

一种分层 P2P 流媒体系统重叠网的构建策略

陈 卓^{1,2} 冯 钢¹ 陆 毅¹

(电子科技大学通信抗干扰技术国家重点实验室 成都 611731)¹

(重庆理工大学计算机科学与工程学院 重庆 400054)²

摘 要 基于 Peer-to-Peer(P2P)技术的流媒体应用具有部署效率高及可扩展性好等突出优势。而采用分层视频编码技术的 P2P 流媒体系统把原视频流分解为多个视频层数据进行分发,让节点能够选择与自己带宽资源相匹配的视频质量,使其很好地适应节点的异构性。但是各分层视频数据传输的路径存在较大差异,使重叠网构建策略面临更大的挑战。因此定义了基于分层视频编码技术环境下的 P2P 流媒体重叠网络构建问题,并证明该问题是一个 NP 难问题。提出了一种构建重叠网的集中启发式算法,同时还提出了一种基于视频组(Streaming Group)的分布式重叠网络构建策略。通过大规模网络仿真实验验证了基于该分布式重叠网络构建策略的分层流媒体系统具备低服务器带宽占用、高数据获取率等优点。

关键词 P2P,重叠网,分层视频编码,流媒体系统

中图法分类号 TP393.03 **文献标识码** A

Overlay Construction Policy in Layered P2P Streaming System

CHEN Zhuo^{1,2} FENG Gang¹ LU Yi¹

(National Key Laboratory of Communications, University of Electronic Science and Technology of China, Chengdu 611731, China)¹

(College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China)²

Abstract Peer-to-Peer(P2P)based streaming applications hold the advantages of high efficient deployment ability and high scalability. Layered video coding based P2P streaming system splits the source video into multiple layer of video data and distributes these streaming data, which makes peer can select the playback quality according to its bandwidth resource. This kind of streaming system is more accommodative to the heterogeneity of peers. However, difference distribution overlay path of different layer of streaming data also leads to challenges of overlay construction in layered P2P streaming system. This paper formulated the overlay construction issue in layered P2P streaming environment, which is a NP-hard problem, and put forward a centralized heuristic algorithm, then designed a streaming group based distributed overlay construction policy. Large scale network simulation shows that this overlay construction mechanism has the characteristics of comparatively low bandwidth consumption at streaming server and high streaming acquisition ratio.

Keywords P2P, Overlay network, Layered video coding, Streaming system

1 引言

基于应用层建立重叠网结构并通过 Peer-to-Peer(P2P)的方式实现流媒体分发,已经成为一种高效部署流媒体应用的主要方法。应用层 P2P 流媒体系统的主要优势在于其不受底层网络拓扑结构的限制,可以在应用层构建一张虚拟的重叠网结构对视频流进行分发。随着网络接入方式的多样化,节点的带宽资源也有较强的异构性,这使得一个优秀的 P2P 流媒体系统在进行数据分发时,更应该考虑让流媒体质量能适应带宽异构,因此基于分层视频编码技术(如: H.264/SVC)的 P2P 流媒体系统近年来引起很大关注^[1,2]。这类系统中,节点根据自己的带宽注册到不同的数据层以获取该层所对应的视频质量,分层视频数据通过重叠网结构进行

分发。另一方面,重叠网的构建策略对于 P2P 流媒体系统的系统性能的影响至关重要。一个好的重叠网络结构有助于提高节点的视频播放质量,同时降低对视频服务器的带宽资源消耗,优化视频流的分发路径。在单层 P2P 流媒体环境中,各节点获取的均是同一视频质量的视频流,因此节点在选择邻居时只需要依据可用带宽资源、节点间数据传输延迟等因素进行选择。而对于分层 P2P 流媒体系统,节点获得的视频层数据流越多,则表示节点的图像质量越好。一个节点进入重叠网时根据自身带宽确定注册的视频层,而影响该节点播放质量的主要因素不仅是邻居节点的可用带宽和延迟,还包括邻居节点自身的播放质量,具备良好播放质量的节点同时缓存了足够的视频流数据,因此也具备较强的服务能力,不同的分层重叠网络结构会①影响节点的播放质量;②导致视

到稿日期:2011-06-28 返修日期:2011-09-18 本文受国家科技重大专项(2008ZX03004),重庆市教委科技项目(KJ110831)资助。

陈 卓(1980—),男,博士生,讲师,主要研究方向为分布式网络、P2P 流媒体, E-mail: chen Zhuo@cqu.edu.cn;冯 钢(1964—),男,教授,博士生导师,主要研究方向为无线通信网络、分布式网络。

