



研究与开发

基于 A3C 的无线异构网络自适应视频流传输控制方法

罗志强, 王伟, 朱晓荣

(南京邮电大学通信与信息工程学院, 江苏 南京 210003)

摘要: 比特率自适应 (ABR) 算法已经成为视频传输中研究的热点之一。然而, 由于 5G 无线异构网络具有信道带宽波动大、不同网络间差异明显等特点, 多终端协同的自适应视频流传输面临着巨大挑战。提出了一种基于深度强化学习的自适应视频流传输控制方法。首先, 建立了视频流动态规划模型, 对传输码率以及分流策略进行联合优化。由于该优化问题的求解依赖于精确的信道估计, 这在信道状态动态变化的网络中很难实现。因此, 将动态规划问题改进为强化学习任务, 并采用 A3C 算法, 动态决策视频码率和分流策略。最后, 根据实测的网络数据进行仿真, 与传统的优化方法相比, 本文所提的方法较好地提高了用户 QoE。

关键词: 无线异构网络; A3C; 码率自适应; 多终端协同; QoE

中图分类号: TP3

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2020308

An adaptive video stream transmission control method for wireless heterogeneous networks based on A3C

LUO Zhiqiang, WANG Wei, ZHU Xiaorong

School of Communication and Information Engineering, Nanjing University of Posts
and Telecommunications, Nanjing 210003, China

Abstract: The adaptive bit rate (ABR) algorithm has become the focus research in video transmission. However, due to the characteristics of 5G wireless heterogeneous networks, such as large fluctuation of channel bandwidth and obvious differences between different networks, the adaptive video stream transmission with multi-terminal cooperation was faced with great challenges. An adaptive video stream transmission control method based on deep reinforcement learning was proposed. First of all, a video stream dynamic programming model was established to jointly optimize the transmission rate and diversion strategy. Since the solution of this optimization problem depended on accurate channel estimation, dynamically changing channel state was difficult to achieve. Therefore, the dynamic programming problem was improved to reinforcement learning task, and the A3C algorithm was used to dynamically determine the video bit rate and diversion strategy. Finally, the simulation was carried out according to the measured network data, and compared with the traditional optimization method, the method proposed better improved the user QoE.

Key words: wireless heterogeneous network, A3C, rate adaptation, multi-terminal cooperation, QoE



1 引言

随着无线通信技术的快速发展,用户对于无线视频业务的需求急剧增长,并且无线视频业务将主导无线数据业务市场^[1],传统的单链路视频传输方式已经无法满足用户的需求。因此,多流并发技术^[2]应运而生。在多流并发中,视频流通过多个异构终端协同传输,每个异构终端具有不同的网络制式,如 5G、TD-LTE、FDD-LTE 等,最后在客户端聚合视频流,可以提高视频传输速率。与此同时,为了进一步提升用户 QoE,ABR 算法^[3]成为当前研究的热点之一。在视频服务器端,视频流被分割为具有相同时间长度的视频块,每个视频块被编码为不同的码率,视频码率越高,意味着客户端接收视频的质量越好。在服务器端运行 ABR 算法,服务器端基于当前网络带宽以及客户端缓冲区占用状况,动态地为每一个视频块选择最优的码率,避免客户端视频播放中断现象的发生。多流并发技术和比特率自适应算法为用户提供无缝的视频流服务创造了条件。

由于 5G 无线异构网络的信道带宽波动大^[4],网络的吞吐量处于不断变化中,很难对信道带宽做出准确的预测。同时,不同制式的网络之间差异明显,使得各异构终端传输能力存在很大差异。因此,多终端协同的自适应视频流传输面临者巨大的挑战。一方面,在复杂的无线信道中,需要根据播放端的缓冲区占用状况,动态地决策出每个视频块的码率,以最大化用户 QoE。但 QoE 指标存在两大矛盾关系,其一是视频质量和视频中断次数之间存在矛盾关系^[5],即当传输的视频质量高时,客户端的缓冲区容易耗尽,导致视频播放出现中断;其二是带宽利用率与视频质量平滑度之间的矛盾关系,即当传输的码率尽可能匹配当前带宽时,码率容易出现频繁切换,导致视频码率的平滑度降低。因此,良好的 ABR 算法不仅需要提供尽可能高的视频质量,同时也要减少视

