

基于微云的移动视频预取系统

纪传伟¹ 白光伟^{1,2} 沈 航¹ 邹露宁¹

(南京工业大学计算机科学与技术系 南京 210009)¹

(南京理工大学高维信息智能感知与系统教育部重点实验室 南京 210094)²

摘 要 现有的移动视频通话研究大多集中在视音频同步、码率调整和能量节约等方面,忽视了用户在公共 AP 间切换时视频通话服务的中断重连问题。针对这一问题,提出了基于微云的移动视频预取系统来尽可能地减少通话重连次数。该系统通过本地微云为移动的用户预取视频流,并设计贪心预取算法来提高预取命中率。同时,设计低概率多阶马尔科夫预取进程调度算法和起始 AP 选择算法来平衡贪心预取引起的微云资源消耗过多的问题。仿真实验表明,CMVP 可以获得 90% 以上的预取命中率,使 QoE 显著提升。

关键词 公共 AP, 视频预取, 本地微云, 多阶马尔科夫

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.012

Cloudlet-based Mobile Video Prefetching System

Ji Chuan-wei¹ Bai Guang-wei^{1,2} Shen Hang¹ Zou Lu-ning¹

(Department of Computer Science and Technology, Nanjing Tech University, Nanjing 210009, China)¹

(Key Laboratory of Intelligent Perception and System for High-Dimensional Information of Ministry of Education of China, Nanjing University of Science and Technology, Nanjing 210094, China)²

Abstract Most researches on mobile video communication(MVC) focus on audio and video synchronization, bit rate adjustment and energy saving, however, the change between APs which leads video communication service to interruption is not considered. We proposed a cloudlet-based mobile video prefetching system(CMVP) to this challenge. CMVP utilizes cloudlets to prefetch video streams for mobile-phones. We designed a greedy prefetching algorithm to reduce the count of service interruption as far as possible. And to balance cloudlets' resource consumption, the low probability K to 1 markov prefetching progress schedule algorithm(LPMPPS) and first AP selection algorithm were designed. Our simulation results demonstrate that CMVP can access to more than 90% hit rate, and QoE can be improved remarkably.

Keywords Public AP, Video prefetching, Cloudlets, K to 1 Markov

1 前言

得益于无线通信技术(3G/4G、Wi-Fi)的发展和移动终端(如智能手机)的升级换代,移动视频通话^[1-3](Mobile Video Communication, MVC)打破了距离带给人们面对面交流的限制,越来越受到人们的欢迎,同时也促使了很多 MVC 应用软件^[4]的兴起。目前绝大多数的视频通话都是基于 P2P 的方式实现的^[5],通话双方先建立 TCP 连接,随后将视频流封装成 UDP 包相互发送。由于当前我国的 3G/4G 网络覆盖不完全且流量资费昂贵,此时公共 AP(按时计费或者免费)成为必然的选择。但是用户(终端)持续移动造成的 AP 多次切换,使得通话双方多次 TCP 重连,严重降低了用户体验(QoE)。

针对上述问题,本文提出基于微云的移动视频预取系统 CMVP,从服务提供角度将系统的体系结构分为 3 层,即移动终端、cloudlet 和 cloud,如图 1 所示。每一个 AP 有线连接一个本地微云(cloudlet),微云是具有有限处理能力的服务器。图 1 中,手机 A 正连接 AP_A 与远端手机 B 进行视频通话,微云_A与微云_B分别在本地为手机 A 和 B 代理视频通话。在手机 A 移动出 AP_A 的有效覆盖范围之前,云层(cloud)为其在 AP_A 所有的邻近 AP 所连的微云上预取来自手机 B 的视频流。AP_A 的邻近 AP 是指与 AP_A 覆盖区域有重叠的 AP。CMVP 在云层设计贪心预取算法来获得可接受的预取命中率,从而有效减少 TCP 重连次数并提高 QoE。同时设计起始 AP 选择算法和低概率多阶马尔科夫预取进程调度算法(Low Probability K to 1 Markov Prefetching Progress Schedule,

到稿日期:2015-04-20 返修日期:2015-07-20 本文受国家自然科学基金(60673185, 61073197),江苏省自然科学基金(BK2010548),江苏省科技支撑计划(工业)(BE2011186),南京邮电大学宽带无线通信与传感网技术教育部重点实验室开放研究基金资助课题项目(NYKL201304),江苏省六大高峰人才基金(第八批)资助。

纪传伟(1989—),男,硕士生,主要研究方向为移动云计算、多媒体网络, E-mail: ji_chuanwei@126.com;白光伟(1961—),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为移动互联网、云计算、网络体系结构和协议、网络系统性能分析和评价、多媒体网络服务质量等;沈 航(1984—),男,博士,讲师,主要研究方向为云计算、移动互联网、无线多媒体通信协议等;邹露宁(1989—),男,硕士生,主要研究方向为移动云计算、多媒体网络。

