

P2P 流媒体中的多发送端选择算法

李泽平^{1,2}, 卢显良¹, 李 梁¹

(1. 电子科技大学计算机科学与工程学院, 四川成都 610054; 2. 贵州大学理学院, 贵州贵阳 550025)

摘要: 在多对单 P2P 流媒体模式下, 如何选择多个发送端, 并为其最优地分配发送速率和数据是一个难题. 为此, 提出了一种新的多发送端选择和最优速率及数据分配算法(MSRDA). 首先把待解决的问题模型化为线性最优化问题, 然后给出了求解最优化问题的算法. 不同于已有的算法只按链路的丢包率来选择发送端, MSRDA 根据到各发送端链路的丢包率与链路的可用带宽比升序排序来选择多发送端, 并最优地分配发送速率和数据. 理论分析和仿真实验结果表明: 在不同的网络条件下 MSRDA 能有效地减少聚合带宽的丢包率, 并具有自适应性, 优于现有同类算法.

关键词: 对等网络 (P2P); 流媒体; 速率分配; 数据分配

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 09-2199-06

Algorithm for Multiple Sender Selection in P2P Streaming

LI Ze-ping^{1,2}, LU Xian-liang¹, LI Liang¹

(1. School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China;

2. The College of Science, Guizhou University, Guiyang, Guizhou 550025, China)

Abstract: In P2P media streaming with the pattern of multiple senders and single receiver, it is still a challenge to select multiple senders and optimally allocate streaming rate and media data among multiple senders. To cope with the problem, a new solution(MSRDA) is proposed. Firstly, the problem is modelled as a linear optimization problem, and then the MSRDA algorithm is presented. The MSRDA selects senders according to the sort ascending of selected path's ratios of packet loss probability to available bandwidth, which is distinct from existing algorithms which select paths only by the packet loss probability of selected paths. Theory analysis and simulation results using NS2 show that the proposed algorithm can effectively reduce the packet loss probability of aggregated bandwidth, dynamically adapt to network fluctuation and achieve a better performance than the related works with different parameters.

Key words: peer-to-peer(P2P); media streaming; rate allocation; data allocation

1 引言

视频点播已成为对等网络中一项重要的应用^[1,2], 视频分发是视频点播系统的核心研究内容. 目前针对 P2P 视频分发技术的划分方法有多种, 按照数据的发送、接收和传输方式, 可将分发策略分为单对多模式、多对单模式和多对多模式三类^[3]. 单对多模式, 指一个节点的数据是来自单一的节点, 而将其发送到其它多个节点; 多对单模式, 指一个请求节点需要多个发送节点给它提供数据; 多对多模式, 指任何一个节点既可以接受多个节点的数据, 也可以向多个节点发送数据.

多对单传输模式适合在 P2P 环境下进行视频分发, 是近几年的研究热点, 而最优地选择多发送端是其研究重点之一, 主要相关工作见文献[4~7]. Hefeeda 等^[4]提出一种多对单的 P2P 流媒体系统 PROMISE, 该系统在选择最优发送端时采用拓扑感知技术来收集到各发送端

的链路状态信息, 其缺点是决策时需要获取太多的网络参数, 通信开销大, 计算复杂度高. Nguyen^[5]提出了一种接收端驱动的、多对单的流化协议, 该协议存在两个问题: ①协议使用的速率分配算法计算复杂度非常大; ②协议要求每个发送端都运行数据包分配算法, 增加了发送端的计算开销. 在文献[6]中 Nguyen 以丢包率最小为目标, 提出了对多发送端进行速率和数据分配的算法. Jurca^[7]提出了一种端到端的多路径选择及速率分配算法.

本文研究发现若采用文献[6,7]的方法选择聚合链路所得到的带宽质量很不稳定, 随机性较大, 有时会选出质量很差的聚合链路, 导致视频请求方的收视质量经常波动. 产生这种情况的原因在于文献[6,7]只根据链路的丢包率参数从低到高来选择聚合链路, 而没有同时考虑链路的可用带宽参数, 只用一个参数不能全面地反映通信链路的质量.

针对现有方法^[4~7]存在的问题, 本文提出一种新的

多发送端选择和最优速率及数据分配算法(MSRDA). 首先,把选择多发送端和为其分配发送速率的问题化为最优化模型,并给出了求解该问题的 MSRA 算法. 然后基于 MSRA 建立了多发送端数据分配模型及数据分配算法(MSDA). 最后,基于 MSRA 和 MSDA 提出了动态速率和数据分配算法(MSRDA). 不同于已有的算法只按链路的丢包率从低到高依次来选择发送端,MSRDA 根据预估的到各发送端链路的丢包率与链路的可用带宽比升序排序来选择发送端. 理论分析和仿真实验表明:本文提出的算法能得到更优质的聚合带宽,算法的计算复杂度低、实用性强,而且对环境没有特别的要求,优于现有同类算法^[4-7].

