视频流点播 Dynamic Batched Patching 算法研究

周建政, 蒋建国, 韩江洪, 齐美彬
(合肥工业大学计算机与信息学院, 安徽合肥 230009)

摘 要: 本文提出了一个新的视频流点播传输策略, 用以解决现有传输策略中存在的系统资源利用率低, QoS 较差等问题. 该策略的思想是服务器根据用户请求到达时刻, 按动态批处理的方式来接纳并服务请求用户, 每组用户必须同时从一个或两个信道接收视频内容. 文中对本策略的性能进行了理论推导与定量分析, 并与现有传输策略作了性能比较, 最后采用仿真实验对前面的理论分析与比较进行了验证. 理论分析及实验结果表明该策略是一个简单高效的传输策略, 适合任意规模的点播应用.

关键词: VOD; 视频流; 传输机制; Dynamic Batched Patching
中图分类号: TP37 文献标识码: A 文章编号: 0372-2112 (2004) 03-0452-05

Dynamic Batched Patching Schedules for Video Stream On-Demand Servers

ZHOU Jian-zheng, JIANG Jian-guo, HAN Jiang-hong, QI Mei-bin
(School of Computer & Information, HFUT, Hefei, Anhui 230009, China)

Abstract: A new transmission scheme called Dynamic Batched Patching was proposed to solve the problems, such as the low u-tilization of system resources and poor quality of service, existing in the existent schemes for video stream on demand. The key intuition behind this proposed schedule is that requests are dynamically batched according to the request arrivals, each batch of requests is served over one or two channels-either a regular channel alone or the combination of a regular channel and a patching channel. We de-rive a closed-form expression for the transmission channel requirements for this scheme. Our simulation experiments and theoretical arithmetic demonstrate that the proposed scheme can significantly outperform the existing schemes and it can be applied in random-scale On-Demand system.

Key words: VOD; video stream; transmission scheme; dynamic batched patching

1 引言

视频流业务可分为直播与点播两种模式, 绝大部分视频流应用都采用点播模式来实现. 在实现实时点播(尤其是在实现交互式点播)时, 需消耗大量系统资源(如网络带宽等). 在 IP 网络中(尤其是在窄带 IP 网络中)实现视频流点播, 这种资源与用户请求间的矛盾尤为突出, 严重阻碍了视频流点播应用的发展. 到目前为止, 绝大多数视频流点播系统都是应用在宽带环境(如局域网环境)下, 对其能同时提供服务的用户数有着严格的限制(通常只有几十到近百), 不能为远程点播请求提供良好的 QoS. 如何采用高效的点播传输机制来克服大型视频流点播系统中用有限的系统资源来满足用户的 QoS 是急需解决的问题.

2 现有视频流点播传输机制

现有视频流点播传输机制可分为两大类: 单播与多播.

在单播传输机制下, 点播应用的实现较简单, 客户端的启动延时小, 对交互能力的支持也较易实现. 但这样的系统所需的系统资源与用户的请求数基本成正比, 系统可扩展性差, 因而采用这种传输机制来提供点播服务的代价很大, 尤其在大型视频流点播应用中这种代价是不可接受的.

在多播传输机制中多个用户共享一个信道以减少对系统资源的消耗, 现有多播传输机制主要可分为两大类:

第一类是 Scheduled Multicast, 即系统根据用户请求的到达来分配系统资源. 这类传输机制主要有以下几种算法. (1) Dan 等[1]提出的 Batching 算法; (2) Hua 等[2]提出的 Patching 算法; (3) Golubchik 等[3]提出的 Piggybacking 算法; (4) Cater 等[4]提出的 Streaming Tapping 算法等.

第二类是 Periodic Broadcast, 即系统预留信道周期性广播视频文件. 这类机制主要有以下几种算法: (1) 文献[5]中提出的 Staggered Broadcasting 算法; (2) 各种分段算法如 Pyramid Broadcasting[6], Skyscraper Broadcasting[7], Harmonic Broadcast-ing[8]等算法. 最后, Kalva 等[9]提出了 Asynchronous Multicasting 算法.