频中断出现的次数,还应该尽量保持视频质量的平滑度。另一方面,在异构网络中,需要根据各异构终端的传输能力,制订合理的分流传输策略,为异构终端分配合理的视频流,以最大限度地利用网络资源,如果资源分配不合理,不仅不能提高传输速率,还会导致更大的时延。综上所述,在多终端协同传输的背景下,需要同时考虑这两个方面,对视频流的传输码率以及各异构终端的分流策略进行联合优化,这使得决策过程变得更加复杂。

目前,已经有大量的文献对自适应视频流传输控制方法进行了研究。参考文献[6]针对无线异构网络下的多流并发场景,对每条并行支路应用连续的 M/M/1 排队论模型,以最小化各支路时延差为目标,利用拉格朗日乘子对优化问题求解,实现流量的自适应分配。参考文献[7]采用基于信道带宽的 ABR 算法,通过实时估计网络带宽,选择与带宽相匹配的码率,在网络的稳定性、公平性和效率之间进行权衡。参考文献[8]提出了基于缓冲区的 ABR 算法。由于无线信道的带宽变化剧烈,无法实时估计网络带宽。该方法选择码率时,只需要获取缓冲区的信息,而仅在视频启动阶段进行简单的信道带宽估计,进而提升了算法的准确性。参考文献[9]认为单一的依据信道带宽信息或者缓冲区信息不能决定码率的选择,而是二者信息的融合。将码率的选择转化为随机动态规划问题,并提出了模型预测控制(model predictive control, MPC)算法,通过预测未来一定时间段的网络带宽并结合缓冲区的占用量,解决该优化问题。参考文献[10]研究了多个服务器的场景,将各个服务器提供的视频分片组合成视频块,再进行传输。提出了一种比例微分控制方法来选择视频码率,具有较高的响应性和稳定性。参考文献[11]研究了多用户的资源分配问题,将该问题转化为马尔可夫决策过程,并使用深度强化学习的方法求解。但却仅使用了缓冲区的信息,不能很好利

用带宽信息。参考文献[12]将深度强化学习的方法用到了单链路的 ABR 算法中,提出的 Pensieve 系统能够自适应地选择视频码率,而不需要精确的信道估计。

上述文献主要针对单链路视频传输场景,对于多流并发场景研究较少。另外,大多数文献都把 ABR 算法的生成过程当作传统优化问题进行处理,通过在特定场景下建模求解。然而,这些文献所提出的 ABR 算法基于大量的假设,算法的应用场景很局限,并且不可避免地需要预测信道的带宽。没有依据不同应用场景提出合适的视频传输控制方法。

针对上述问题,本文将深度强化学习方法和视频传输控制方法相结合。以多终端协同传输为背景,首先建立了动态规划模型,以最大化用户 QoE 为目标,对传输码率以及分流策略进行联合优化。由于该优化问题的求解依赖于精确的信道估计,该问题适合利用强化学习的方法对信道环境进行探索和学习,将动态规划问题改进为强化学习任务,并在 Pensieve^[12]系统的基础上,利用 A3C (asynchronous advantage actor-critic) 算法对该任务进行合理决策,避免信道带宽预测带来的误差,从而很好地提升了终端用户的 QoE。

2 系统模型

本文的应用场景如图 1 所示,各异构终端和视频客户端通过 Wi-Fi 局域网组成虚拟大终端,由视频客户端向视频服务器发起视频请求。视频服务器根据各异构终端的网络状况以及视频客户端的缓冲区占用率,运行 ABR 算法和分流策略,确定视频块的码率,并将选定的视频块通过 5G、TD-LTE、FDD-LTE 等异构网络分流传送到各异构终端。最后通过 Wi-Fi 局域网传输到视频客户端聚合并播放。各异构终端协同传输,提高了视频传输速率,能更好地满足用户需求。