2 计算模型

设视频文件在分发前已被统一分段、分包(chunk)和编码,并且预存在各发送端. 当有节点请求视频时,每个视频子段的包由多个发送端共同提供,每个发送端应发送数据包的序号及个数由视频请求节点确定. 发送端集合 S 由 P2P 网络中能提供视频服务的计算机 $Sender_1, Sender_2, \dots, Sender_n$ 组成; Receiver 表示视频请求节点; $L_i(t) = (b_i(t), p_i(t))$ 表示在时间段 $(t_0, t_0 + T]$ 由 Receiver 估计的到 $Sender_i$ 链路的当前状态, $b_i(t)$ 和 $p_i(t)$ 分别表示链路 $L_i(t)$ 的最大可用带宽和丢包率;用 $B(t) = (b_1(t), b_2(t), \dots, b_n(t))$, $P(t) = (p_1(t), p_2(t), \dots, p_n(t))$ 分别表示 S 的可用带宽向量和丢包率向量. 活跃集是 Receiver 根据它到 S 中各发送端链路的当前状态选出的最优发送端子集,用 U 表示;为方便表述,文中用 S_U 表示活跃集中发送端的下标构成的集合;表 1 是后文使用的主要符号.

表 1 主要符号表

符号	含 义
S	发送端集合
U	活跃集
Receiver	表示视频请求节点
$b_i(t)$	Receiver 到发送端 $Sender_i$ 链路的最大可用带宽
$p_i(t)$	Receiver 到发送端 $Sender_i$ 链路的丢包率
$L_i(t)$	Receiver 到发送端 $Sender_i$ 链路的当前状态
S_U	由活跃集中发送端的下标构成的集合
R_U	分配给活跃集中的发送端的发送速率构成的集合
λ	Receiver 的最大下行带宽
T	最小请求间隔
$B(t)$	发送端集 S 的可用带宽向量
$P(t)$	发送端集 S 链路的丢包率向量
D_U	发送端集 U 的发包序号区间向量
d	存放各发送端的数据分配
$d_i(t)$	分配给发送端 $Sender_i$ 的应发数据包的个数
$D_i(t)$	分配给发送端 $Sender_i$ 的应发数据包的序号区间
H, L	H 一个视频子段所包含视频包的个数, L 视频包的长度

Receiver 获取视频的方法如下:首先从导航服务器(Tracker Server)获取能提供视频文件的 S , 然后从 S 中的发送端获取视频. Receiver 与 S 交互的主要过程如下:①Receiver 向每个 $Sender_i$ ($Sender_i \in S$) 请求它想获取的视频;②若 $Sender_i$ 能为 Receiver 提供服务,则它立即发回确认 ACK;③Receiver 估计它到每个可用 $Sender_i$ 链路的当前状态 $L_i(t)$, 然后根据收集到的链路信息 $L_i(t)$, 从 S 中选出 U , 并从 U 请求视频数据;④ $Sender_i$ 向 Receiver 发送被请求的视频.

本文约定收集 $Sender_i$ 链路的当前状态 $L_i(t)$ 由 Receiver 节点承担,而估算 Receiver 到 $Sender_i$ 链路最大可用带宽 $b_i(t)$ 的方法是直接应用公式(1)^[8]:

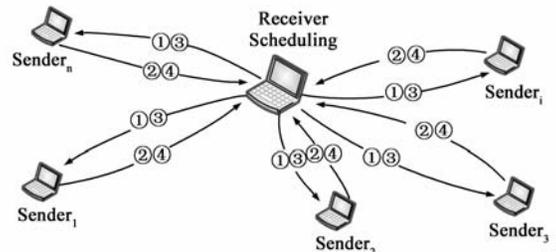


图 1 Receiver与Sender交互模型

$$b_i(t) = \frac{z}{t_R \sqrt{\frac{2ge}{3}} + t_C \min(1, 3\sqrt{\frac{3ge}{8}}) e(1 + 32e^2)} \quad (1)$$

其中 z 为数据包的大小, t_R 为包的往返时间, t_C 为 TCP 重传超时时间, e 为丢包事件率, g 是一个 ACK 应答代表接收到的报文数, $b_i(t)$ 为 Receiver 估算出的可用发送速率上限.