LPMPSS)来平衡服务区中微云的负载。

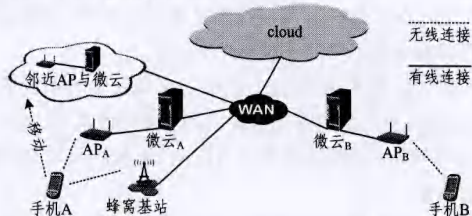


图1 CMVP三层结构示意图

本文第2节是相关工作和问题的提出；第3节详细介绍CMVP系统的体系结构各层的交互；第4节给出相关算法；第5节是仿真实验与性能分析；最后总结全文。

2 问题的提出

为了提升移动视频的服务质量，文献[6]提出一种云辅助的框架，通过调节云端-终端资源分配比率、信道带宽-码率权重，来最大化蜂窝网络下移动视频的服务QoE。该框架在提高移动视频服务质量的同时有效降低了终端的额外能量开销，但是蜂窝网络下昂贵的视频服务费用限制了它的实用性。

文献[7]针对无线网络性能的不确定性对移动应用任务迁移效果带来的不良影响，提出一种三层体系结构ENDA。ENDA通过预测用户踪迹来选择最优的AP，完成终端向云层的任务迁移，从而实现终端能耗最小化。然而ENDA在AP切换时不能支持迁移任务，这显然不能适应移动视频服务，并且它的踪迹预测算法需要用户不断地利用蜂窝网络上自己的GPS信息，这增加了终端额外的能耗和流量。最后ENDA仅简单地利用GPS位置点的匹配来预测用户位置，并未对预取命中率有任何分析和实验。

本文有针对地选择了公共AP环境来避免蜂窝网络流量昂贵的问题。另外本文不同于文献[7]需要连续的位置信息，所以几乎不需要终端消耗额外的能量和流量。最后本文不同于文献[8,9]等通过研究用户行为或者位置相关性来预测用户下一时刻最可能出现的位置或者区域，然后进行预取工作，而是设计贪心预取算法来获得可接受的预取命中率。本文的目标是尽可能减小Wi-Fi切换对QoE的不利影响，手段是提高预取命中率来尽可能减少中断重连次数。下面将给出命中率与QoE的关系，但首先要探讨公共AP环境下视频通话的用户QoE模型及其参量。

通常来说，用户能够很直观地感受到通话中断重连次数、视频失真和视音频同步这3个因素对视频通话服务的影响。故本文对公共AP环境下的用户QoE模型参量的设定如下。

- 中断重连次数：移动中的用户切换AP时需要中断视频通话服务，完成AP切换后才能重新建立通话，所以此时通话重连次数等于AP切换次数。

- 视频质量：影响用户评价视频质量的因素有很多，如由网络性能引起的画面失真、噪音等，文献[10]给出了视频质量的量化标准。

- 视音频同步时间：网络延迟和抖动造成的视频和音频播放时间差，文献[10]给出了用户对视音频同步时间的可接受范围(160ms以内)。

- AP切换时间：终端切换AP时会产生一定的时延，而国际电信联盟推出的ITU-TG. 114建议VoIP类型业务的AP切换时间应在150ms之内。

中断重连导致用户QoE降低的原因主要有：TCP连接的重新建立延时 T_{TCP} 、通话对象的视频流重发的到达时间 T_L （通话双方链路延迟）和原链路上UDP数据包的丢失，而且 T_L 越大对QoE的影响越明显。

预取命中率就是在整个通话服务中，CMVP为用户成功预取的次数与用户AP切换次数的比值。实际环境下每个微云的带宽有限，预取进程数目有限，故会出现预取失败的情况，即用户切换到邻近AP时，微云并未预先为其建立预取进程。考虑到不同用户切换AP的次数并不相同，本文设定预取命中率 P_S 为过去 T' 时间内服务区中所有用户预取成功总次数与所有用户AP切换总次数的比值。

文献[10,11]针对IPTV服务综合考虑视频质量 V_q 、视音频同步时间差 t_{sync} 和频道切换时间 t_{swp} ，得出了一个用户体验模型 QoE_U 。而本文重点针对视频通话断开重连问题，采用与 QoE_U 相同的思考角度，提出了自己的QoE模型：

$$QoE = \frac{\log_{10}(u_q)}{K_1 \cdot T + n_w \cdot T_{Handover} + t_{sync}} \quad (1)$$