频服务器较大的带宽耗费^[2]。针对①的原因是,不同的视频层数据所分发的重叠网络路径存在差异,导致不同视频层数据到达各节点的先后次序存在较大的差异。然而,视频应用通常具备较强的实时性,这又要求不同视频流数据到达节点的时间差异需要满足一定的时间限制,否则会直接影响视频质量。这对分层 P2P 视频应用的重叠网结构也即邻居选择策略提出了更高的要求。而对于②,其原因是部分注册在较低层的节点处于离视频服务器较近的位置,从而阻塞了部分高层数据的分发,使得注册在较高视频层的节点只能通过直接向服务器请求来获得被阻塞的视频层数据。本文重点研究在分层视频编码环境下的 P2P 重叠网络构建策略。

2 相关研究工作

P2P 流媒体的重叠网结构可以分为两类:结构化和非结构化的。结构化的重叠网络结构有固定的拓扑结构,节点之间通常按照固有的树形结构实现视频流的分发,这种拓扑结构存在维护开销较大且当节点异常退出时影响较大等问题^[3]。因此实际的系统基本都使用网状重叠网拓扑结构,这种结构不需要维护严格的结构,节点随机地建立邻居关系形成网状结构的重叠网^[1]。文献[4]深入研究了分层流媒体环境下的视频调度策略,并给出了一个简单的重叠网络构建的方法。Chamelon 系统^[2,5]采用了一种基于依据带宽资源分组

的方式建立重叠网络拓扑,但由于探测节点之间的带宽需要耗费较长时间,使得这种重叠网络构建策略难以被实际系统采用。文献[4]提出一种基于带宽延迟比较的分层流媒体重叠网构建方法-OCals。该方法需要不断地通过 ping 的方式比较多个节点之间的相对距离,因此效率通常不够理想,且节点需要经过较长时间才能够建立起自己的邻居关系。

3 分层视频流环境下的重叠网构建问题

本节首先定义分层流媒体环境下的重叠网构建的问题。

定义 1 若节点 N_i 注册在 L_k 层,在时间段 Δt 内从其邻居节点获得的 L_m ($L_1 \leq L_m \leq L_k$) 视频层数据为 D_m 。当满足 $D_1=r_1, \dots, D_j=r_j, \dots, D_k=r_k, (1 \leq j \leq k)$ 时,称节点 N_i 达到完全服务状态(r_j 表示视频层 L_j 的流速率)。

假设视频源被分解为 3 个视频层: L_1, L_2 和 L_3 , 其视频流量分别为 $r, 2r, 3r$ 。有 n 个注册在不同层的节点共同建立重叠网,根据带宽资源状况,假设一共有 $n=11$ 个节点相继进入重叠网,节点 $N_1^{L_3}, N_2^{L_3}$ 注册在 L_3 层,其上行带宽为 $3r$ 。节点 $N_3^{L_2}, N_4^{L_2}, N_5^{L_2}$ 注册在 L_2 层,其上行带宽为 $2r$ 。而节点 $N_6^{L_1}, N_7^{L_1}, N_8^{L_1}, N_9^{L_1}, N_{10}^{L_1}, N_{11}^{L_1}$ 注册在 L_1 层,其上行带宽为 r (下标表示节点的编号,上标表示该节点期望注册的视频层)。节点进入重叠网的顺序随机,因此有可能产生多种重叠网结构,图 1 显示的是其中 3 种可能的重叠网结构。