2.1 多发送端选择与最优速率分配模型

设 Receiver 到 S 的各发送端链路是相互独立的,且在时间段 $(t_0, t_0 + T]$ Receiver 到 $Sender_i$ 链路的当前状态为 $L_i(t) = (b_i(t), p_i(t))$, 记 $Sender_i$ 给 Receiver 提供的平均发送速率为 $r_i(t)$ ($r_i(t) \leq b_i(t)$), 把 S 中各发送端的流速率之和 $\sum_{Sender_i \in S} r_i(t)$ 定义为 S 提供给 Receiver 的聚合流带宽;定义式(2)为 S 所提供聚合流带宽的平均丢包率:

$$\frac{\sum_{Sender_i \in S} r_i(t) * p_i(t)}{\sum_{Sender_i \in S} r_i(t)} \quad (2)$$

其中, $t \in (t_0, t_0 + T]$. 设 Receiver 的下行带宽为 λ , 多发送端选择与最优速率分配的目标是:在时间段 $(t_0, t_0 + T]$ 内、根据 Receiver 到 S 中各发送端链路的当前状态 $L_i(t)$ ($i = 1, 2, \dots, |S|$) 选出 U , 并为 U 中的每个成员 $Sender_i$ 最优地指定一个发送速率 $r_i(t)$, 使 U 能为 Receiver 提供最优质量的聚合流带宽 $\sum_{i \in S_U} r_i(t) = \lambda$. 选择 U 以及为 U 中的成员 $Sender_i$ 最优地指定发送速率

$r_i(t)$ 的问题, 可以通过求解下列线性规划 A 问题得到最优解 S_U 和 R_U :

$$\text{规划 A } \quad \text{Min} \quad \sum_{i \in S_U} \frac{r_i(t) * p_i(t)}{b_i(t)}$$

$$\text{s.t.} \quad r_i(t) \leq b_i(t), \quad i \in S_U \quad (3)$$

$$\sum_{i \in S_U} r_i(t) = \lambda \quad (4)$$

$$r_i(t) > 0, \quad i \in S_U \quad (5)$$

其中, $t \in (t_0, t_0 + T]$, $r_i(t)$ 是 Receiver 分配给 U 中发送端 Sender_i 的发送速率, 后文用 R_U 表示分配给 U 中的发送端的发送速率构成的集合, 即 $R_U = \{r_i(t) \mid i \in S_U\}$. 约束条件(3)限制发送端 Sender_i 的发送速率不能超过被估计的最大可用带宽 $b_i(t)$; 约束条件(4)规定 U 中各发送端的发送速率之和等于 Receiver 的下行带宽; 约束条件(5)规定 U 中的每个成员都参与视频分发.

2.2 多发送端数据分配模型

设视频文件在分发前已被划分成长度相等的子段, 每个子段由 H 个等长的包组成, 包的长度为 L . Receiver 要求当前 U 中的发送端每个时间段 $(t_0, t_0 + T]$ 向它发送一个视频子段. 如何确定 T 值非常关键, 若 T 值太小, 各发送端所发的数据包由于受带宽的限制不能在 T 时间内到达请求方. 但 T 值又不能太大, 因为 Receiver 对视频传输有时限要求, 在播放前它必须到达请求方. 下面说明如何根据多发送端选择与最优速率分配模型所得到的最优解 (S_U 和 R_U) 为多发送端进行最优数据分配建模. 为方便描述, 设选出的 U 有 m 个发送端, 且 S_U, R_U 的成员分别为: $S_U = \{1, 2, \dots, m\}$, $R_U = \{r_1(t), r_2(t), \dots, r_m(t)\}$. 为发送端 Sender_i 进行数据分配, 即确定它在一个时间段 $(t_0, t_0 + T]$ 内应发送数据包的个数 $d_i(t)$. 由于 Receiver 要求每个视频子段在时间段 $(t_0, t_0 + T]$ 内到达请求方, 对 U 中的发送端进行数据包分配的问题, 可以通过求解下列线性规划问题 (B) 找到满足约束条件(6) ~ (10) 的最小请求间隔 T 和最优数据分配 $d = (d_1(t), d_2(t), \dots, d_m(t))$:

$$\text{规划 B } \quad \text{Min} \quad T$$

$$\text{s.t.} \quad d_i(t)L \leq r_i(t)T, \quad i = 1, 2, \dots, m \quad (6)$$

$$r(t) = \sum_{i=1}^m r_i(t) \quad (7)$$

$$H = \sum_{i=1}^m d_i(t) \quad (8)$$

$$HL \leq r(t)T \quad (9)$$

$$d_i(t) \geq 0, \quad i = 1, 2, \dots, m \quad (10)$$

其中 $t \in (t_0, t_0 + T]$, $d_i(t)$ 是分配给发送端 $\text{sender}_i (i = 1, 2, \dots, m)$ 在 T 时间内应发送数据包的个数, $d_i(t)$ 必需满足约束条件(6); 约束条件(8)规定各发送端所发送数据包之和应等于一个视频子段数据包的总数; 约束条件(9)限定一个视频子段的所有包必须在时间段 $(t_0,$