其中， K_1 表示影响因子，它是一个常量， T 表示AP切换造成的视频中中断重连时间， n_w 表示用户整个通话过程中AP的切换次数， $T_{Handover}$ 表示终端的AP切换时间。

$$T = n \cdot (T_{TCP} + T_L) \quad (2)$$

式中， n 表示通话中断重连次数。由式(1)、式(2)可推出预取命中率与用户QoE的关系如下：

$$QoE = \frac{\log_{10}(u_q)}{K_1 \cdot (1 - P_S) \cdot n_w \cdot T + n_w \cdot T_{Handover} + t_{sync}} \quad (3)$$

从式(3)可以看出，提高 P_S 将有效减少整个通话的重连次数，显著提高用户QoE。而怎样获得可接受的 P_S 是CMVP要解决的主要问题。下一节将介绍CMVP各层交互的细节。

3 体系结构

CMVP系统的体系结构分为3层：移动终端、(本地)微云和(远端)云层，各层相互联系、互相协作。CMVP在不额外增加终端工作量的同时，利用微云的有限处理能力，采用贪心预取策略，以较大的预取命中率为移动的用户预先建立TCP连接，预先接收通话视频流，尽可能地减少用户多次切换AP情况下的通话重连次数。图2详细描绘了CMVP的体系结构。

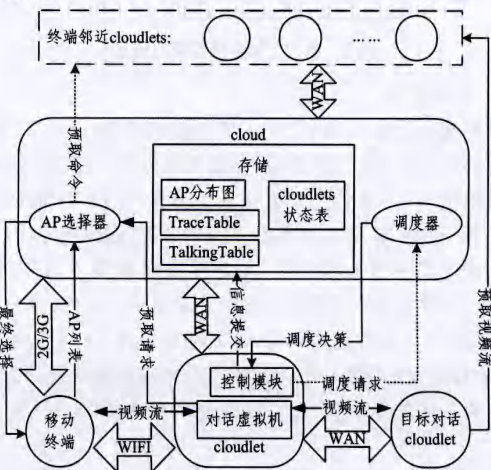


图2 CMVP各层交互图

3.1 移动终端与云层交互

在 CMVP 中,终端只需要在通话服务开始前将其能搜索到的 AP 按照终端接收到信号的强度由强到弱排列成表,并通过 2G/3G 网络发送到云层,当前的 Wi-Fi 技术和终端处理能力完全能做到这点。这样云层就可以权衡无线信号强度与微云负载情况(计算负载和带宽负载)在服务区内为终端选择首个 AP,从而在保证无线信号质量的同时平衡服务区中的各个微云的负载。而正在通话的终端切换 AP 时,为减少终端与 AP 的重连时间和终端额外的能量消耗,仅自动连接无线信号强度最强的 AP。用户的 AP 踪迹和通话信息均由微云收集并上传给云层保存,避免给终端增加额外的工作量。

3.2 本地交互

CMVP 中微云与 AP 直接相连,并且通过 WAN 对外交互。微云建立通话虚拟机来代理终端视频通话,并为地理上邻近的用户建立预取进程来预取远端通话视频流。终端只需要向微云发送自己的采样视频流,并从微云接收由远端通话者发送过来的已被通话虚拟机重新编码过的视频流。预取进程仅仅实现一个 TCP 连接的建立和远端视频流的接收,并不缓存和处理视频流,所以计算和存储开销极小,但是带宽占用明显,以 Skype 为例,大约是通话虚拟机带宽占用的 40%~60%^[12],具体还要由视频流的编码和码率控制等因素决定。通话虚拟机主要负责:

1) 通话管理。负责向云层提交通话信息(通话目的地址和用户当前所连的 AP)。

2) 请求预取。实时视频通话的终端需要不间断地上传视频流,那么通话虚拟机的 RFID 检测模块就可以根据现有的无线定位技术^[13]来监测终端与其所连 AP 的距离,当距离超过设定阈值 d 时,通话虚拟机向云层上的 AP 选择器发出预取请求,而这不会增加终端的工作量。

3) 有效的视频缓冲与视频流处理。降低终端处理视频流的额外能耗,同时支持不同类型的移动设备,这是一个通用框架,适应各种编解码策略。由于视频预取的需要,虚拟机承担着远端通话视频流的接收和本地视频流的多播发送任务,如图 3 所示。

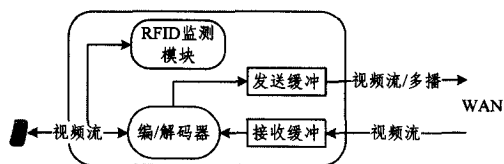


图3 微云上的通话虚拟机模型

3.3 云间交互

本地微云与云层的交互主要由控制模块(见图2)负责,它时刻监测微云的负载信息,并提交给云层。控制模块在新的虚拟机建立后便向云层发送通话信息(用户 ID、通话目的地址和当前所连 AP),并在微云达到带宽负载阈值时向云层上的调度器发出调度请求,而调度器响应调度请求,执行 LPMPPS 算法并返回调度决策给控制模块。另外,微云上的通话虚拟机在监测到终端与 AP 的距离超过 d 时向云层上的 AP 选择器发出预取请求,而 AP 选择器响应请求并命令用户邻近的所有微云为用户建立预取进程来预取视频流(贪心预取)。