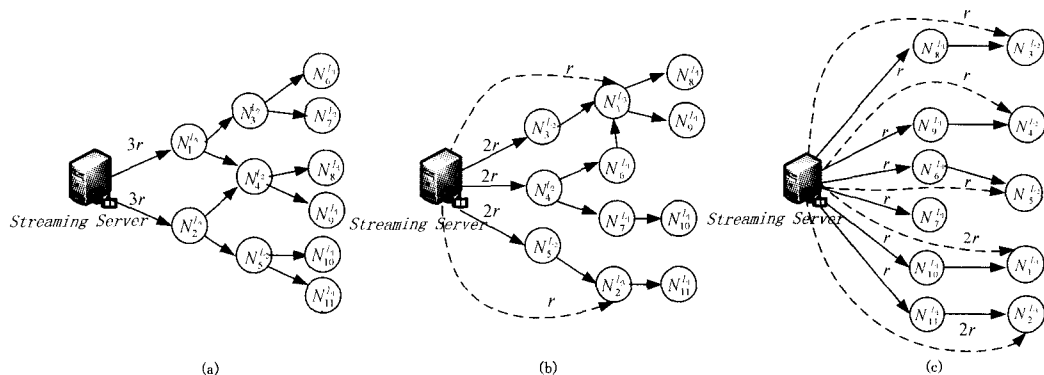


图 1 可能形成的 3 种不同的重叠网络结构

分层 P2P 流媒体系统中,一个节点要达到满意状态必须要获取到各视频层数据流数据,而由于视频流具备一定的实时性,如果节点在一段时间内无法从邻居节点处获取到(通常采用基于“拉”或者基于“推”的方式)部分视频层的数据,则需要向流媒体服务器获取,以达到完全服务状态。图 1 中的虚线即表示节点为了达到完全服务,而通过流媒体服务器获取的部分视频层数据,如:图 1(b)中 $N_1^{L_3}$ 从邻居处无法获得 L_3 层视频流数据则向视频服务器请求了 L_3 层视频流。通过图 1 所描述的 3 种不同的重叠网络结构,在保证重叠网节点都达到满意状态的前提下,视频服务器的带宽负荷有很大的差异。图 1(a)中视频服务器的负荷 C_s 为 $6r$,图 1(b)中节点对服务器的带宽需求 C_s 为 $8r$,而图 1(c)中节点对服务器的带宽需求 C_s 为 $13r$ 。这里假设重叠网络中的节点都愿意尽可能地利用自己的上行带宽为其它节点分层提供数据流服务,而不考虑节点自私性。从图 1 中不同的重叠网络结构导致对流媒体服务器负荷的显著不同可以分析得出,导致服务器负荷的主要原因是不同的邻居关系导致部分注册在低视频层的节点阻塞了部分高视频层数据流的分发,使得不同视频层数

据流到达节点的时间差异较大。为了降低节点向视频流服务器请求空缺视频数据的概率,其核心问题是建立一个重叠网络的拓扑结构,使不同层的视频流能以较小的时间差异到达节点。

4 重叠网构建的模型

通过图 $G(V, E)$ 表示重叠网络, V 为重叠网络节点集合(包括流媒体服务器 S 和节点), $E=V \times V$ 为节点间的连接集合。在基于 Mesh 结构的重叠网中,视频流数据仍然可以视作由多棵分发树(Delivery Tree)进行分发^[3]。通过重叠网构建的不同策略对视频流服务器的带宽需求来描述模型。

定义 2 视频源分为 $L_1, \dots, L_k$ 共 k 层,这些视频流通过重叠网 $G(V, E)$ 进行分发,对于其中某一个视频层 L_i ($1 \leq i \leq k$),视频流可视作通过一个单层重叠网 $G_{L_i}(V_{L_i}, E_{L_i})$ 进行分发,且有 $V_{L_k} \subset \dots \subset V_{L_i} \subset \dots \subset V, (E_{L_k} \subset E)$ 。

对于分层视频流的数据分发,设视频流分为共 k 层,这 k 层层分视频流都通过在重叠网 $G(V, E)$ 中通过 P2P 的方式实现分发。可以进一步抽象出 k 个单层重叠网 $G_{L_1}(V_{L_1}, E_{L_1})$,

$\dots, G_{L_1}(V_{L_1}, E_{L_1}), \dots, G_{L_k}(V_{L_k}, E_{L_k}), (1 \leq i \leq k)$, 如图 2 所示。一个注册在 L_k 层的节点需要同时获取 $L_1, \dots, L_k$ 层视频流, 因此该节点应同时属于 $G_{L_1}, G_{L_2}, \dots, G_{L_k}$ 这 k 个子重叠网 (Sub-Overlay)。

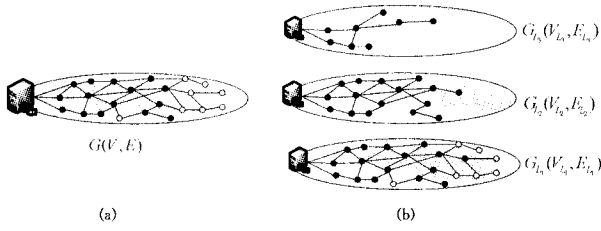


图 2 重叠网及按层次划分的子重叠网

定义 3 在一个单层重叠网 $G_{L_i}(V_{L_i}, E_{L_i}), (1 \leq i \leq k), L_i$ 层视频流通过 n_i 棵视频分发树 (生成树) $T_{L_i}^1, \dots, T_{L_i}^{n_i}, \forall T_{L_i}^j \in T_{L_i}, (1 \leq j \leq n_i)$ 把该层视频流数据分发至节点集 $V_{L_i}, (V_{L_i} \subset V)$, 每棵视频分发树分发的视频流为 $\lceil \frac{r_{L_i}}{\sigma} \rceil$, 假设 σ 为节点间最小传输速率。

由于各视频层的数据流实际仍然是在 $G(V, E)$ 中进行分发, 因此各层的数据分发树有可能共享部分节点间的连接。通过上面的假设, 若注册在 L_k 视频层的节点 $N_i^{L_k}$ 能够获取到来自 $T_{L_1}, \dots, T_{L_k}$ 这 k 个集合中的每棵分发树的流数据, 则该节点达到完全服务; 如果只获取到 $T_{L_1}, \dots, T_{L_k}$ 中部分视频分发树的数据, 则节点 $N_i^{L_k}$ 只能直接向视频服务器 S 请求。因此, 降低 S 的带宽负荷等同于节点能同时位于 $T_{L_1}, \dots, T_{L_k}$ 所有分发树上, 也即让节点 $N_i^{L_k}$ 的入度最大, 故节点 $N_i^{L_k}$ 达到完全服务且降低对服务器的带宽负荷问题归结为一个求最大值的最优化问题, 也即:

$$\text{Max}(d(N_i^{L_k})) \quad (1)$$

式中, $d(N_i^{L_k})$ 表示 $N_i^{L_k}$ 的度, 其约束条件为:

$$\begin{aligned} \textcircled{1} d(N_i^{L_k}) &\leq \sum_{j=1}^{L_k} n_j \\ \textcircled{2} \forall G_{L_i}(V_{L_i}, E_{L_i}), \sum_{j \in V_{L_i}} U_j &\geq (|V_{L_i}| - 1) \cdot r_{L_i} \quad (1 \leq i \leq k) \end{aligned}$$