$t_0 + T]$ 内到达请求节点; 约束条件(10)规定数据包的个数是非负的. 当得到各发送端的数据分配 $d = (d_1(t), d_2(t), \dots, d_m(t))$ 后, 根据 d 可以进一步确定各发送端的发包序号区间分配 $D_U \leftarrow (D_1(t), D_2(t), \dots, D_m(t))$, 计算 $D_i(t)$ 的公式为 $D_1(t) \leftarrow [1, d_1(t)]$, $D_i(t) \leftarrow [\sum_{j=1}^{i-1} d_j(t) + 1, \sum_{j=1}^{i-1} d_j(t) + d_i(t)]$, ($i = 2, \dots, m$).

3 多发送端选择与最优速率及数据分配算法

本节在第 2 节计算模型的基础上给出相应的算法.

3.1 多发送端选择与最优速率分配算法

利用算法 1 可得到线性规划 A 问题的最优解.

算法 1 MSRA($B(t), P(t), \lambda, S_U, R_U, m$)

Input: $B(t), P(t), \lambda$; Output: S_U, R_U, m

1 Sort the $\frac{p_i(t)}{b_i(t)} (i = 1, 2, \dots, n)$ in increasing order:

$$\frac{p_{i_1}(t)}{b_{i_1}(t)} \leq \frac{p_{i_2}(t)}{b_{i_2}(t)} \leq \dots \leq \frac{p_{i_n}(t)}{b_{i_n}(t)}$$

2 $S_U \leftarrow \phi, R_U \leftarrow \phi, y \leftarrow 0, j \leftarrow 0$

3 repeat

4 $j \leftarrow j + 1$

5 if $y + b_{i_j}(t) \leq \lambda$ then $r_{i_j}(t) \leftarrow b_{i_j}(t)$

else $r_{i_j}(t) \leftarrow \lambda - y$

6 $S_U \leftarrow S_U \cup \{i_j\}, R_U \leftarrow R_U \cup \{r_{i_j}(t)\}, y \leftarrow y + b_{i_j}(t)$

7 until $y \geq \lambda$

8 $m \leftarrow j$

9 return S_U, R_U, m

算法解释 算法先计算每个链路的平均丢包率, 然后对各链路的平均丢包率按升序排序; 行 2 ~ 8 找最小的正整数 $m (1 \leq m \leq n)$, 使 $\sum_{j=1}^m b_{i_j}(t) \geq \lambda$. 当 $\sum_{j=1}^m b_{i_j}(t) = \lambda$ 时, 分配给 m 个发送端: $\text{sender}_{i_1}, \text{sender}_{i_2}, \dots, \text{sender}_{i_m}$ 的发送速率依次为 $b_{i_1}(t), b_{i_2}(t), \dots, b_{i_m}(t)$; 当 $\sum_{j=1}^m b_{i_j}(t) > \lambda$ 时, 分配给前 $m - 1$ 个发送端的发送速率依次为: $b_{i_1}(t), b_{i_2}(t), \dots, b_{i_{m-1}}(t)$, 分配给第 m 个发送端的发送速率为: $\lambda - \sum_{j=1}^{m-1} b_{i_j}(t)$; 行 9 返回 S_U, R_U, m .

定理 1 算法 1 能得到线性规划 A 问题的最优解.

证明 从算法 1 选择发送端链路的过程行 2 ~ 7 可以看出, 算法是按链路的平均丢包率从低到高贪婪地选择低丢包率链路: $b_{i_1}(t), b_{i_2}(t), \dots$, 直到聚合带宽等于 λ . 用这种方法选择发送端和分配速率所得到的 m 个发送端的发送速率之和正好等于视频请求节点的最

大下载速率 λ , 而聚合带宽 λ 的平均丢包率被降到最低. \square

3.2 多发送端数据分配算法

线性规划 B 问题的最优解由算法 2 给出.