云层上的存储服务器负责组织微云提交的各种信息并保存,以供 AP 选择器和调度器处理,这些组织后的信息如下:

1) AP 分布图,即服务区中所有 AP 的分布图,而每一个 AP 都与一个微云连接,所以这也是微云分布图,这样云层可以轻松得到通话用户的邻近 AP 和微云。

2) 微云状态表,记录服务区中所有微云实时提交的负载信息,包括计算负载和带宽负载。基于实际成本考虑,本文认为微云与 WAN 之间的带宽是有限的,而贪心预取会导致微云带宽不足。

3) 正在服务用户表,如表 1 所列,记录服务区中所有正在通话的用户的信息,包括当前所连 AP、通话目的地址和 AP 踪迹。微云一旦为用户建立了通话虚拟机,便向云层提交信息来触发表 1 中记录的建立或者更新。

表1 正在服务用户

用户标识	AP	通话目的地址	AP 踪迹
1	W ₆	10.8.250.135	W ₄ , W ₅

4) 历史用户踪迹表,如表 2 所列,收集过去 T 天内的历史用户的 AP 踪迹信息作为多阶马尔科夫预测模型的学习数据。当用户主动结束通话或者长时间离开服务区时,云层将删除表 1 中对应记录,并且将其 AP 踪迹保存在表 2 中。

表2 历史用户踪迹

踪迹编号	AP 踪迹	建立时间
1	W ₁ , ..., W _i , ...	ss-mm-hh-DD-MM-YY

下一节将介绍 CMVP 的两个主要功能模块:AP 选择器和调度器,以及在这两个模块上执行的 3 个算法(起始 AP 选择算法、贪心预取算法和 LPMPPS 算法)。

4 功能模块

CMVP 最主要的两个功能模块(AP 选择器和调度器)被放在云层上运行,以获取最快的响应速度。AP 选择器执行贪心预取算法和起始 AP 选择算法,而 LPMPPS 算法被放在调度器上执行。

4.1 调度器

调度器响应带宽负载达到阈值 μ_b 的微云的预取进程调度请求,执行 LPMPPS 算法,并返回调度决策给微云,从而达到降低服务区微云带宽负载的目的。LPMPPS 算法的核心是多阶马尔科夫预测模型,以此来计算用户的多阶转移概率,从而得出用户向邻近 AP 切换的概率。

多阶马尔科夫模型是 K 到 1 阶马尔科夫模型的联合,随着用户踪迹长度的增加,其可用的马尔科夫模型阶数也增加,转移概率更加可靠,预测精度随之提高。从常理上也可以理解这一点,采用的马尔科夫模型阶数越高,需要匹配的用户踪迹长度越长,用户由当前状态转向邻近状态的转移概率越不均匀,马尔科夫预测模型作用也越明显。本文采用时齐的离散时间参数的马尔科夫链。时间片 t_{slot} 设定为:

$$t_{slot} = \frac{R-d}{V_{max}} \quad (4)$$

其中, R 为 AP 的有效覆盖半径, d 为距离阈值, V_{max} 为人步行的最大速度。

假设 1 除服务区边缘外,每一个 AP 有效覆盖范围之外的区域均被其邻居 AP 完全覆盖。

定义 1(K 阶马尔科夫模型) 它由状态集合 S^K 与转移矩阵 T^K 定义。 K 阶马尔科夫模型将用户过去以及当前所连的 K 个 AP 视为一个马尔科夫状态,如图 4 所示,用户 U_m 的

AP 踪迹为(A,B,C),并且正与 C 连接,则 U_m 的 1 到 3 阶马尔科夫模型的当前状态分别为 C、BC 和 ABC。 T^K 是一个稀疏矩阵, U_m 下一个时间片内只会转向邻近状态(ABC 的邻近状态是 BCB、BCD、BCE 和 BCF)和当前状态,转向其他状态的概率为 0。

定义 2(转移概率) 用户 U_m 在 n 时刻的状态为 W ,则其 $n+1$ 时刻的状态集合为 $\{W_{N1}, W_{N2}, \dots, W_{Nk}, W\}$,如图 5 所示,对于任意 $W \in S^K$,称 $P\{X_{n+1}=W_{Ni} | X_n=W\}=p$ 为 U_m 在 n 时刻的一步转移概率,且 p 与 n 无关。本文通过学习过去 T 天用户的历史踪迹数据得出一步转移概率:

$$P\{X_{n+1}=W_{Ni} | X_n=W\} = \frac{\text{从 } W \text{ 到 } W_{Ni} \text{ 的人数}}{\text{进入 } W \text{ 的总人数}} \quad (5)$$