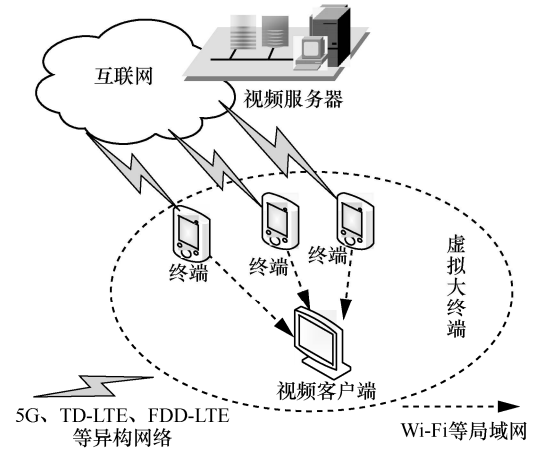


图 1 多终端视频流协同传输场景

(1) 视频服务器

多终端视频流协同传输系统架构如图 2 所示。视频流在视频服务器被分割为 N 个连续的视频块,每个视频块的播放时间长度均为 T_c 秒,并将视频块按照播放顺序标号为 $n = \{1, 2, \dots, N\}$ 。每个视频块被编码为不同的码率,用 R_{all} 表示视频块可选码率的集合,则选择的第 n 个视频块的码率可以表示为 $R_n \in R_{all}$ 。

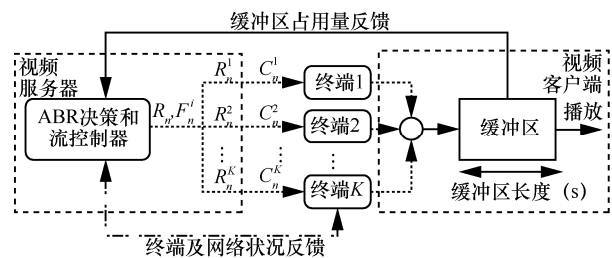


图 2 多终端视频流协同传输系统架构

每个视频块被分割为 F 个视频分片,每个分片的长度均为 T_f 秒,因此,视频块和分片之间满足: $T_c = FT_f$ 。选定好视频块的码率 R_n 后,根据分流策略,将选定的视频块切割成分片,并分流到 K 个终端,将终端编号为 $i = \{1, 2, \dots, K\}$,则第 i 个终端分到的分片数量可表示为 $F_n^i \in \{0, 1, \dots, F\}$,分到的数据量可以表示为 R_n^i ,其中 R_n 与 R_n^i 有如下关系:

$$R_n = \frac{1}{T_c} \sum_{i=1}^K R_n^i \quad (1)$$



其中, R_n^i 由分得的分片数量决定, 因此有

$$R_n^i = \frac{F_n^i}{F} T_c R_n^i.$$

(2) 协同终端和信道

视频分流后, 在各自终端所分配的信道中传输。 C_t 表示 t 时刻的信道带宽, C_t^i 表示 t 时刻第 i 个终端的信道带宽。 C_n^i 表示第 n 个视频块传输时, 第 i 个终端的平均带宽。其中, C_t^i 与 C_n^i 有如下关系:

$$C_n^i = \frac{\int_{t_n^s}^{t_n^e} C_t^i}{t_n^e - t_n^s} \quad (2)$$

其中, t_n^s 、 t_n^e 分别表示第 n 个视频块开始下载的时间和完成下载的时间, 则 $t_n^e - t_n^s$ 表示第 n 个视频块下载所需的时间。只有整个视频块的所有分片都下载完成后, 视频流才能在视频客户端聚合, 于是下载第 n 个视频块所需的时间为:

$$t_n^e - t_n^s = \max_{i \in \{1, 2, \dots, K\}} \left\{ \frac{R_n^i}{C_n^i} \right\} \quad (3)$$

其中, $\frac{R_n^i}{C_n^i}$ 表示第 i 条并行支路的下载时间。将 K 条并行支路中的最长下载时间, 作为当前视频块的下载时间。

(3) 播放缓冲区

在客户端聚合后的视频块, 将被装入播放缓冲区等待播放。缓冲区以秒作为衡量单位, 每秒播放 1 s 的视频块, 并移出播放缓冲区。缓冲区的容量设为 B_c 秒, $B_t \in [0, B_c]$ 表示 t 时刻缓冲区的占用量。用 B_n 表示第 n 个视频块开始下载时播放缓冲区的占用量。一旦视频块下载完毕, 就会有 T_c 秒的视频装入缓冲区, 并且缓冲区会有 $t_n^e - t_n^s$ 秒的视频被播放。如果在一个视频块下载完成之前, 缓冲区被耗尽 (即 $B_t = 0$), 那么视频播放就会出现中断, 因此 ABR 算法要避免耗尽缓冲区。如果信道带宽特别大, 而选择的码率却很小, 那么缓冲区就有可能被填满 (即 $B_t = B_c$), 触发慢启动协议^[13], 等待 T_s 秒后, 再请求下一个视频块, 于是从第 n 个视频块下载结束到第 $n+1$ 视频视频块开始下载的