收稿日期: 2003-01-16; 修回日期: 2003-08-25
基金项目: 安徽省自然科学基金(No. 00043235); 教育部基金项目(No. 00071)

但现有的这些点播传输机制不能完全解决点播应用中存在的问题, 如所需信道多, 客户端启动延时大, 需很大的客户端缓存空间(> 50% 整个视频流文件) 与客户端 I/O 带宽, 且需复杂的接收机制或复杂的先验信息等, 从而很难在实时视频流点播上应用.

3　视频流点播 Dynamic Batched Patching 算法

本文提出一个 Dynamic Batched Patching (简称为 DBPatching) 算法用来解决现有传输机制中所存在的问题. 下面具体介绍该算法.

311　术语定义与说明

首先对本文中将要用到的几个术语与符号作出说明或定义.

说明: 假设某视频文件 i 的长度为 L, 客户端最大允许启动延时为 D0, 最大缓存容量为 B0, 最多可同时从两个信道接收视频流. 视频 i 的请求率 K(i) 在本文中指在一段时间内请求该视频文件的请求个数.

定义 1　传输整个视频文件的多播称为 Regular Multicast, 其对应的信道称为 Regular Channel(简称 R 信道), 在该信道中传输的视频流称为 Regular Stream(简称 R 流). 而只传输视频文件前一部分的多播称为 Patching Multicast, 其对应的信道则称为 Patching Channel(简称 P 信道), 所传输的视频流称为 Patching Stream(简称 P 流). R 流与 P 流的速率都与视频流的回放速率相同. 在本文中, 信道指为用户提供一个视频流服务所需的系统资源, 也称逻辑信道.

定义 2　以某请求到达时刻(如图 1(a) 中的 t0, t_j) 为起始时刻的长为 B 的时间段定义为一个 B 域(如图 1 所示). B(B - DF B0) 的值为预先设定. 并定义最近请求所在的 B 域为当前 B 域, 当前 B 域的起始时刻设为 Rb. B 域的特点是同一 B 域内到达的所有请求将共享系统为该 B 域分配的 R 信道, 接收该 R 信道中的 R 流.

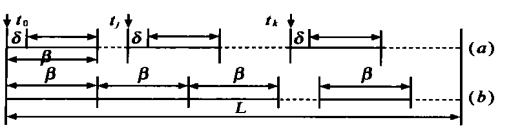

图 1　L 长度内 B 域的分布情况

定义 3　以某请求到达时刻(如图 2 中的 t_r, t_m) 为起始时刻的长为 D 的时间段定义为一个 D 域, 有一例外是任意 B 域内的最后一个 D 域的长度可能小于 D(如图 2(b) 所示). D(DF D0) 为预先设定的任意请求组的启动延时. 并定义当前请求所在的 D 域为当前 D 域, 当前 D 域的起始时刻设为 Pb. 同一 D 域(每个 B 域内的第一个 D 域除外) 内到达的所有请求共享系统为该 D 域分配的 P 信道, 接收该 P 信道中的 P 流, 同时共享系统为该 D 域所在 B 域分配的 R 信道, 接收该 R 信道中的 R 流. 第一个 D 域内到达的请求用户只需共享系统为其所在 B 域分配的 R 信道, 接收该 R 信道中的 R 流.

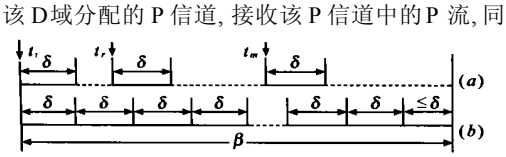

图 2　一个 B 域中 D 域的分布情况

312　DBPatching 算法设计

我们首先只对请求一个视频文件的情况进行讨论. 为讨论方便假设由于网络和服务器调度引起的延时为零.