该优化问题的约束条件②表示各单层视频重叠网中, 节点提供用于分发该视频层数据流的上行带宽之和, 能够满足节点对于该层视频流的流量需求之和。该优化问题是一个 NP 难问题, 可通过单层重叠网模型来分析此问题, 并定义各单层重叠网的权值为节点间的延迟。对于单层重叠网 $G_{L_1}(V_{L_1}, E_{L_1})$, 分发 L_1 层视频流涉及到建立 n_{L_1} 棵视频分发树, 每棵视频分发树需要连接从 S 总连接 $G_{L_1}(V_{L_1}, E_{L_1})$ 中所有节点 V_{L_1} , 且要求总的时间延迟最小。由于 $V_{L_1} = V$, 可以通过多项式时间内的最小生成树算法, 如 Prim 算法^[5]实现。对于 $G_{L_i}(V_{L_i}, E_{L_i}), (1 < i \leq k)$, 需要建立 n_{L_i} 棵视频分发树来连接节点集 V_{L_i} 。由于 $V_{L_i} \subset V$, 该问题等价于在 $G(V, E)$ 中建立 n_{L_i} 棵连接部分节点的斯坦纳最小树 (SMT)。而在给定节点集合中, 斯坦纳树的问题是一个 NP 难问题。所以在 $G(V, E)$ 中建立多棵延迟最小的视频分发树为一个 NP 难问题。

5 基于子重叠网合并的中心启发式算法 (Centralized Heuristic Algorithm)

由于为分层流媒体建立延迟最优的分发树问题是一个

NP 难问题, 因此提出一种基于合并树的集中启发式算法, 该集中式算法运行的条件是已经知道重叠网络中各个节点注册的视频层以及节点之间的延迟。设源视频流分解为 $L_1, \dots, L_k$ 一共 k 层, 而各层视频分发树的数量分别 $n_{L_1}, \dots, n_{L_k}$, 可以简单计算出 $L_i (1 \leq i \leq k)$ 上的视频分发树的数量为 $n_{L_i} = \lceil \frac{r_{L_i}}{\sigma} \rceil (1 \leq i \leq k)$, 其中, r_{L_i} 为视频层 L_i 的流媒体速率, σ 为节点间的最低传输速率。该集中启发式算法的基本思想是从视频流媒体服务器 S 开始, 依据节点间的延迟为度量标准, 利用求解最小生成树的 Prim 算法建立 k 个视频层的视频分发树。由于分层视频编码中层次的依赖关系, 使得低视频层数据的优先级高于高视频层数据, 该算法在建立视频分发树时也是按照先低层后高层的原则分别建立分发树。在分别得到 k 层视频分发树集合 $T_{L_1}, \dots, T_{L_k}$ 后, 再对各层视频分发树进行叠加, 这样节点间可能会合并部分连接, 也可能产生新的连接, 最后得到整个 $G(V, E)$ 上的分层视频分发树集合 T 。整个算法分为两个主要步骤, 即构造过程、合并过程。

Step1 Construction process:

Build $T_{L_1}, \dots, T_{L_k}$ under node uplink capacity constraint for a specific layer, each $T_{L_i} (1 \leq i \leq k)$ only fully deliver layer of L_i streaming data.

Step2 Combination process:

Combine $T_{L_1}, \dots, T_{L_k}$, may intergate some links and may add some links. Each node compare its end to end latency of different of different layers of streaming data and change some neighborships.

算法具体细节如下:

Construction Algorithm

Input:

odeset: $V, |V| = n$
node streaming layer registration set: R
node uplink bandwidth: $U_i, i \in V$
streaming server: S , streaming source is divided into k single layers
initial delay matrix: $D_{n \times n}$
single layer rate: $r_{L_1}, \dots, r_{L_k}$

Output:

$T_{L_1}, \dots, T_{L_k}$, each T_{L_i} is a single layer mesh, which can delivery data withtime efficiency.
for $1 \leq i \leq k$ do
 $T_{L_i} \leftarrow \phi$;
 $V_{L_i} \leftarrow \text{SelectNode}(V, L_i)$;
 $n_{L_i} \leftarrow \lceil \frac{r_{L_i}}{\sigma} \rceil$;
 foreach node $j \in V_{L_i}$ do
 bandforlayer[j] $\leftarrow U_j \cdot \frac{r_{L_i}}{\sum_{m=1}^k r_{L_m}}$;
 end
 for $1 \leq x \leq n_{L_i}$ do
 $t_{L_i}^x \leftarrow S$;
 while $t_{L_i}^x \neq V_{L_i}$ do
 $u \leftarrow \text{null}$;
 $v \leftarrow \text{null}$;

```

MinDelay ← MAX;
foreach node  $j \in t_{l_1}^x$  do
    foreach node  $\eta \in V_{l_1} \& \eta \notin t_{l_1}^x$  do
        if ( $D_j(t_{l_1}^x) + d_{\eta,j} < \text{MinDelay} \& \text{bandforlayer}[j] > 0$ )
            then
                MinDelay ←  $D_j(t_{l_1}^x) + D[\eta, j]$ ;
                 $u \leftarrow \eta$ ;  $v \leftarrow j$ ;
                 $\text{bandforlayer}[j] \leftarrow \text{bandforlayer}[j] - \sigma$ ;
            end if
        end
    end
    NodeConnect( $u, v$ );
     $t_{l_1}^x \leftarrow t_{l_1}^x \cup u$ ;
end
end
Refresh( $D_{n \times n}$ );
end

```

构造算法用于建立 k 个单层视频分发树集合 T_{l_i} , 即构造出 k 个单层重叠网。如果一个单层重叠网中节点的数为 n , 节点依据延迟选择邻居时, 算法考虑到了可能出现的对于某些上行带宽资源丰富的节点(通常是注册在高层的节点)的过度占用, 节点对某视频层 L_i 所分配的带宽按照 L_i 占整个视频层的比例分配。图 3 为建立同一视频层中的两棵视频分发树(实线和虚线分别表示两棵单独的视频分发树)时, 如果 N_D 和 N_E 之间的延迟最小, 则 N_E 还需要转发其它视频数据, 而 N_D 与其建立连接后需要获取同层数据流, 如果超过 $\text{bandforlayer}[E]$ 则 N_E 拒绝 N_D 的邻居建立请求。此时, N_D 需要向其它节点请求, 如和 N_C 建立邻居关系。

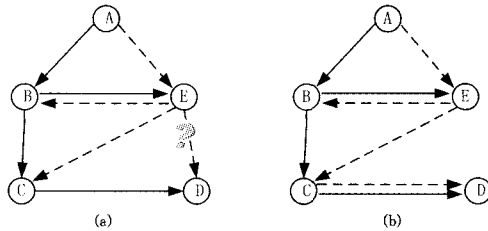


图 3 构造算法对节点的上行带宽的合理使用

在各层的视频分发树建立好以后, 第二个阶段即树的合并阶段, 把各层视频分发树进行叠加, 某节点对之间可能在多个视频层都存在连接, 则合并这样的连接。同时, 由于不同视频层分发树的邻居关系差异导致在合并阶段可能使节点对之间产生新的连接关系。合并算法如下。

Combination Algorithm

Input:

node set of different layer: $V_{l_1}, \dots, V_{l_k}, V_{l_k} \subset V_{l_{k-1}} \subset \dots \subset V_{l_1} = V$.

single overlay: $T_{l_1}, \dots, T_{l_k}$, each T_{l_i} is a singlelayer mesh.

streaming server: S , streaming source is divided into k single layers

Output:

T , final overlay network, which can deliver k layers of streaming data with low delivery time.

DealedNodeVector ← ϕ ;

DealedNodeVector ← DealedNodeVector + S ;

$k \leftarrow i$;

for $k \geq i \geq 1$ do

```

foreach node  $j \in V_{l_i} \& j \notin \text{Dealed NodeVector}$  do
    Combine Node Link( $j$ );
    Dealed Node Vector ← Dealed Node Vector +  $j$ ;
end
i ← i - 1;
end

```

合并算法主要把 $T_{l_1}, \dots, T_{l_k}$ 合并为 T , 即得到一个分发多层视频流的重叠网结构。图 4 为把多个单层重叠网合并后的节点邻居关系, 图 4(a) 为合并前 3 个单层重叠网结构, 图 4(b) 为合并后的重叠网结构。

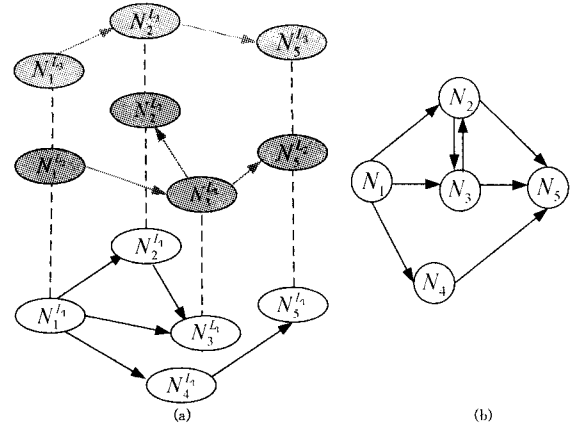


图 4 多个单层重叠网的合并

6 基于视频组的分布式重叠网构建策略

由于第 5 节所讨论的算法需要知道所有节点的带宽信息, 且需要专门的服务器用于计算, 导致这种中心式算法难以在实际系统中实现, 因此提出一种分布式的基于视频组的重叠网构建策略。