时间可表示为:

$$t_{n+1}^s - t_n^e = \begin{cases} 0, & B_{t_n^e} < B_c \\ T_s, & \text{其他} \end{cases} \quad (4)$$

其中, $B_{t_n^e}$ 表示第 n 个视频块完成下载时, 缓冲区的占用量。因此, 第 $n+1$ 个视频块开始下载时, 缓冲区的占用量可表示为:

$$B_{n+1} = \max \left(T_c + \max \left(B_n - (t_n^e - t_n^s), 0 \right), B_c \right) - (t_{n+1}^s - t_n^e) \quad (5)$$

由于视频块必须下载完成后, 才能装入缓冲区。因此, 在视频块还未下载完成时, 缓冲区的占用量为 $\max(B_n - (t_n^e - t_n^s), 0)$, 其中, 当 $B_n - (t_n^e - t_n^s) < 0$ 时, 缓冲区向下溢出, 此时缓冲区占用量为 0, 当视频块下载完成, 第 $n+1$ 个视频块开始下载时, $B_{n+1} = T_c$ 。当视频块下载完成, $T_c + \max(B_n - (t_n^e - t_n^s), 0) > B_c$ 时, 缓冲区向上溢出, 即此时系统会触发慢启动机制, 系统会暂停 $t_{n+1}^s - t_n^e = T_s$ s, 再请求下一个视频块, 此时 $B_{n+1} = B_c - T_s$ 。

3 视频流传输控制模型

本文首先将基于多流并发的自适应视频流传输控制问题转化为动态规划问题, 在单链路方法 MPC^[9]的基础上, 建立动态规划模型, 分析模型求解的难点, 以及利用传统优化算法求解的缺点。同时, 介绍强化学习的方法在处理这类问题时的优点, 将动态规划问题转化为强化学习任务, 形成最终的视频流传输控制模型。

3.1 视频流动态规划模型

无论是多流并发技术还是 ABR 算法, 最终的目的都是为了提高用户的 QoE。本文用户 QoE 的衡量指标被分为视频感知质量、播放中断程度和视频质量平滑度, 定义为:

$$\text{QoE} = \mu \sum_{n=1}^N m(R_n) - \beta \sum_{n=1}^N f(T_n) - \lambda \sum_{n=1}^{N-1} |m(R_{n+1}) - m(R_n)| \quad (6)$$

其中, R_n 表示第 n 个视频块的码率, $m(\cdot)$ 表示映射函数, 将视频块的码率映射到用户感知的视频质量, 则 $m(R_n)$ 表示第 n 个视频块的用户感知质量。 T_n 表示第 n 个视频块播放时, 出现中断的时间, $f(\cdot)$ 是中断时间的映射函数, 以惩罚因为选择的码率过高而带来的缓冲事件。前后两个视频块的视频感知质量的差值 $|m(R_{n+1}) - m(R_n)|$ 表示视频质量的平滑度。 μ 、 β 、 $\lambda \in (0,1)$ 分别表示各项指标的权重系数, 以衡量各项指标的重要程度。

(1) 视频感知质量 $m(R_n)$

映射函数 $m(\cdot)$ 有多种形式, 考虑到随着码率的增大, 用户对于码率的区分度会下降。因此, 本文选择对数形式, 并将其归一化, 表示为:

$$m(R_n) = \log \frac{(R_n)}{(R_{\min})} / \log \frac{(R_{\max})}{(R_{\min})} \quad (7)$$

其中, $R_n/R_{\min}$ 是为了确保映射函数始终大于或等于 0, 通过归一化后, $m(R_n) \in [0,1]$ 。

(2) 播放中断程度 $f(T_n)$

当视频播放缓冲区耗尽时, 视频块还未下载完毕, 也就是 $(t_n^e - t_n^s) > B_n$ 时, 视频播放就会出现中断。因此, 播放中断时间 T_n 定义为:

$$T_n = \begin{cases} 0, (t_n^e - t_n^s) \leq B_n \\ (t_n^e - t_n^s) - B_n, \text{其他} \end{cases} \quad (8)$$