当在一段时间内的第一个用户请求到达时(记为 t0 时刻), 这时将同时创建一个新的 B 域与一个新的 D 域, 即 Rb= Pb= t0. 在该 D 域结束时系统将为该 D 域及该 B 域分配一个新的 R 信道来服务该域内的所有请求. 假若之后某请求同一视频文件的请求于时刻 t 到达. 可分两种情况进行讨论:

Case1:

如果请求到达时刻 t 与当前 B 域的起始时刻 Rb 满足 t- Rb[B, 则该请求属于当前 B 域. 满足这个条件的请求又可分两种情况:

Case1.1 如果 t 与当前 D 域的起始时刻 Pb 满足 t- Pb[D 时, 即该请求属于当前 D 域. 该请求用户共享系统为当前域分配的 R 信道与 P 信道, 并接收相应信道中的视频流内容.

Case1.2 如果 t 与当前 D 域的起始时刻 Pb 满足(t2Pb)> D 时, 该请求不属于当前 D 域, 则以该时刻为起始时刻创建一个新的当前 D 域, 即 Pb= t. 系统将于当前 D 域结束时为当前 D 域分配一个 P 信道, 当前 D 域的请求在接收该 P 信道中的 P 流的同时立即缓存当前 B 域所属的 R 流.

Case2:

如果请求到达时刻 t 与当前 B 域的起始时刻 Rb 不满足(t- Rb) F B, 则该请求不属于当前 B 域肯定也不属于当前 D 域. 则立即以 t 时刻为起始时刻同时创建一个新的当前 B 域与一个新的当前 D 域, 即 Rb= Pb= t, 并为该当前域分配新的 R 信道传输相应的 Regular Stream.

当请求于任意时刻 t 到达时, 按以上条件进行判断, 将其分配到当前 B 域与 D 域或新的 B 域与 D 域, 以共享当前域所属的信道或重新分配新的信道.

DBPatching 算法流程如图 3 所示, 图中没有描述信道的回收. 由于实际网络延时非零, 不宜采用服务器来判断是否可回收某信道, 而是当客户端检测到两种信道中的视频流已同步或该视频文件已播放完毕后立即通知服务器回收相应信道.

需要说明的是 R 流与 P 流都有生命周期. 由定义 1 可知 R 流的生命周期为 L. 对于当前 B 域内除第一个 D 域外的所有 D 域来说, 在该 D 域结束时该 D 域的所有用户立即接收并缓存当前 B 域的 R 流, 同时接收并回放系统为该 D 域分配的 P 流来弥补比当前 B 域所属的 R 流所缺的视频内容. 由于 R 流与 P 流的速率都与视频流回放速率相同. 所以, 在(Pb- Rb) 时间后, P 流将与缓存中的 R 流同步. 这里所指的同步是指 P 流中的当前回放点已存在于被缓存的 R 流中. 这时系统回收该 P 信道. 当两种信道同步后, 该 D 域的用户在剩下的时间内继续接收、缓存相应 R 流, 同时回放线程将从缓存中读取数据进行回放. 显然, 当前 D 域所属的 P 流的生命周期为(Pb2Rb), 值得说明的是任意 B 域内的最后一个 D 域所属的 P

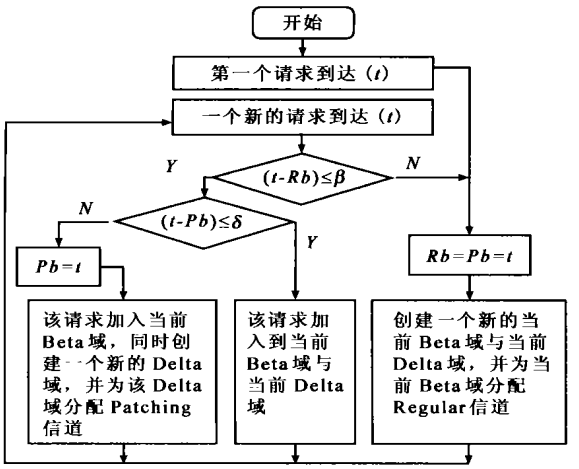

图3 Dynamic Batched Patching算法流程图

流的生命周期为 B- D,其值小于等于(Pb2Rb).