算法 2 MSDA(R_U, H, L, T, d, D_U, m)

Input: R_U, H, L, m ; Output: d, D_U, T

```

1   $r(t) \leftarrow \sum_{i=1}^m r_i(t), T \leftarrow \lceil (HL)/r(t) \rceil$ 
2   $f \leftarrow H, i \leftarrow 0$ 
3  repeat
4     $i \leftarrow i + 1$ 
5    if  $f > 0$  then
6       $d_i(t) \leftarrow \lceil (r_i(t)H)/r(t) \rceil$ 
7      if  $(d_i(t)L)/T > r_i(t)$  then  $d_i(t) \leftarrow d_i(t) - 1$ 
8       $W \leftarrow f - d_i(t)$ 
9      if  $W > 0$  then  $f \leftarrow f - d_i(t)$  else  $d_i(t) \leftarrow f$ 
10   until  $W \leq 0$ 
11   if  $i + 1 \leq m$  then for  $j \leftarrow i + 1$  to  $m$  do  $d_j(t) \leftarrow 0$ 
12    $D_1(t) \leftarrow [1, d_1(t)]$ ,  $g \leftarrow d_1(t)$ 
13   for  $k \leftarrow 2$  to  $i$  do
14      $D_k(t) \leftarrow [g + 1, g + d_k(t)]$ 
15      $g \leftarrow g + d_k$ 
16   end for
17   for  $k \leftarrow i + 1$  to  $m$  do  $D_k(t) \leftarrow [0, 0]$ 
18    $d \leftarrow (d_1, d_2, \dots, d_m)$ ,  $D_U \leftarrow (D_1(t), D_2(t), \dots, D_m(t))$ 
19   return  $d, D_U, T$ 

```

算法解释 行 1 计算 $\sum_{i=1}^m r_i(t)$ 和 T ; 行 2~10 计算把 H 个包分配给前 i 个发送端 $Sender_k (k = 1, 2, \dots, i)$. 每个发送端被指派发送的数据包个数为 $d_k(t)$, 它是在满足约束条件 $d_k(t)L \leq r_k(t)T$ 下, 找到的最大的正整数 $d_k(t)$, 其目的是充分利用发送端 $Sender_k$ 的带宽, 又保证能在时间 T 内把数据包发送到请求方; 行 11 指定 $m - i$ 个发送端的发包数为零 ($d_k(t) = 0, k = i + 1, \dots, m$); 行 12~17 根据 d 计算发送端的发包序号区间分配; 行 19 返回计算结果.

定理 2 算法 MSDA 能得到线性规划 B 问题的最优整数解.

证明略.

3.3 多发送端选择与最优速率及数据分配算法

多发送端选择与最优速率及数据分配算法 (MSR-DA) 主要由 MSRA 和 MSDA 组成, 它们分别完成选择 U 和进行最优速率及数据分配. 由于视频分发一般会持续较长时间, 在视频传输期间, 从发送端到接收方的链路可能会出现拥塞或故障等情况, 这些变化会导致 U 中发送端链路的状态变坏, 提供的视频质量下降. MSR-DA 算法具有自适应性, 其选出的 U 能随网络条件的变

化而不断地更新其成员, 使 U 能持续地为请求节点提供最优质量的视频.

算法 3 MSRDA(S, λ, L, H)

Input: S, λ, L, H

```

1  ( $B(t), P(t)$ )  $\leftarrow$  ML( $S, B(t), P(t)$ )
2  while session is not over do
3    ( $R_U, S_U, m$ )  $\leftarrow$  MSRA( $B(t), P(t), \lambda, S_U, R_U, m$ )
4    ( $d, D_U, T$ )  $\leftarrow$  MSDA( $R_U, H, L, T, d, D_U, m$ )
5     $U \leftarrow$  Select( $S_U, S$ )
6    Connect( $U$ )
7    update  $\leftarrow$  false
8    while update = false do
9      SendAssignment(SegmentNo,  $U, R_U, D_U$ )
10     ReceiveChunk(SegmentNo,  $U, D_U$ )
11     ( $B_U(t), P_U(t)$ )  $\leftarrow$  ML( $U, B_U(t), P_U(t)$ )
12     if  $\frac{H - \sum_{Sender_i \in U} \text{Receive}(Sender_i, D_i)}{H} \geq \alpha_1$  then
13       update  $\leftarrow$  true
14     end while
15     for each  $Sender_i \in U$  do
16       if  $\frac{|d_i(t) - \text{Receive}(Sender_i, D_i)|}{d_i(t)} \geq \alpha_2$  then
17          $S \leftarrow S - \{Sender_i\}$ ;
18       end for
19      $R \leftarrow S - U$ 
20     ( $B_R(t), P_R(t)$ )  $\leftarrow$  ML( $R, B_R(t), P_R(t)$ )
21      $B(t) \leftarrow B_U(t) \oplus B_R(t), P(t) \leftarrow P_U(t) \oplus P_R(t)$ 
22   end while