并且可得

$$P\{X_{n+1}=W | X_n=W\} = 1 - \sum_{i=1}^k P\{X_{n+1}=W_{Ni} | X_n=W\} \quad (6)$$

注意:式(5)中进入 W 的总人数包括其邻近状态转入的人数和从 W 开始发起新的通话的人数,这是为了覆盖那些从通话开始到结束未切换 AP 的用户。

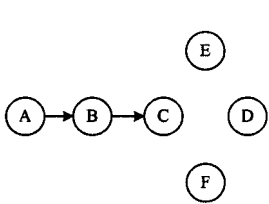


图4 位置与状态图

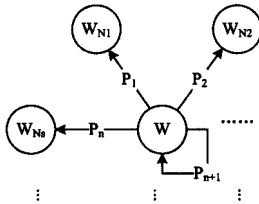


图5 状态转移图

假设用户 U_m 已有的 AP 踪迹长度为 L , U_m 的邻近微云 C_i 发出请求调度,则此时可根据定义 1 和定义 2 得出 U_m 下一个时刻迁入 C_i 的 L 到 1 阶马尔科夫一步转移概率 $(P_m^L, \dots, P_m^1)$ 。LPMPPS 算法先计算预取队列中各个用户的 $(P_m^L, \dots, P_m^1)$,然后从最高阶 L 逐阶对比概率阈值 ρ ,若有 P_m^i 低于 ρ ,则停止对比并踢出该用户的预取进程。考虑到预取队列中所有用户的 $(P^L, \dots, P^1)$ 均不小于 ρ 的极端情况,采用先进先出调度策略(FIFO)踢出预取队列中队首进程,因为相当长时间的预取,用户依然没有转入 C_i ,所以有理由认为下一个时隙内用户依然不会迁入。因为用户在他所有的邻近微云上的预取队列几乎都是在同一时刻建立的,可能会出现某个用户所有预取队列都被 FIFO 踢出的情况,所以当微云上的预取进程 PL_i 是被先进先出踢出时,则其对应的用户在其他微云上的预取进程的队列位置(若已在队首)将后移一位。

算法 1 低概率预取进程调度(LPMPPS)

1. Input: PL , TalkingTable, TraceTable
// PL 是微云 C_i 上的预取进程队列
// TalkingTable 是正在服务用户表(同算法 2,3)
// TraceTable 是历史用户踪迹表(同算法 2,3)
2. Output: List<Progresses> // 调度决策: 需要被调出 PL 的进程
3. List<Progresses> $\leftarrow$ null;
4. for $i \leftarrow 1, i \leq PL.length$ && C_i 's BandwidthLoad $\geq \mu_b, i++$ do
5. User $\leftarrow$ Dequeue(PL); // 取队首进程对应的用户 U_m
6. Trace $\leftarrow$ GetTrace(User, TalkingTable); // 取 U_m 的 AP 踪迹
7. $K \leftarrow$ Trace.length;
8. while $K \geq 1$ do
9. $P_K \leftarrow K_Possible(\text{Trace}, \text{TraceTable})$;
// 计算 U_m 下一 t_{slot} 转入 C_i 的 K 阶转移概率 P_K

10. if $P_K > 0$ && $P_K < \rho$ then
11. List<Progresses> $\leftarrow$ add(User, List<Progresses>);
// 将低于概率阈值 ρ 的预取进程加入调度决策列表
12. endif
13. $K \leftarrow K-1$;
14. endwhile
15. endfor
16. while C_i 's BandwidthLoad $\geq \mu_b$ do
17. List<Progresses> $\leftarrow$ FIFO(PL); // 按 FIFO 将队首进程加入调度决策
18. endwhile
19. return List<Progresses>

算法 1 的第 8 到 14 步计算出用户 U_m 的 $(P_m^L, \dots, P_m^1)$,并逐阶与概率阈值 ρ 对比,同时过滤掉转移概率为 0 的情况,这是因为转移概率主要是学习历史用户踪迹得来的。转移概率为 0 有可能是因为 U_m 的 AP 踪迹过长,导致选取的马尔科夫模型阶数过高,而 TraceTable 中没有这么长的历史 AP 踪迹。所以转移概率为 0 并不代表 U_m 的转移概率极低,相反表明这个转移概率是不可信任的,此时应当降低马尔科夫模型阶数再进行对比。第 17 步是在多阶转移概率调度失败之后按照改进的先进先出踢出预取队列中的进程。