考虑到中断时间达到一定程度之后对用户的影响会逐渐趋于稳定, 并且为了将其归一化, 本文选择改进后的对数几率函数:

$$f(T_n) = \frac{2}{1 + e^{-T_n}} - 1 \quad (9)$$

其中, $T_n \geq 0$, 并且 $f(T_n) \in [0,1]$ 。

综上所述, 视频流动态规划模型为:

$$\begin{aligned} \max_{F_n^i, R_n} \text{QoE} = & \mu \sum_{n=1}^N m(R_n) - \beta \sum_{n=1}^N f(T_n) - \\ & \lambda \sum_{n=1}^{N-1} |m(R_{n+1}) - m(R_n)| \end{aligned} \quad (10)$$

$$\begin{aligned} & \begin{cases} R_n \in R_{\text{all}} \\ R_n = \frac{1}{T_c} \sum_{i=1}^K R_n^i \\ R_n^i = \frac{F_n^i}{F} T_c R_n, F_n^i \in \{0, 1, \dots, F\} \\ C_n^i = \int_{t_n^s}^{t_n^e} C_t^i / t_n^e - t_n^s \\ \text{s.t. } \begin{cases} t_n^e - t_n^s = \max_{i \in \{1, 2, \dots, K\}} \left\{ \frac{R_n^i}{C_n^i} \right\} \\ t_{n+1}^s - t_n^e = \begin{cases} 0, B_{t_n^e} < B_c \\ T_s, \text{其他} \end{cases} \\ B_{n+1} = \max \left(T_c + \max \left(B_n - (t_n^e - t_n^s), 0 \right), B_c \right) - \\ \quad (t_{n+1}^s - t_n^e) \\ B_1 = T_{\text{start}} \\ n = \{1, 2, \dots, N\}, i = \{1, 2, \dots, K\} \end{cases} \end{cases} \end{aligned} \quad (11)$$

其中, T_{start} 表示视频启动时的缓冲区初始值, F_n^i, R_n 是决策变量, 需要决策每一个视频块的码率 R_n 以及分流到各个终端的分片数量 F_n^i 。该动态规划问题的本质是根据当前缓冲区的占用量 B_n 、未来信道带宽的估计 C_n^i 、过去比特率和分流策略的选择进行决策, 可表示为:

$$\{F_n^i, R_n\} = p \left(B_n, C_n^i, \{F_1^i, \dots, F_{n-1}^i\}, \{R_1, \dots, R_{n-1}\} \right) \quad (12)$$

其中, 未来带宽的估计 C_n^i 是解决该动态规划问题的难点。能够获取的信道带宽只能是过去的, 因此只能通过过去的信道带宽来估计未来的信道带宽。然而, 在无线异构网络中, 信道带宽变化较快, 想要预测出准确的信道带宽是十分困难的。传统的优化方法对于信道带宽的预测过于保守, 没有充分利用网络的资源。而免模型的强化学习任务能够对环境进行探索, 根据过去的网络信息选择动作, 并基于环境反馈的奖励值来调整策略, 而不涉及对未来信道带宽的预测。通过“试错”的方式来学习最优的策略, 进而选择在该网络状况下, 所获奖励值最大的动作, 而不是事先对环境进行假设和建模, 具有很强的迁移和应用能力。



因此, 本文选择使用强化学习的方法来解决自适应视频流传输问题。

3.2 基于强化学习的视频流任务

传统的动态规划模型在信道带宽急剧变化的情况下已不再适用, 因此本节将探讨一种基于强化学习的视频流任务。

基于强化学习的视频流任务如图 3 所示, 视频流模型被改进为基于强化学习的视频流任务。在强化学习任务中, 视频客户端、协同终端以及无线信道被视为视频传输和播放的环境; 视频服务器中的 ABR 决策和分流控制器相当于决策代理。首先, 位于视频服务器中的决策代理根据当前的网络状态 s_n , 运行基于 A3C 的视频流控制算法, 算法会根据神经网络的输出选择一个动作 a_n , 执行动作 a_n 后, 基于状态转移概率, 环境会发生变化, 视频块在经过下载和播放后, 网络状态由 s_n 变为 s_{n+1} 。同时, 环境会基于定义的奖励函数产生一个奖励值 r_n , 以衡量执行动作 a_n 的好坏。 s_{n+1} 和 r_n 被一同反馈到决策代理, 以进行下一次的决策。

