长度之和大于 L 时, 将前 m 个服务节点构成新的服务节点列表 M_i .

(5) 将选出的候选节点列表 M_i 返回给请求节点 j .

3.3 服务节点任务量分配策略

本文规定, 在进行服务节点的选择同时分配服务节点的任务量, 不同的服务节点分配不同的任务量. 在一个轮询周期中, 一个请求节点会将数据分组的下载任务依次分配给它的每一个服务节点. 鉴于数据之间的连续性, 同时为了减少授权次数, 降低管理开销, 本文的任务量分配策略采用连续任务量分配机制, 即给每一个服务节点授权连续的数据包.

在一个轮询周期内, 称分配给某一个服务节点的连续数据包为服务节点的例行任务量, 即 TOA. 规定一个客户端的所有服务节点的 TOA 交集为空, 一个服务节点在两个相邻轮询周期中 TOA 的位置相同, 其距离为一个轮询周期的长度. 在每个轮询周期内, 服务节点只传输自己的 TOA, 传输完成后将跳到下一个轮询周期继续传输自己的 TOA, 这样就实现了服务节点之间的分工协作, 即:

$$\begin{cases} L_i = \bigcup_{j=1}^{M^i} TOA_j^i.length \\ \forall j, k \in [1, M^i], j \neq k : TOA_j^i \cap TOA_k^i = \Phi \end{cases} \quad (4)$$

其中, M^i 为接收端 i 的服务节点个数, TOA_j^i 表示请求节点 i 给服务节点 j 分配的 TOA, $(TOA_j^i).length$ 表示它的长度.

IP 网络采用基于存储转发的中间节点独立选路的路由机制, 在同一网络环境中, 两个节点的距离越远, 其业务受到拥塞的概率越高, 网络时延越大, 传输能力也随之降低. 基于以上原因, 本文提出一个根据网络距离的远近为节点分配任务量的算法.

由于一个服务节点支持有限的上行并发子流, 若超过其限制, 每个子流带宽和性能就要下降, 因此设定一个节点的并发子流上限值 N_p , 则请求节点 i 给其服务节点 j 分配任务量 TOA 的公式为:

$$TOA_j^i = (\text{int})\left(\left(\frac{W_1 - W_2}{d_{\max}}\right) * (d_{\max} - d_j^i) + W_2\right) / N_p \quad (5)$$

节点之间距离越大, 则所分配的任务量越小. 其中, $[W_1, W_2]$ 为请求节点感知的服务节点的 TOA 的范围. 每一个服务节点在注册时, 被赋予最大的 TOA 值 W_1 , 也就是 $d_j^i = 0$ 局域网内传输能力. 在局域网中的距离

最大时, 即:

$$d_{\max} = K \sum_{y=1}^{4(16)} 256^y \quad (6)$$

设网络的传输带宽为 w_0 , 通常情况下用居民平均国际带宽表示, 则它对应最小的传输能力时的 TOA 的值为 W_2 :

$$W_2 = \text{int}((w_0 * t) / (8 * p * C_1)) \quad (7)$$

4 仿真实验与分析

本文将上述算法应用于实验室开发的“基于 Mesh 的 P2P 流媒体业务系统”上, 本实验室开发的“基于 Mesh 的 P2P 流媒体业务系统”具有高传输、高质量等性能特点, 可为测试以网络拓扑结构为基础的节点选择机制提供一个较为理想的实验平台.

4.1 实验数据配置

基于 Mesh 的 P2P 流媒体业务系统的部分参数配置为:

设置每个节点支持的最大子流数为 $N=6$, $p=512$ 字节, $C=10$, $TOA_{\min}=5$, $T_S=1228ms$. 若请求节点与服务节点处于局域网中, 则认为只需一个服务节点即可满足传输能力的要求, 因此, $W_{\max}=L=210$, $W_{\min}=1$.

设置服务节点的 IP 采用 IPv4 版本, IP 的 32 个字段随机产生. 在机群中的节点假设具有异构带宽类, 即(512Kb, 128Kb), (768Kb, 256Kb), (1024Kb, 512Kb), (1563Kb, 768Kb), (2048Kb, 1024Kb), 括号中的成员从左至右分别表示节点的最大下行速度和最大上行速度.

4.2 仿真结果分析

(1) 测试点到 211.71.76.113(北京交通大学主机)的平均时延测试结果和路由跳数, 如图 2 和图 3 所示.

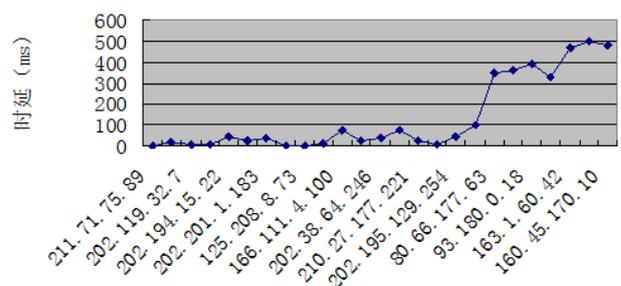


图 2 测试点一天的平均延时

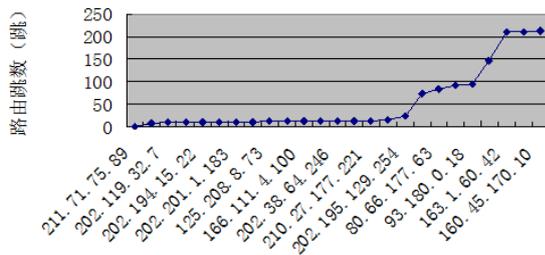


图 3 测试点一天的平均路由跳数

(2) 测试点到 211.71.76.113 的网络距离, 用文中公式(1)计算的网络距离, 如图 4.