4.2 AP 选择器

AP 选择器的目标有两个:1)为可能切换 AP 的用户(与其所连 AP 的距离超过 d)在其所有的邻近微云上建立预取进程;2)为需要启动通话服务的用户选择首个 AP,这主要是为了平衡服务区中微云的计算负载和带宽负载,相比于进程调度,这是一个轻量级的工作,所以响应速度更快。实现这两个目标的具体算法如算法 2、算法 3 所示。

算法 2 贪心预取

1. Input: Map<AP>, TalkingTable
2. Output: Null
3. UserQueue.add(RequestListener()); // 监听预取请求并排队
4. while UserQueue $\neq$ null do
5. Cloudlet $\leftarrow$ Dequeue(UserQueue);
6. List<Neighbor> $\leftarrow$ gen(Cloudlet, Map<AP>);
// 查找用户邻近的 Cloudlets
7. for all elements in List<Neighbor> do
8. Addr $\leftarrow$ Address(Cloudlet, TalkingTable);
9. SendTCPRequest(Addr, element);
10. endfor
11. endwhile

算法 2 的输入包括 AP 分布图 Map<AP>(也是微云分布图)和 TalkingTable 列表。算法响应预取请求,找出用户所有邻近微云,并命令这些微云为用户建立预取进程。第 5 步是获取发出队首请求的微云,而第 6 到 10 步则是通过查找 AP 分布图找出用户所有邻近微云,并向这些微云发出用户通话的目的地址,命令它们为用户建立预取进程。

算法 3 起始 AP 选择

1. Input: List<AP>, List<Load>, μ_c, μ_b
2. Output: AP or null
3. for all AP in List<AP> do
4. AP.ComputLoad $\leftarrow$ FindComputLoad(List<Load>);
5. AP.BandwidthLoad $\leftarrow$ FindBandwidthLoad(List<Load>);
// 查找 AP 对应 Cloudlet 的负载情况
6. end for
7. for $i \leftarrow 1, i \leq \text{List<AP>.length}, i++$ do

```

8. if AP[i].ComputLoad> $\mu_c$  then
9.     continue;
10. endif
11. if AP[i].BandwidthLoad< $\mu_b$  then
12.     return AP[i];
13. endif
14. endfor
15. return null;

```

算法3的输入包括移动终端所监测到的AP列表List(AP)(按信号强度降序排列)和微云状态表List(Load),后者包含微云的计算负载 μ_c 和带宽阈值 μ_b 。算法3的输出则是一个满足条件的AP或者空值(没找到符合条件的AP)。第8行排除对应微云的计算负载超过阈值 μ_c 的AP,这主要是考虑到微云上的预取进程只有极小的计算开销,而主要的计算开销来自通话虚拟机,因此为了保障已建立虚拟机的用户的通话服务,此AP需要拒绝新用户的接入,直到对应微云的负载低于阈值。

5 仿真实验与性能分析

5.1 实验设计

本节通过仿真实验对CMVP及其相应算法对视频预取命中率的提高效果进行评估与分析,同时分析命中率提高对用户QoE的影响。为便于表示和实验,实验将前面提到的AP和微云的功能全都定义在AP对象中。

- 命中率:服务区中所有节点预取失败总次数与服务区中所有节点的AP切换总次数的比值。预取失败指的是移动节点切换到新的AP时,此AP的预取队列中没有它的记录。

- 仿真服务区:一个4×4的AP覆盖区域,每个AP的坐标固定并且已知,AP的属性参数统一如表3所列。为了简化实验,限制预取队列长度为20来表示微云带宽负载阈值 μ_b 。服务区AP覆盖率并非100%,但是并不影响本实验对命中率的计算。

表3 AP属性参数

AP数/个	半径R/m	距离阈值d/m	预取队列长度
16	100	80	20

移动节点模型:实验中的移动点采用随机路点模型来模拟服务区中用户的移动,移动节点属性参数如表4所列。移动点的初始点均匀分布在各个AP的坐标点上,各个移动点将初始点对应AP作为自己的首个AP,每个移动节点随机选择服务区内的一个目的点,并以速度 v 向目的点移动, v 在 $[V_{min}, V_{max}]$ 上均匀产生。移动节点到达目的点后,随机选择下一个目的点,重复该过程直到其完成生命周期。节点脱离已连AP的覆盖后将自动连接离自己最近且与自己距离小于 R 的AP。

表4 移动节点属性参数

速度下限 $V_{min}/m/s$	速度上限 $V_{max}/m/s$	初始坐标	生命周期上限
0.5	1.2	AP _i	10

- 生命周期:通话服务结束或者终止。当移动节点连续连接10个AP,或者脱离AP覆盖区,或者到达服务区边界时,就认为该移动点完成一次通话,或者由于脱离服务区而终止通话。