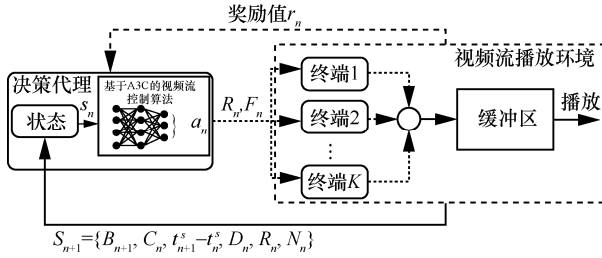


图3 基于强化学习的视频流任务

(1) 状态集

网络环境的状态是决策的依据, 由式 (11) 可知, 为了决策出 $\{F_n^i, R_n\}$, 就需要获取网络环境的状态信息, Pensieve^[12] 已对单链路的场景提出了可行的方案, 但对于多流并发场景并不适用, 本文在此基础上设定状态集 S 为:

$$S = \{B_n, C_{n-1}, t_n^s - t_{n-1}^s, D_{n-1}, R_{n-1}, N_{n-1}\} \quad (13)$$

其中, 第一项是当前缓冲区占用量。第二项 $C_{n-1} = \{C_{n-1}^1, C_{n-1}^2, \dots, C_{n-1}^K\}$ 表示下载上一个视频块

时, K 条并行支路的平均带宽, 以衡量各条支路上一次视频下载的能力, 而不是估计未来的信道带宽 C_n^i 。第三项表示前 l 个视频块的下载时间 (包括缓冲区上溢的时间) 的集合, 即 $t_n^s - t_{n-1}^s = \{t_n^s - t_{n-1}^s, \dots, t_{n-l+1}^s - t_{n-l}^s\}$ 。第四项表示前 l 个视频块整体下载速率的集合, 即 $D_{n-1} = \{D_{n-1}, D_{n-2}, \dots, D_{n-l+1}\}$ 。第五项是上一个视频块的码率。第六项是剩余视频块的个数。

(2) 动作集

动作集包含决策算法可选择的全部动作, 与上述动态规划模型中的决策变量相对应, 本文设定动作集 A 为:

$$A = \{(R_n, F_n)\} \quad (14)$$

其中, $F_n = \{F_n^1, F_n^2, \dots, F_n^K\}$ 表示各条并行支路分片的数量, 因此可选动作的个数由可选码率集合、协同终端的数量以及分片的总数共同决定。

(3) 状态转移概率

状态转移概率是指, 在执行动作 a_n 后, 环境状态由 s_n 转移到 s_{n+1} 的概率。由于强化学习任务具有马尔可夫特性, 即下一时刻的状态只由当前时刻的状态决定, 因此, 状态转移概率可表示为:

$$P_{s_n \rightarrow s_{n+1}}^{a_n} = P(s_{n+1} | a_n, s_n) \quad (15)$$

由式 (12) 可知, 环境状态中下载时间包含信道的信息, 而信道的转移概率是不可预知的, 也就不能确定状态转移概率, 因此只能采取无模型的强化学习算法。

(4) 奖励函数

执行动作 a_n 后, 环境会反馈一个奖励值 r_n , 以衡量动作执行的好坏。在自适应视频流任务中, 用户 QoE 是评价决策好坏的指标。因此, 本文定义奖励函数 r_n 为:

$$r_n = \mu m(R_n) - \beta f(T_n) - \lambda |m(R_n) - m(R_{n-1})| \quad (16)$$

由于一个视频包含很多视频块, 因此面临多轮决策, 而衡量一个策略的好坏要通过累计奖励来判断。本文采用折扣累计奖赏 $E[\sum_{n=0}^{\infty} \gamma^n r_n]$ 。其中,

$\gamma \in (0,1)$ 是折扣因子, $\mathbb{E}[\cdot]$ 表示随机变量的期望。

强化学习的目标是最大化累计奖励。因此, 基于强化学习的视频流任务可以表示为:

$$\max_{a_n=(F_n^i, R_n)} \mathbb{E}[\sum_{n=0}^{\infty} \gamma^n r_n] \quad (17)$$

$$\text{s.t.} \begin{cases} R_n \in R_{\text{all}} \\ R_n = \frac{1}{T_c} \sum_{i=1}^K R_n^i \\ R_n^i = \frac{F_n^i}{F} T_c R_n, F_n^i \in \{0,1,\dots,F\} \\ n = \{1,2,\dots,N\}, i = \{1,2,\dots,K\} \end{cases} \quad (18)$$