```

算法解释 函数 ML($Z, X(t), Y(t)$) 的功能是测量发送端集 Z 的最大可用带宽向量 $X(t)$ 和链路的丢包率向量 $Y(t)$. 算法行 1 Receiver 估算它到 S 的可用带宽 $B(t)$ 和链路的丢包率 $P(t)$; 行 3 调用算法 1 选出活跃集 S_U , 计算最优速率分配 R_U ; 行 4 调用算法 2 计算数据分配 d, D_U 以及确定最小请求间隔 T ; 行 5 根据 S 及 S_U 选出活跃集 U ; 行 6 Receiver 建立到 U 中发送端的并行连接; 行 9 根据 U, R_U 和 D_U , Receiver 向 U 请求段号为 SegmentNo 的视频子段, 并向 U 中的发送端 $Sender_i$ 提出数据 $D_i(t)$ 及发送速率 $r_i(t)$ 请求; 行 10 Receiver 接收来自活跃集 U 、段号为 SegmentNo 的视频子段的数据包; 行 11 Receiver 估算到 U 中各发送端的最大可用带宽 $B_U(t)$ 和丢包率 $P_U(t)$; 行 12 Receiver 判断在一次视频子段的请求中整个 U 的发包丢失率是否大于或等于阈值 α_1 ; 行 8~14 每次请求一个视频子段, 若在一次请求中整个 U 的发包丢失率超过阈值 α_1 , 则立即退出循环、更新 U , 否则 U 保持不变、继续从同一 U

取下一视频子段;行 15~18 对 U 中的每个发送端 $Sender_i$ 检查其发包丢失率,若 $Sender_i$ 的发包丢失率超过规定的阈值 α_2 ,则 $Sender_i$ 被从发送端集合 S 中删除;行 19 R 是 S 中非活跃集的成员构成的集合;行 21 更新 $B(t)$ 和 $P(t)$;循环行 2~22 可能被重复执行多次,每次循环选择一次 U ,直到视频请求结束。

3.4 算法分析与比较

设 $n = |S|$, $m = |U|$ 。

算法 1 计算 $p_i(t)/b_i(t)$ ($i = 1, 2, \dots, n$) 需要 n 次运算,对 $p_i(t)/b_i(t)$ ($i = 1, 2, \dots, n$) 从小到大排序的平均时间复杂度为 $O(n \log_2 n)$,选择 S_U 的运算量至多为 $O(n)$,因此算法 1 的计算时间复杂度为 $O(n \log_2 n)$ 。

算法 2 计算数据包分配 ($d_1(t), d_2(t), \dots, d_m(t)$) 的复杂度为 $O(m)$,计算发包序号区间分配 ($D_1(t), D_2(t), \dots, D_m(t)$) 的复杂度也是 $O(m)$,因此算法 2 的计算复杂度为 $O(m)$ 。

算法 3 算法 1、算法 2 的计算复杂度分别为 $O(n \log_2 n)$ 、 $O(m)$,而 $m \leq n$,所以 MSRDA 执行一次外循环的时间复杂度为 $O(n \log_2 n)$ 。一般情况下,请求 V 个视频子段的循环次数小于 V ,因为 MSRDA 每次只请求一个视频子段,若上一次 U 发送的视频子段丢包率在允许的范围内、不需要更新 U ,继续从同一 U 请求视频。在最坏的情况下每请求一个视频子段,需要更新一次 U (更新 U 需执行一次外循环),因此 MSRDA 请求 V 个视频子段的时间复杂度至多为 $O(Vn \log_2 n)$ 。

MSRDA 与同类算法^[4~7]比较:文献[4]没有给出算法的复杂度分析。在选择 U 时 PROMISE^[4]需要收集太多的网络参数,通信开销大、算法的复杂度高;而 MSRDA 不需要网络拓扑信息减少了通信开销。

文献[5]使用算法 RAA 计算对多个发送端进行速率分配。RAA 计算 H 个包在 m 个发送端进行速率分配的复杂度为 $O(H^m)$;而 MSRDA 算法在进行数据包分配时使用 MSDA,其计算复杂度为 $O(m)$,与待分配包的个数 H 无关。文献[5]在进行数据包分配时还要求每个发送端都运行数据包分配算法(PPA),增加了发送端的计算开销。而 MSRDA 算法没有此要求。文献[6]没有给出算法的时间复杂度分析,它在进行数据包分配时也要求每个发送端都运行数据包分配算法(PPA)。