- 学习数据:过去 T 天历史用户的AP踪迹数据。本实验设定历史踪迹为20000条,由于随机路点移动模型各个节点的移动是相互独立的,因此可以简化为采用单个移动点在服务区中完成20000次生命周期来收集学习数据。

- QoE参量:因本文工作旨在提升预取命中率来减少通话重连次数,所以式(3)中的音视频同步 t_{sync} 和视频观看质量 V_q 均参照文献[10]设定成50ms和10,而 $T_{Handover}$ 根据ITU-TG.114推荐给VoIP类型业务的建议设定为100ms。

- 影响因子 K_1 :设定值为3。

实验移动节点设定为50到500个,每一组参数的值固定后,进行10次实验取命中率平均值。

5.2 结果分析

针对设置不同阶数上限 K 、节点数 Num 和低概率阈值 ρ 进行了多组实验。首先验证执行LPMPPS算法的CMVP是否能够获得令人满意的预取命中率;然后与采用FIFO算法的CMVP进行对比,接着分析 K 和 ρ 对采用LPMPPS算法的CMVP的预取命中率的影响并得出改进方向;最后,分析命中率和以太网链路延迟 T_L 对用户QoE的影响,评估LPMPPS对QoE的积极作用。

图6表明LPMPPS($K>2$)算法在 Num 低于250时可以获得90%甚至95%以上的命中率,这说明采用LPMPPS算法的调度策略不会对贪心预取算法的命中率造成太大影响。虽然FIFO算法计算复杂度低,但是随着 Num 增加,CMVP的预取命中率急速下降。而LPMPPS在 K 取1时,由于随机路点模型近乎均匀分布的特性使得节点在每个状态的一阶转移概率大体相同,极少有低于0.1的转移概率,此时的命中率与FIFO的命中率几乎重合。图6还表明提高 K 可以提高CMVP的命中率,这是因为 K 的提高意味着更多的过去的AP踪迹得到匹配,高阶状态转移概率更加可靠。另外随着 Num 增加,预取进程调度速度越来越快,很多节点的多阶转移概率虽然高于 ρ ,但是因为已经处于预取队列队首而被踢出,导致了预取命中率下降,此时需要增加预取队列长度,即增加服务区微云带宽。虽然 K 值增加可以提高命中率,但当 $K>4$ 后命中率提高幅度有限,这是因为切换5个AP以上的移动节点数量明显减少,所以对命中率的提升有限,而且实际情况下视频通话用户切换4个以上AP的可能性很小,毕竟每个公共AP的有效覆盖半径都约有100m。

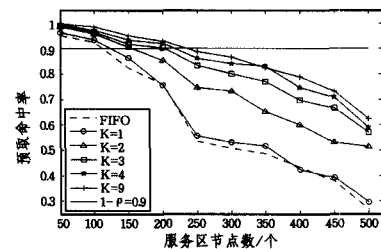


图6 LPMPPS(不同 K)与FIFO的命中率对比

图7中 K 取固定值3,改变低概率阈值 ρ ,可以看出 ρ 值的变动对采用LPMPPS算法的CMVP的命中率影响很大。很容易理解的是:随着 ρ 值增大,预取队列中满足调度条件(转移概率小于 ρ)的进程数目增多,那么触发先进先出调度的可能性也在降低,命中率曲线下降速度减缓。但是 ρ 越大,意味着被淘汰进程对应的节点下一时刻迁入AP的转移概率

越大,导致在 Num 较低时命中率也不高。从图 7 可以看出,降低 ρ 值并不能提高命中率,因为 ρ 越低,满足调度条件的进程数目越少,先进先出调度的次数越多,甚至 ρ 取 0.05 的命中率曲线几乎与 FIFO 曲线重合。图 7 给出了一点启示:当服务区人数增多时,从节省成本的角度考虑,可以适当降低 ρ 值来平缓命中率曲线的下降趋势。

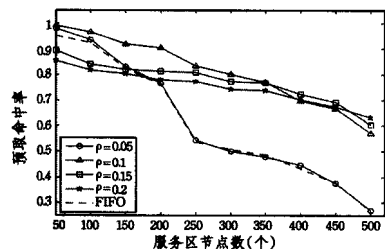


图 7 ρ 值、 Num 和命中率的关系

由仿真结果可知, n_w 的平均值为 3.85, 故根据式(3)推出在命中率为 100% 时 QoE 的上限为 2.22, 命中率为 0 时 QoE 的下限为 1.07。图 8 的曲线 $K=3$ 表明节点数 Num 小于 200 时, LPMPPS 能够保证令人满意的用户 QoE (>2), 同时也从用户体验角度上说明 LPMPPS 算法明显优于 FIFO 算法。图 9 表明网络链路延迟越大, 拥有很好的预取命中率 (90% 以上) 的 CMVP 系统对视频通话服务的意义越发重要, 即便在 $T_L=40\text{ms}$ 时, 用户 QoE 依然可以保持在 2 以上。