在该强化学习任务中, 由于状态转移概率未知, 无法事先对策略进行评估, 从而导致策略的调整受阻。于是, 解决该强化学习任务需要运用无模型的强化学习算法。考虑到网络环境的状态维度巨大, 并且无法穷举状态—动作对。因此, A3C 算法应运而生, 将策略梯度方法和值函数相结合, 并引用神经网络来产生策略和值函数, 以适应高维度的网络状态。

4 基于 A3C 的视频流控制算法

决策代理需要通过不断的尝试来学习策略 π , 这里的策略 π 是一种概率。即在当前状态 s_n 下, 执行动作 a_n 的概率为 $\pi(a_n|s_n) \in [0,1]$ 。状态值函数 $V^\pi(s_n)$ (简称为值函数) 是策略更新的依据, 即在状态 s_n 下, 运用策略 π 所带来的长期累计奖赏, 其中基于折扣奖赏的值函数可表示为:

$$V_\gamma^\pi(s_n) = \mathbb{E}_\pi[\sum_{n=0}^{\infty} \gamma^n r_n | s_0 = s_n] \quad (19)$$

传统的 Actor-Critic 算法^[14]运行两张卷积神经网络 (convolutional neural network, CNN), 作为 Actor 网络和 Critic 网络, 利用非线性函数拟合的方式, 分别产生策略 π 和值函数 V 。即 $\pi(a_n|s_n) \approx \pi(a_n|s_n; \theta_a)$ 和 $V_\gamma(s_n) \approx V_\gamma(s_n; \theta_c)$ 。Critic 网络的作用是对 Actor 网络产生的策略进行评估, 辅助 Actor 网络更新参数。但 Actor-Critic 算法无

法去除样本间的相关性, 导致神经网络很难收敛。A3C^[15]算法采用多线程的异步学习架构。并行运行的多个局部代理与全局代理具有相同的网络结构, 局部代理之间彼此独立, 各自与环境交互, 并收集经验样本。这样就消除了样本之间的相关性。同时, 多个局部代理一同更新全局代理的网络参数, 大大加快了网络收敛的速度。

在 A3C 算法中, Critic 网络采用基于时序差分 (temporal difference, TD) 的梯度下降法来更新网络参数 θ_c , 即:

$$\theta_c \leftarrow \theta_c - \alpha \sum_n \nabla_{\theta_c} (r_n + \gamma V_\gamma^{\pi(\theta_a)}(s_{n+1}; \theta_c) - V_\gamma^{\pi(\theta_a)}(s_n; \theta_c))^2 \quad (20)$$

其中, α 表示 Critic 网络的更新步长。 $r_n + \gamma V_\gamma^{\pi(\theta_a)}(s_{n+1}; \theta_c) - V_\gamma^{\pi(\theta_a)}(s_n; \theta_c)$ 表示 TD 误差。Actor 网络采用基于优势函数的梯度上升法来更新网络参数 θ_a , 即:

$$\theta_a \leftarrow \theta_a + w \sum_n \nabla_{\theta_a} \log \pi(a_n|s_n; \theta_a) A^{\pi(\theta_a)}(a_n|s_n; \theta_c) + \varphi \nabla_{\theta_a} H(\pi(a_n|s_n; \theta_a)) \quad (21)$$

其中, w 表示 Actor 网络的更新步长。 $A^{\pi(\theta_a)}(a_n|s_n; \theta_c)$ 表示优势函数, 即运用策略 $\pi(\theta_a)$ 所获奖励与预期奖励之间的差值。由于 Critic 网络的目的是输出值函数, 以评估 Actor 网络产生的策略。因此, 将 TD 误差作为优势函数的估计, 即 $A^{\pi(\theta_a)}(a_n|s_n; \theta_c) \approx r_n + \gamma V_\gamma^{\pi(\theta_a)}(s_{n+1}; \theta_c) - V_\gamma^{\pi(\theta_a)}(s_n; \theta_c)$ 。 $H(\cdot)$ 表示正则化熵值, 以保证决策代理充分的探索环境空间。 φ 表示熵的系数, 一开始可以设置为较大的值, 使得 Actor 网络的参数向着熵值增大的方向改变。具体的算法步骤 (任意一个线程的局部代理) 如算法 1 所示。

算法 1 基于 A3C 的自适应视频流传输控制算法

状态空间 $S = \{B_n, C_{n-1}, t_n^s - t_{n-1}^s, D_{n-1}, R_{n-1}, N_{n-1}\}$, 动作空间 $A = \{(R_n, F_n)\}$