6.1 播放质量的计算

我们定义系统中节点在各层的播放质量, 也即给节点在某个时刻的播放质量计分(playback quality count)。在已有研究中, 通常直接通过 Track Server 提供注册在同层的节点作为新节点的候选节点, Track Server 的服务压力较大且注册在某一层的节点播放质量有可能并不理想, 且它不具备为新节点提供服务的能力, 因此提出通过节点播放质量的统计分值更合理地描述节点的播放质量和服务能力。为简单起见, 注册在 L_k 层的节点 $N_i^{l_k}$ 播放质量属于区间 $[k-1, k]$ 。换句话说, 一个注册在 L_k 层的节点, 其播放质量分值最差的情况统计分值为 $k-1$, 最佳情况统计分值为 k , 即完全服务。另外, $N_i^{l_k}$ 在一段时间内各数据层的服务率为 $SR_{ij}^{l_k}$, ($1 \leq j \leq$

k), 可以简单通过 $SR_{ij}^{l_k} = \frac{R_{l_j}}{r_{l_j} \cdot \Delta t}$ 计算, 其中, R_{l_j} 表示在时间内节点在 L_j 层。在一段时间内的播放质量统计得分计算方法为 $S_i = (k-1) + SR_{ij}^{l_k}$ 。由于分层流媒体的层次依赖性, 因此要求节点 $N_i^{l_k}$ 必须在满足 $SR_{i1}^{l_1} \geq \dots, \geq SR_{ik}^{l_k}$ 时计算 S_i 才是有意义的。

6.2 强服务节点的选择

根据文献[6]的研究结果, 在 P2P 流媒体的重叠网中实际存在部分稳定节点, 这些节点有足够的带宽且稳定性良好, 通过对实际系统如 PPLive 做测试, 表明通常存在约 5.5%~15.4% 这类稳定节点, 我们希望把这类节点有效地利用起来,

并使它们成为搜集视频组播放信息的管理节点。通过定义两个因素选择这类管理节点,即①节点在网时间因素;②节点注册的视频层。注册在较高视频层的节点需要相对较短的在网时间,因此它即可成为强服务节点。一些特殊的节点,如:内容服务商的专用服务节点、运营商部署的分布式服务节点等,都可以直接选择为管理节点。

6.3 视频组播放质量的统计方法

视频组并不是把节点按其注册的视频层进行分类,而是由注册在不同视频层的节点组成的集合进行分类,这样可以注册在不同层的节点之间有机会共享不同层的视频数据。每一个视频组均由一个管理节点和若干个普通节点组成,视频组所能容纳普通节点的数量是由管理节点的可用带宽决定。管理节点负责统计该视频组的播放质量,并在一定条件下把分值提供给 *TrackServer*,管理节点统计播放质量的方法是通过收集来自视频组中节点的播放质量计分得到的。为了降低开销,普通节点只在前后两次播放质量出现了较大变化(即当 $|S_i' - S_i| \geq \mu$, ($0 < \mu < 1$))或经过较长时间后才需要把播放质量消息发送给管理节点。

6.4 节点进入重叠网及邻居关系的建立

TrackServer 记录重叠网中的部分管理节点信息,其中包括对所管理的视频组在各视频层上的统计。当一个新节点 $N_i^{L_k}$ 进入重叠网时,首先联系 *TrackServer*,请求获取部分在视频层 L_k 的统计值较高的几个视频组。 $N_i^{L_k}$ 根据管理节点的 IP 地址相似性及视频组中已有节点的数量选择一个视频组进入。管理节点缓存了一段时间内本视频组中普通节点的视频得分, $N_i^{L_k}$ 会向管理节点请求获取部分视频组中候选邻居节点的信息,然后再通过邻居选择算法建立邻居关系,邻居选择算法为:

Neighbor Selection Algorithm—Request

Input:

candidate node list from M Node; V

Output:

Neighbor list of i; Neb_i

Neb_i ← ϕ ;

foreach node v ∈ V do

if $S_v > k$ then

hVector ← hVector + v;

end

if $k-1 \leq S_v \leq k$ then

sVector ← sVector + v;

end

if $S_v < k$ then

lVector ← lVector + v;

end

end

while sVector ≠ ϕ & i not fully served do

u ← MaxAdgIndex(sVector);

ConnectNode(i, u);