3.3 算法性能分析及证明

算法性能分析主要以定理的形式给出并作出相应证明.

定理 1 客户端的最大启动延时为 D,平均启动延时为 D/2.

证明 本结论的前提是暂时不考虑网络与服务器端的延时,在本算法中 D是事先根据目标用户的要求来确定的(即 D F D0). 由于所有请求都将属于一个当前 D域,并在当前 D域结束时,当前 D域内所有请求用户才得到服务. 当前 D域中延时最大的请求是当前D域的第一个请求,延时为 D. 延时最小的是最后进入当前 D域的请求,延时为 0. 由于请求用户是随机进入的,任意请求用户在当前 D域内任意时刻进入的概率相等,所以对于任意一个用户来说其平均启动延时为 D/2.

定理 2 任意当前时刻系统为某视频所分配的 R 信道最多为7 L/ B6个.

证明 在实际点播过程中,由于 R 信道的生命周期为 L,所以在任意当前时刻该视频所占用的 R 信道数必然是系统在距当前时刻长度为 L 的时间段内为该视频所分配的所有 R 信道数. 由于任意请求都属于一个 B域,每个 B域内的所有请求共享同一R 信道,所以当且仅当 L 是一个由B域(互不重叠) 构成的完全划分时如图 1(b) 所示,在任意当前时刻系统为该视频所需分配的 R 信道数最多,且为7 L/ B6.

定理 3 任意当前时刻系统为某视频所分配的 P 信道最多为õ7 B/ D6/ 28个.

证明 在一个 B域内存在若干个互不重叠的D域,该 B域内每个D域(第一个 D域除外) 内的请求除共享系统为该 B域分配的 R 信道外还将共享系统为该 D域分配的 P 信道. 与定理 2同理,当且仅当 L 是由D域构成的一个完全划分时,任意当前时刻某视频所占用的 P 信道数最多.

在当前 B域内系统除为第一个 D域分配一个 R 信道外,将为其余每个 D域分配一个 P 信道. 至第 i(i I [0, 7 B/ D6- 1])个 D域结束时系统在当前B域内曾为该视频分配的 P 信道为 i 个. 由于 P 信道存在生命周期,在当前 B域内第 n 个D域结束时分配的 P 信道在第 2n 个D域结束时会被回收. 所以在当前 B域内第 i 个D域结束时系统在当前B域内为该视频分配但尚未回收的 P 信道数为 i- õi/ 28= 7 i/ 26个.

而且在当前 B域内还有系统曾为上一个B域所分配但尚未释放的 P 信道. 由于在上个 B域内系统曾为该视频分配的 P 信道数为(7 B/ D6- 1),到当前 B域内第 i 个D域结束时已释放的 P 信道为(õ(7 B/ D6+ i)/ 28). 所以当前时刻在上个 B域内分配但尚未回收的 P 信道数为((7 B/ D6- 1)- (õ(7 B/ D6+ i)/ 28)) = 7 7 B/ D6/ 2- 1- i/ 26.

所以在任意当前时刻系统为该视频分配的 P 信道数最多为:

$$7\,7B/D6/2-1-i/26+7\,i/26=\begin{cases}7B/D6/2-1 & 7B/D6=\text{Even},\ i=\text{Even}\\ 7B/D6/2 & 7B/D6=\text{Even},\ i=\text{Odd}\\ õ7B/D6/28 & 7B/D6=\text{Odd},\ i=\text{Even}\\ õ7B/D6/28 & 7B/D6=\text{Odd},\ i=\text{Odd}\end{cases}\quad(1)$$

由式(1)可知任意当前时刻系统为该视频分配的 P 信道将不多于õ7 B/ D6/ 28

定理 4 任意当前时刻系统为某视频所分配的信道数不超过7 L/ B6+ õ7 B/ D6/ 28,且与 K无关.