文献[7]提出一种端到端的多路径选择和速率分配算法(OSRA),OSRA 的计算复杂度为 $O(n \log_2 n)$ (n 为链路数),与本文相同。

从上述分析可以看出,MSRDA 在实现方面不需要“底层的拓扑信息”,在进行数据包分配时也不要求发送端运行“包分配算法”,因此它的实现开销低于文献[4~6]中的算法;在计算时间复杂度方面,MSRDA 低于

RAA^[5],与 OSRA^[7]相同。但 MSRDA 最突出的优点是能为视频请求方提供更高质量的聚合带宽。

4 仿真实验及结果分析

根据前文的分析在同类算法^[4~7]中,文献[4,5]提出的算法存在计算复杂度高,实现开销大等问题,因此在实验中本文算法只与文献[6,7]中的算法进行比较。实验的目的:比较 MSRDA 与同类算法^[6,7]所提供聚合带宽的质量以验证提出算法的有效性。

为方便表述,称 MSRDA 为 A 算法,而把 MSRDA 中的子算法 MSRA 用文献[6,7]中选择(发送端)链路和进行速率分配的算法替换后所得到的算法称为 B 算法。A 算法与 B 算法的本质区别在于二者选择发送端链路的策略不同:A 算法根据到各发送端链路的丢包率与链路的可用带宽比升序排序来选择多发送端,并最优地分配发送速率;而 B 算法只按链路的丢包率从低到高依次来选择发送端和为其分配发送速率。为着重比较两种选择链路策略存在的差异,在实验中算法 A 和算法 B 都使用同一数据分配算法(MSDA)。实验参考了文献[5,6]的仿真方法以及使用 NS-2^[9]作仿真工具。仿真实验的网络拓扑如图 2 所示。在实验中,发送端集由 50 个节点组成 ($n = |S| = 50$),每个节点的发送速率随机地从 [5kbps, 100kbps] 区间抽取,它们通过路由器 R_A 和 R_B 与 3 个接收节点连接。从路由器 R_B 到 Receiver 的最大下行带宽为 1Mbps,其中 R_A 与 R_B 之间的链路带宽为 3Mbps。路由器 R_A 与 R_B 、 $R_1 \sim R_{50}$ 之间的链路延时为 10ms,路由器 R_B 、 $R_1 \sim R_{50}$ 与端系统之间延时均为 20ms。在每次实验中,一个 296.1Mbytes 的视频段被分成 10 个子段,每个子段被分成 2166 个包,Receiver 每隔 T 秒请求一个视频子段,连续请求 10 个视频子段,持续时间 41.4 分钟。在实验期间 Receiver 分别按算法 A 和算法 B 选择 U ,并进行速率和数据分配。为了测试算法对动态网络环境的自适应性,在分发视频期间,从 S 到 Receiver 的链路状态被不断地改变,而 Receiver 分别应用算法 A 和 B 在每次请求视频子段前通过重新测量 U 的丢包率来感知网络条件的变化:若 U 的发包丢失率超过阈值,则 U 立即被更新,并重新为 U 的成员分配发送速率和数据,否则 U 保持不变。实验中使用的主要参数见表 2。

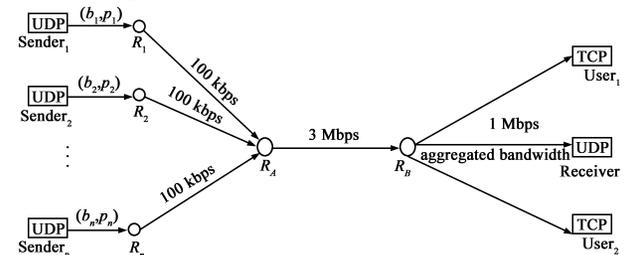


图2 网络拓扑

表 2 主要实验参数

chunk (KB)	pktsize (byte)	g	λ (kbps)	$b_i(t)$ (kbps)	$p_i(t)$	$T(s)$
14	1500	1	1000	5 ~ 100	0.001 ~ 0.3	249