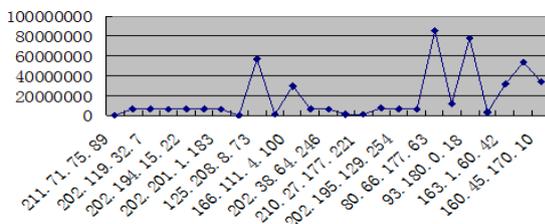


图 4 公式(1)计算得出的网络距离

实验结果表明, 平均跳数和网络时延同网络距离服从相似分布, 尤其是国内和国际节点之间的区别明显(以 80.66.177.63 为界).

(3) IP 距离公式的验证

通过 Ping 得到北京交通大学的主机(211.71.76.113)到测试主机的时延, 把一天内测得的多次时延的平均值作为距离的标准和由公式(1)计算出的值进行比较, 结果如图 5.

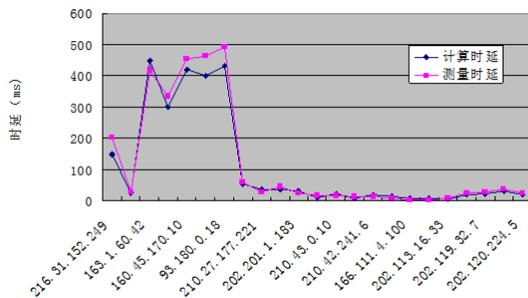


图 5 距离公式和实际对比

计算单位距离对应的时延值时, k 值为:

$$k = 1/133674.628$$

实验结果表明, 计算的网路距离和测量的平均网路时延在国际和国内(以 80.66.177.63 为界)的特征一致. 验证了公式(1)可以用来评估网络之间的距离.

(4) 服务节点选择机制性能比较

为了验证本文提出的节点选择机制具有可靠性, 证明根据本文提出的服务节点选择机制, 请求节点会优先选择距离近、能力值高的服务节点, 能优化系统网络路由, 将本文提出的服务节点选择机制与 PeerStreaming 系统^[7]的随机选择节点机制进行对比. 通过比较业务系统中数据包选择路由的跳数是否减少, 判断节点选择机制是否进行路由优化. 仿真结果如图 6.

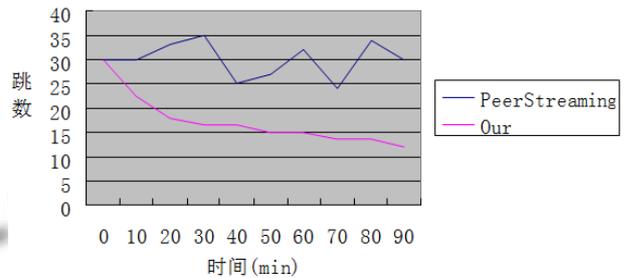


图 6 路由优化

PeerStreaming 系统中的节点采用随机选取模式选择服务节点. 当有 1000 个节点时, 随着时间的推移, 本文提出的服务节点选择机制选取的节点的路由跳数不断减小. 因为节点间的信息不断更新, 更容易选择近距离的节点提供服务. 而 PeerStreaming 系统没有位置意识, 路由跳数一直在波动变化, 因此本文提出的节点选择机制对系统性能具有较好的影响.

5 结束语

本文在对现有的 P2P 节点选择机制研究的基础上, 针对现有节点选择机制的问题, 提出一种基于 IP 地址和节点综合能力评估策略的服务节点选择机制, 在确定节点间距离的基础上, 进一步评估服务节点的综合能力值. 仿真结果表明, 相比于现有的算法, 本文提出的服务节点选择机制可以更好地解决网络拓扑中的失配问题, 降低系统的网络时延, 同时, 由于考虑服务节点的综合服务能力, 选出一组综合能力强的节点可以保证传输资源的稳定, 提升用户体验.

参考文献

- 1 郑婕,张松.基于 Mesh 的 P2P 流媒体节点选择机制研究.微电子学与计算机,2007,24(12):95-99.
- 2 郑婕,张松,雷迎春,齐洁.P2P 流媒体节点选择机制的研究与仿真.计算机工程与设计,2007,28(22):5396-5399.
- 3 庄雷,郭永平,孔燕,孙一潇.P2PCenter 节点选择机制的分析

- 及改进.郑州大学学报(工学版),2011,32(5):70-72.
- 4 李彦,王丽娜.P2P 流媒体系统中基于直觉模糊集的节点选择策略.计算机科学,2013,40(6A):280-282.
 - 5 何海亮,黄道颖,张安琳,李健勇,李建春.两段式 P2P 流媒体节点选择算法.计算机工程,2011,37(20):63-65,68.
 - 6 唐朝伟,肖俊,王恒,胡佩,刘倩男,宋俊平,李晓辉.异构环境下的 P2P 流媒体节点选择算法.中南大学学报(工学版), 2015, 46(9):3287-3293.
 - 7 Li J. PeerStreaming: An on-demand peer-to-peer media streaming solution based on a receiver-driven streaming protocol. 2005 IEEE 7th Workshop on Multimedia Signal Processing. 2005. 1-4.
 - 8 Liu NW, Wen Z, Yeung KL, Lei ZB. Request-peer selection for load-balancing in P2P live streaming systems. 2012 IEEE Wireless Communications and Networking Conference: Services, Applications, and Business. 2012. 3227-3232.
 - 9 Huang GM, Liu PS, Gong X. A novel peer selection strategy in P2P VoD system using biased gossip. 2015 IEEE International Conference on Communication Software and Networks. 2015. 372-377.