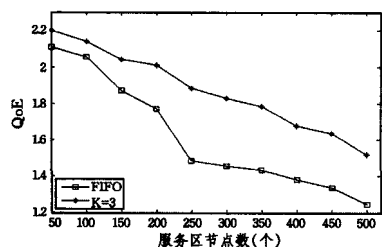


图 8 LPMPPS($K=3$)与 FIFO 的 QoE 对比($T_L=10\text{ms}$ [10])

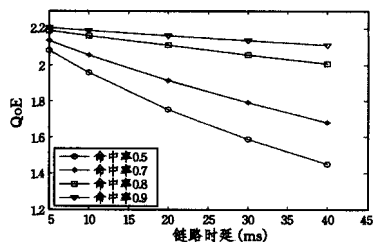


图 9 QoE 随命中率和链路时延 T_L 的变化情况

结束语 本文提出了基于微云的移动视频预取系统 CMVP, 它通过本地微云为移动终端代理视频通话, 从而能够实现本地的视频流预取工作, 并设计了贪心预取算法来获取令人满意的预取命中率, 从而减少用户多次 AP 切换情况下的视频通话重连次数, 显著地提升了用户 QoE。另外, CMVP 设计 LPMPPS 算法来缓解贪心预取算法带来的本地微云带宽不足的情况, 同时设计起始 AP 选择算法来平衡本地微云的计算和带宽负载, 而这些计算开销很大一部分的工作均由云层负责, 数据收集则由本地微云负责, 从而使终端避免不必要的额外工作。

- [1] Yeh C H, Jiang S J F, Lin C Y, et al. Temporal video transcoding based on frame complexity analysis for mobile video communication[J]. IEEE Transactions on Broadcasting, 2013, 59(1): 38-46
- [2] Wang Meng-si, Song Xiao, Lei Quan, et al. A power-efficient cloud-based compressive sensing video communication system [C]//Proceedings of Signal and Information Processing Association Annual Summit and Conference (APSIPA), Asia-Pacific, 2013: 1-4
- [3] Qadeer M A, Shah K, Goel U. Voice-Video Communication on Mobile Phones and PCs' Using Asterisk EPBX [C]//Proceedings of Communication Systems and Network Technologies (CSNT), Rajkot, 2012: 534-538
- [4] 移动视频通信应用下载[OL]. http://shouji.baidu.com/s?wd=%D2%C6%B6%AF%CA%D3%C6%B5%CD%A8%BB%B0&data_type=app&from=web_alad_5
- [5] Ramzan N, Park H, Izquierdo E. Video streaming over P2P networks: Challenges and opportunities [J]. Signal Processing: Image Communication, 2012, 27(5): 401-411
- [6] Miao Dan, Zhu Wen-wu, Luo Chong, et al. Resource allocation for cloud-based free viewpoint video rendering for mobile phones [C]//Proceedings of the 19th ACM international conference on Multimedia. Scottsdale, 2011: 1237-1240
- [7] Li Ji-wei, Bu Kai, Liu Xuan, et al. ENDA: embracing network inconsistency for dynamic application offloading in mobile cloud computing [C]//Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing. HongKong, 2013: 39-44
- [8] Xiong Hao-yi, Zhang Da-qing, Gauthier V, et al. Predicting mobile phone user locations by exploiting collective behavioral patterns [C]//Proceedings of Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC). Fukuoka, 2012: 164-171
- [9] Papandrea M, Giordano S. Location prediction and mobility modelling for enhanced localization solution [J]. Journal of Ambient Intelligence and Humanized Computing, 2014, 5(3): 279-295
- [10] Garcia M, Canovas A, Edo M, et al. A QoE management system for ubiquitous IPTV devices [C]//Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM). Sliema, 2009: 147-152
- [11] Lin Chuang, Hu Jie, Kong Xiang-zhen. Survey on Models and Evaluation of Quality of Experience [J]. Chinese Journal of Computer, 2012, 35(1): 1-15 (in Chinese)
- [12] 林闯, 胡杰, 孔祥震. 用户体验质量 (QoE) 的模型与评价方法综述 [J]. 计算机学报, 2012, 35(1): 1-15
- [13] Qi Xin, Yang Qing, Nguyen D T, et al. Context-aware frame rate adaption for video chat on smartphones [C]//Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication. Zurich, 2013: 111-114
- [13] Liu Yun-hao, Yang Zheng, Wang Xiao-ping, et al. Location, localization, and localizability [J]. Journal of Computer Science and Technology, 2010, 25(2): 274-297