Neb_i ← Neb_i + u;

sVector ← sVector - u;

end

while hVector ≠ ϕ & i not fully served do

u ← MinAdgIndex(hVector);

ConnectNode(i, u);

Neb_i ← Neb_i + u;

hVector ← hVector - u;

end

while lVector ≠ ϕ & i not fully served do

u ← MaxAdgIndex(hVector);

ConnectNode(i, u);

Neb_i ← Neb_i + u;

lVector ← lVector - u;

end

这里引入邻居的服务优先级 *AdgIndex* 的概念, $AdgIndex = \frac{S_i}{D(i, j)}$, 候选节点的视频统计分值 S_i 越大, 表示其具有

越好的数据播放质量, 其具备服务 i 的能力也越强。而 $D(i, j)$ 表示候选节点 i 到 j 的端到端延迟。*AdgIndex* 越大则该候选节点被选择为邻居节点的优先级也越大。节点 $N_i^{L_k}$ 首先从播放质量统计值在 $[k-1, k]$ 的候选节点中选择邻居节点, 依据 *AdgIndex* 从大到小的策略选择建立邻居关系, 直到 $N_i^{L_k}$ 达到完全服务状态。如果 *sVector* 中的节点选择完后仍然没有达到满意的服务, 则向 *hVector* 中的候选节点请求, *hVector* 中的节点都是统计分值高于 k 的, 即注册的视频层高于 L_k 的节点组成的集合。根据分层视频流的解码依赖性, 分值高于 k 的节点在 L_k 层具有较高的统计分值。因此算法从 *hVector* 中得分最低的节点开始选择, 直到 $N_i^{L_k}$ 达到完全服务。如果 $N_i^{L_k}$ 仍然无法达到完全服务, 则只能从 *lVector* 中选择节点建立邻居关系。

当候选节点 j 接收到来自 $N_i^{L_k}$ 的邻居建立请求时, 会根据自己现有邻居的数量及播放质量来统计分值, 决定是否同意成为 $N_i^{L_k}$ 的邻居。如果 j 当前的邻居数小于系统允许的最大邻居数 *MaxNum* 时直接同意, 而相反, j 从自己的邻居中选择出统计分值最低的节点和 $N_i^{L_k}$ 进行比较决定是否同意 $N_i^{L_k}$ 的请求。具体邻居选择响应算法如下:

NeighborSelectionAlgorithm—Respond

Input:

the current neighbor list; NBlist

the maximal neighbor number; MaxNum

the current number of neighbors; N

Output:

Accept or refuse the neighbor connection

request from i

RcvQuest(i);

k ← i. layer;

if $N < \text{MaxNum}$ then

SendAccept(i);

AddNeighbor(i, NBlist);

else

v ← SelectLowestScore(NBlist);

if $S_v > k$ then

SendReject(i);

if $S_v \leq k-1$ then

EraseNeighbor(v);

SendAccept(i);

AddNeighbor(i, NBlist);

if $k-1 < S_v \leq k$ then

RcvNodewithProbability;

end

7 实验与结论

本节给出重叠网络构建策略的实验及对实验结论的分析

对比。基于 Oversim 的高仿真事件仿真平台, 实现了分层视频流媒体环境下的重叠网构建策略。为了保证实验的客观性, 同时采用了拓扑产生器 GT-ITM^[9] 产生底层物理拓扑结构, 实验中 intra-transit 区域和 intra-stub 区域的带宽均设置为 20Mb, 而 transit-stub 区域的带宽设置为 10Mb。整个物理拓扑结构有 1 个 transit-domain, 该区域含有 28 个骨干路由器, 每个骨干路由器能够连接 3 个 stub-domain, 而每个 stub-domain 含有 8 个接入路由器, 这样整个底层物理网络结构由 672 个接入路由器构成, 可以模拟一个较大规模的互联网络环境。进入重叠网的节点个数规模最大为 10000 个。每个节点最多可以有 15 个邻居节点。采用 Oversim 系统提供的 LifetimeChurn^[8] 模拟节点的随机进入和退出, 同时设置了节点的两类平均在线时间(Online Time), 即对于非稳定节点其平均在线时间为 1,000s, 而对于稳定节点的平均在线时间为 10,000s, 通过对 PPLive 的实际数据收集^[10], 表明这样的时间设置是合理的。对于视频流的产生, 采用 JSVM 产生了长度为 2 个小时时长的视频流, 同时视频流分为 4 个层, 其视频流速率分别为 620Kbps, 825Kbps, 945Kbps 和 1065Kbps。本文只探讨重叠网络构建, 而对于调度则采用文献[11]提出的基于优先调度的算法。

实验主要比较的性能参数为: ①重叠网络构造策略所导致的视频服务器带宽的消耗, 重点把本文提出的策略和 OCals 系统^[1]所使用的重叠网构造策略进行了比较; ②节点平均可播放数据, 我们定义一个节点平均可播放数据为单位时间内节点可以播放其注册的视频层对应的视频图像的比例。由于分层视频编码的特性是各层数据之间具有相互依赖性, 因此对底层视频层有较高的重要性, 高层视频的解码依赖于底层视频数据的获取, 虽然高层数据获取到了, 但如果对应的底层数据却没有获得, 则这些高层数据仍然是无效数据。