证明 任意当前时刻系统为某视频分配的信道总数为两种信道数之和. 显然,当这两种信道数各取最大值时其和最大. 由定理 2与定理 3 可知,当且仅当 L 是一个由D域构成的完全划分时,任意当前时刻系统为该视频分配的信道数最多,其值为

$$7L/B6+õ7B/D6/28\quad(2)$$

定理 5 客户端所需的最大缓存容量为 $\sqrt{2LD}-D$.

证明 为进一步优化系统性能使得任意当前时刻系统为该视频文件最多分配的信道数最少. 由初等数学中的定理 a + bE2 $\sqrt{a@b}$(a= b 时取等号) 可知,当7 L/ B6= õ7 B/ D6/ 28时式(2) 可取最小值. 但由于该条件不便讨论,我们用 L/ B = B/ 2D即 B= $\sqrt{2LD}$代替上面的条件代入式(2) 中,得出的最多信道数肯定不是最优,而是一次优解. 在实际点播应用中我们可以依据这个次优解来进行参数设置,即在根据用户的要求设定 D后,由 B= $\sqrt{2LD}$可得到次优解时的B值,这两个值也就是实际应用中 B域与D域的大小. 显然,在这个条件下客户端所需的最大缓存容量也就是P流的最大生命周期为 B- D= $\sqrt{2LD}-D$.

3.4 与现有算法的比较

下面通过举例将本文提出的算法与现有的重要算法作一性能比较.

由定义说明可得到表 1 中本文算法的前三个指标(A表示客户端所需 I/ O带宽或需同时接收的视频流数). 而接收机制的复杂度是指客户端需不需复杂的信道间同步与信道切换机制,本算法中客户端在接收视频流时无需进行信道间同步与切换,所以接收机制简单. 表中的 L/N 说明客户端的启动延时与视频文件长度及其分段数有关.

当 D= 7 1 5sec 时,由 B= $\sqrt{2LD}$可得到B为 5 1 477min. 将 D与B代入式(2) 中可知在此情况下该视频在任意当前时刻所占用的信道数不大于 44. 表 2 中的/ [0表示最坏情况下所需

的信道数. 表 2 中 K 代表在该算法中该视频共分为 K 段, 根据不同的分段算法, K 将取不同的值. 在我们讨论的条件下, 对于 Pyramid Broadcasting 算法, K= 10; 对于 Skyscraper Broadcasting, K= 16.

表 1　各种视频流点播传输机制的技术参数比较

系统参数 / 点播传输机制	D	B	A	接收机制
D B Patching	F D0	F B0	1~ 2	简单
Unicast	F D0	F B0	1	简单
Patching	F D0	B0	1~ 2	简单
Stream Tapping	F D0	10~ 30	2~ 4	复杂
Staggered Multicast	L/ N	0	1	简单
Segmentized Broadcast	F D0	10~ 90	2~ 6	复杂

表 2　各种点播传输机制在不同请求率情况下满足相同所需的信道数比较(L= 120min)

K(D= 715 秒) / 点播传输机制	6	60	600	6000
D B Patching	[6	[44	[44	[44
Unicast	6	60	600	6000
Patching	F 6	[60	[600	[6000
Stream Tapping	F 6	[60	[600	F 6000
Staggered Multicast	960	960	960	960
Segmentized Broadcast	K	K	K	K
这时缓存空间约需 5. 352min 视频流文件大小				