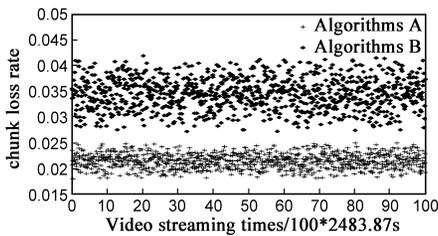


图3 1Mbps聚合带宽的平均丢包率

图3是 U 为 Receiver 提供 1Mbps 聚合带宽的平均丢包率,实验结果说明算法 A 每次都能得到比算法 B 平均丢包率更低的聚合带宽,因此在动态的网络环境下算法 A 优于算法 B.算法 A 在 100 次实验中的平均丢包率为 2.0143%,算法 B 为 3.5219%,算法 B 比算法 A 的平均丢包率高 1.5076%.若每个视频子段包含 2166 个包,则请求一个视频子段,算法 A 的平均丢包数为 43.629,算法 B 平均为 76.284,算法 B 比算法 A 平均多丢 32.655 个包.图4是 100 次实验中所使用的平均路径数.算法 A 使用的路径数平均为 15.278,算法 B 平均为 23.403,在相同的实验环境下,提供 1Mbps 的聚合带宽,算法 B 比算法 A 使用的路径数平均多 8.125,因此提供同样尺度的聚合带宽使用算法 A 的开销低于算法 B.

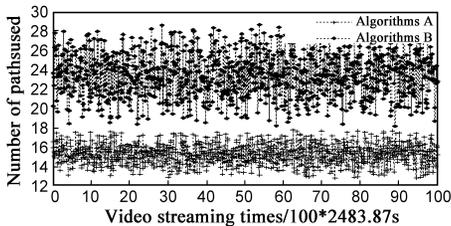


图4 1Mbps聚合带宽使用的平均路径数

5 结论

多发送端选择与最优速率分配是视频分发的关键技术之一,在理论和工程中具有良好的应用前景.本文提出了一种新的多发送端选择与最优速率及数据分配算法 MSRDA.不同于已有的算法只按链路的丢包率从低到高依次来选择发送端,MSRDA 根据到各发送端链路的丢包率与链路的可用带宽比升序排序来选择多发送端,并最优地分配发送速率和数据.理论分析和仿真实验结果表明:在不同的网络条件下 MSRDA 具有自适应性,能有效地减少聚合带宽的丢包率,优于现有同类算法^[4~7].

参考文献:

[1] 杨戈,廖建新,等.流媒体分发系统关键技术综述[J].电子学报,2009,37(1):137-145.

Yang Ge, Liao Jian-xin, et al. Survey of key technologies of

the distribution system for streaming media[J]. Acta Electronica Sinica, 2009, 37(1): 137-145. (in Chinese)

[2] 林珉,王新,等.面向大规模汇聚接入网的分布式 VoD 服务的性能仿真[J].电子学报,2008,36(10):2035-2039.

Lin Min, Wang Xin, et al. Performance simulation of distributed VoD service on large-scale converging access networks[J]. Acta Electronica Sinica, 2008, 36(10): 2035-2039. (in Chinese)

[3] 尹浩,林闯,等.大规模流媒体应用中关键技术的研究[J].计算机学报,2008,31(5):755-774.

Yin Hao, Lin Chuang, et al. Research on key technologies of large-scale streaming media[J]. Chinese Journal of Computers, 2008, 31(5): 755-774. (in Chinese)

[4] Hefeeda M, Habib A, Botev B, et al. PROMISE: Peer-to-Peer media streaming using CollectCast[A]. Proc The 11th ACM Int'l Conf on Multimedia[C]. New York: ACM Press, 2003. 45-54.

[5] Nguyen T, Zakhora A. Multiple sender distributed video streaming[J]. IEEE Transactions on Multimedia, 2004, 6(2): 315-326.

[6] Nguyen T, Zakhora A. Distributed video streaming over internet[A]. Proc SPIE/ACM MMCN[C]. New York: ACM Press, 2002. 36-49.

[7] Jurca D, Frossard P. Media flow rate allocation in multipath networks[J]. IEEE Transactions on Multimedia, 2007, 9(6): 1227-1240.

[8] Padhye J, Firoiu V, Towsley D, et al. Modeling TCP throughput: A simple model and its empirical validation[A]. Proc ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication[C]. Vancouver, CA, 1998. 303-314.

[9] Network Simulator 2 (Ns-2) [OL]. <http://www.isi.edu/nsnam/ns/>.

作者简介:



李泽平 男,1964年2月生于贵州贵阳,现为电子科技大学博士研究生;贵州大学副教授,硕士生导师.主要研究领域为计算机网络与流媒体技术.

E-mail: lizeping@uestc.edu.cn



卢显良 男,1943年生于河南洛阳,现为电子科技大学教授,博士生导师,主要研究方向为计算机网络与流媒体技术,高级操作系统等.

E-mail: xlu@uestc.edu.cn