从图 5 的对比可以看到, 随着实验时间的推移(从 0 到 90min), 节点随机地进入和退出重叠网并建立邻居关系, 构建重叠网络的结构。OCals 系统的重叠网构造策略中服务器的带宽消耗增长较快, 由于本文在第 5 节中提出的集中启发式算法是在获取所有节点信息的情况下建立的邻居关系, 可以严格地按照节点注册的视频层选择其在重叠网上的位置, 也即注册在高层的节点距离视频服务器较近的原则建立整个重叠网, 因此可以充分利用各个节点的上行带宽, 这样也就对视频服务器的带宽消耗最少, 其也是最理想的重叠网构建方法。从图 5 还可以看到, 本文提出的基于视频组的分布式重叠网构建策略的分层流媒体系统对于服务器的带宽消耗也较低, 呈现出逼近最理想的集中启发式算法的性能, 明显低于经典的其它策略, 如 OCals。

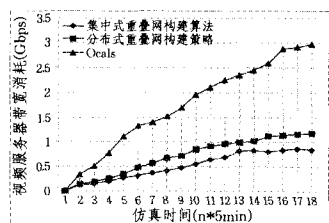


图 5 服务器带宽消耗对比

图 6 给出了随仿真时间的推移, 节点不断加入到网络中

的情况下节点平均可播放的数据比率。基于提出的分布式重叠网构建策略(DCO)的系统比基于 OCals 的系统明显具有较高的数据播放比率, 其主要原因是 DCO 可以较好地保证节点和其邻居节点具有可用的数据, 较好地避免了由于节点在重叠网中的位置导致部分视频层数据“阻塞”出现。

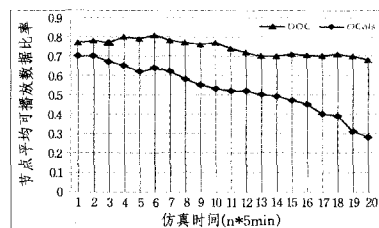


图 6 节点平均可播放比率

结束语 通过建立数学模型描述了基于分层视频流环境下的重叠网构建策略, 并通过集中式算法和分布式启发式算法给出了理想情况和实际可实现的逼近理想情况的策略。通过时间比较得出分布式启发策略的有效性。进一步的工作可以研究如何在实际网络环境如 PlanetLab 下部署这样的重叠网构建策略。

参考文献

- [1] Xiao X, Shi Y, Zhang B, et al. OCals: A Novel Overlay Construction Approach for Layered Streaming Layered Streaming [C]//Proceeding of IEEE ICC. May 2008;235-241
- [2] Nguyen A T, Li Bao-chun, Eliassen F. Quality- and Context-aware Neighbor Selection for Layered Peer-to-Peer Streaming [C]//Proceeding of IEEE ICC. 2010;432-437
- [3] Magharei N, Rejaie R, Guo Y. Mesh or Multiple-tree, a comparative study of live P2P streaming approaches [C]//Proceeding of IEEE INFOCOM. Anchorage, AK, May 2007;1424-1432
- [4] Rejaie R, Ortega A. PALS: Peer-to-Peer Adaptive Layered Streaming [C]//Proceeding of ACM NOSSDAV. 2003;654-659
- [5] Cormen T H, et al. Introduction to Algorithms [M]. Cambridge, US, MIT Press, MA, 1990
- [6] Wang F, Liu J, Xiong Y. Stable Peers: Existence, Importance, and Application in Peer-to-Peer Live Video Streaming [C]//Proceeding of IEEE INFOCOM. Phoenix, AZ, USA, April 2008; 985-994
- [7] Baumgart I, Heep B, Krause S. OverSim: scalable and flexible overlay framework for simulation and real network applications [C]//Proceeding of IEEE P2P'09. Berlin, Germany, 2009; 211-220
- [8] Baumgart I, Heep B, Krause S. OverSim: A Flexible Overlay Network Simulation [C]//Proceedings of 10th IEEE Global Internet in conjunction with IEEE Infocom. 2007;168-177
- [9] Calvert K, Doar M, Zegura E W. Modeling Internet Topology [J]. IEEE Communications Magazine, 1997, 5;34-46
- [10] Hei Xiao-jun, Liang Chao, Liang Jian. A Measurement Study of a Large-Scale P2P IPTV System [J]. IEEE Transactions on Multimedia, 2007, 9(8); 1672-1687
- [11] Liu Z, Shen Y, Ross K W, et al. LayerP2P: Using Layered Video Chunks in P2P Live Streaming [J]. IEEE Transactions on Multimedia, 2009, 6;221-235