在 D B Patching 算法中, P 流的平均生命周期为 7 B/ D6 @ D/ 2, 而在一个 L 长度时间内服务器为该视频分配的 P 信道数最多为 7 L/ D6 - 7 L/ B6, 所以在 D B Patching 算法中某视频在任意时刻所占用的平均信道数不超过:

$$7L/B6 + \frac{7B/D6 @ D@(7L/D6 - 7L/B6)}{2L} \qquad (3)$$

而在 Patching 算法中, P 流的平均生命周期为 B/ 2. 所以在 Patching 算法下该视频在任意时刻所占用的平均信道数为:

$$\frac{[(K- 7L/B6) @ B/2]}{L} + 7L/B6 \qquad (4)$$

将 L= 120min, D= 715sec, 代入式(3)、(4) 中可得表 3. 表 3 中的 K 与表 2 中的 K 完全相同.

从表 1 可得出能同时满足 DF D0, BF B0, A[1~ 2 且客户端接收机制简单的传输机制只有 Patching、Unicast 与 D B Patching 算法. 由表 2, 表 3 可知实现点播系统时需为该视频分配的信道数较少的有 Segmentized Broadcast 与 D B Patching 算法. 综合三表中这些参数可以发现本文提出的 D B Patching 算法在实现点播时较现有算法优越, 尤其是系统规模越大该算法的优点越突出.

表 3　一个视频文件在某时刻平均占用信道数比较

K(D= 715 秒) / 点播传输机制	6	60	600	6000
D B Patching	[6	[2218 7	[3512	[4315
Unicast	6	60	600	6000
Patching	[6	2218 7	3512	15814
Staggered Multicast	960	960	960	960
Segmentized Broadcast	K	K	K	K

315　D B Patching 算法仿真实验

本节给出在不同请求率条件下 (3000/ 2h、6000/ 2h) 对 D B Patching 算法与 Patching 算法进行一系列的仿真实验. 实验仿真了 4 个小时内用户请求某视频文件时系统按这两种算法为该视频所分配、回收信道的总的情况. 仿真实验结果如图 4~ 7 所示. (在仿真实验中, 两种算法中的 B 取相同的值, D B Patching 算法中的取 715sec)

以上实验结果可证明: (1) 本文 313 节与 3. 4 节中的算法性能分析是完全正确的; (2) 本算法的性能较 Patching 优, 且随着请求率的增大 D B Patching 算法的优越性越来越明显.

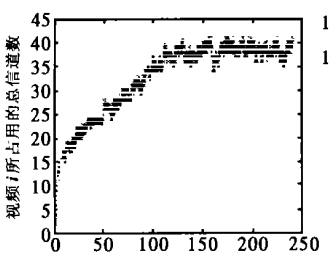

图 4　D B Patching 算法在 K= 3000/ 2h 时所需的实时总信道数

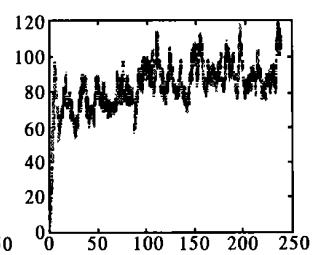

图 5　Patching 算法在 K= 3000/ 2h 时所需的实时总信道数

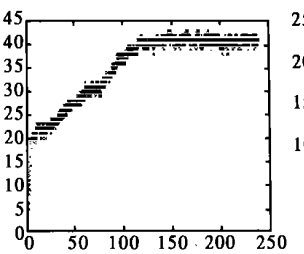

图 6　D B Patching 算法在 K= 6000/ 2h 时所需的实时总信道数

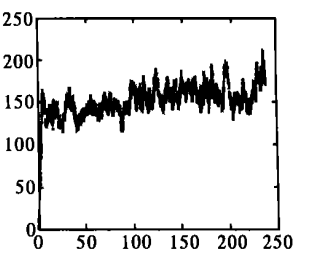

图 7　Patching 算法在 K= 6000/ 2h 时所需的实时总信道数

4　结论

本文提出的算法有以下几个特点: (1) 启动延时少(DF D0); (2) 所需的客户端缓存容量满足 BF B0; (3) 所需客户端 I/ O 带宽[2 倍回放速率; (4) 对每一个视频, 系统只需为其分配相对较少的信道, 系统资源利用率高, 而且用户请求率越高系统资源利用率也越高. 因此, 在给定系统资源条件下可以满足更多用户对更多视频文件的请求, 同时适合任意规模的

点播应用;(5) 客户端接收回放机制简单;(6) 本算法实现简单,对于系统中所有的视频及对这些视频的请求都采用同一算法,无需根据复杂的先验信道来预留信道.

从以上几点可以看出,该算法既具有 Unicast 与 Patching 等 Scheduled Multicast 算法的优点(启动延时小,对客户端缓存容量需求小,客户 I/O 带宽小,接收机制简单等),同时也具有分段算法的优点(资源利用率高),解决了现有算法中所存在的问题,能为用户提供很好的 QoS,完全可以用来实现视频流点播应用.另一点需说明的是,一个实际的点播系统必需提供对交互功能的支持,因为直接用多播技术来实现对交互功能的支持不太方便.本文暂没有考虑对这种功能的支持,它将是我们下一步要解决的问题.

参考文献:

[1] A Dan, et al. Scheduling policies for an on2demand video server with batching [A]. Proc ACM Multimedia. 94 [C]. San Francisco, CA, USA: PACMM, 1994. 15- 23.

[2] Y Cai, et al. Optimizing patching performance [A]. In Proceedings of Multimedia Computing and Networking [C]. San Jose, CA: SPIE 3654, 1999. 204- 15.

[3] L Golubchik, et al. Adaptive piggybacking: A novel technique for data sharing in video2on2demand storage servers [J]. ACM Multimedia Syst, 1996, 4(30): 14- 55.

[4] S W Carter, et al. An efficient implementation of interactive video2on2demand [A]. Proceedings of the 8th International Symposium on Mod2eling, Analysis and Simulation of Computer and Telecommunication Systems[C]. San Fancisco, CA: MASCOTS, 2000. 172- 179.

[5] K Almeroth, et al. On the use of multicast delivery to provide a scalable and interactive video on2demand service [J]. IEEE Journal on Selected Areas in Communications, 1996, 14(6): 1110- 1122.

[6] S Viswanathan, et al. Metropolitan area video2on2demand service using pyramid broadcasting [J]. ACM Multimedia Syst, 1996, 4(4): 197- 208.

[7] KA Hua, S Sheu. Skyscraper broadcasting: a new broadcasting scheme for metropolitan video2on2demand systems [A]. In Proceedings of[C]. Cannes, France: SIGCOMM 97, 1997. 89- 100.

[8] L2S Juhn, L2M Tseng. Enhanced harmonic data broadcasting and re2ceiving scheme for popular video service [J]. IEEE Transactions on Consumer Electronics, 1998, 44: 343- 346.

[9] H Kalva, B Furht. Techniques for improving the capacity of video2on2demand systems [A]. Proceedings of the 29th Hawaii Int Conference on System Sciences [C]. Wailea, Hawaii: HICSS, 1996. 308- 315.

作者简介:

周建政 男,1976 年 8 月生于湖南省湘乡市,合肥工业大学讲师,在职博士生,主要研究方向:视频编码与视频流,数字信号处理,大规模集成电路设计等. Email: xiao_zho@yahoo.com.cn.

蒋建国 男,1955 年 10 月生于安徽省,合肥工业大学教授,博士生导师,主要研究方向:数字图像分析与处理,DSP 技术应用等.

韩江洪 男,1954 年 9 月生于安徽省广德,研究员,博士生导师,现任合肥工业大学副校长,安徽省政府技术专家委员会委员,全国电子信息类专业教学指导委员会委员,中国仪器仪表学会微型计算机学会理事长,主要研究方向:计算机控制、信息系统、智能家居及信息家电等.