输入 全局代理的网络参数 θ_c 和 θ_a ，全局迭代次数 m ，全局最大的迭代次数 M

输出 迭代后的网络参数 θ_c 和 θ_a

1) 初始化

初始化视频块播放序号 $n = 1$

2) repeat

初始化起始状态 $s_n = s_0$

初始化网络参数增量 $d\theta_c = 0, d\theta_a = 0$

将全局代理的网络参数赋值给局部代理

$\theta'_c = \theta_c, \theta'_a = \theta_a$

for $n = 1, 2, \dots, N$ do

基于策略 $\pi(a_n | s_n; \theta'_a)$ ，执行动作 a_n

从视频播放环境中获得奖励值 r_n 和新的状态

s_{n+1}

$n \leftarrow n + 1, m \leftarrow m + 1$

end for

计算状态 s_N 的值函数 $V_\gamma^{\pi(\theta'_a)}(s_N; \theta'_c)$

for $i = N - 1, N - 2, \dots, 1$ do

更新 θ_a 的增量 $d\theta_a \leftarrow d\theta_a + \nabla_{\theta_a} \log \pi(a_i | s_i; \theta'_a)$

$A^{\pi(\theta'_a)}(a_i | s_i; \theta'_c) + \phi \nabla_{\theta_a} H(\pi(a_i | s_i; \theta'_a))$

更新 θ_c 的增量 $d\theta_c \leftarrow d\theta_c + \nabla_{\theta_c} (r_i + \gamma V_\gamma^{\pi(\theta'_a)}$

$(s_{i+1}; \theta'_c) - V_\gamma^{\pi(\theta'_a)}(s_i; \theta'_c))^2$

end for

更新全局代理的网络参数 $\theta_a \leftarrow \theta_a + wd\theta_a$,

$\theta_c \leftarrow \theta_c - \alpha d\theta_c$

until $m > M$

由上述算法步骤可知，A3C 算法的时间复杂度主要来源于 Actor 网络的复杂度 $O(T_a)$ 、Critic 网络的复杂度 $O(T_c)$ 、视频块的个数 N 、局部代理的个数 N_l 以及迭代的总次数 M 。因此，A3C 算法的时间复杂度可表示为 $O((MNT_a + MT_c) / N_l)$ 。而传统的 Actor-Critic 算法^[14]的时间复杂度为 $O(MNT_a + MT_c)$ 。对比可得，A3C 算法的时间复杂度较低，这是由于 A3C 算法采用了多线程异步

学习架构，可以大大加快训练的速度。

5 仿真分析与性能评估

本文的仿真数据来源于 Pensieve^[12]使用的实测网络轨迹。首先基于第 2 章提出的系统模型构造多流并发模拟器，然后使用 TensorFlow 对提出的基于 A3C 的视频流控制方法进行评估，最后使用 Matplotlib 将评估结果可视化。

5.1 仿真环境和参数设置

为了构建多流并发模拟器，需要对环境参数进行设置。本文设置的环境参数见表 1。

表 1 多流并发环境参数

参数	取值
视频块个数 N	48
视频块长度 T_c/s	4
视频分片长度 T_f/s	1.3
缓冲区容量 B_c/s	50
缓冲区起始长度 T_{start}/s	4
视频分片个数 F	3
异构终端个数 K	3
缓冲区上溢等待时间 T_s/s	0.5

除了表 1 中的参数外，由于异构终端的数量为 3，需要配置 3 个信道的网络带宽。因此，本文根据异构终端网络的实际差异，从网络轨迹中选择差异性明显的数据进行组合，最终选择了 62 组不同的网络轨迹。本文配置的可选码率集合 $R_{all} = \{0.3, 0.75, 1.2, 1.85, 2.85, 4.3\}$ Mbit/s。

根据码率集合 R_{all} 以及视频分片个数 F ，本文设定可选的动作个数为： $6 \times 10 = 60$ 。A3C 算法的参数设置见表 2。

在训练阶段，熵的系数 ϕ 设置为较大值 0.2，以充分地探索视频播放环境。随着迭代次数增加，逐渐减小 ϕ 的值到 0.000 1，使得网络收敛。在测试阶段，直接根据训练完成的 Actor 网络输出策略 $\pi(a_n | s_n; \theta_a)$ ，决策代理根据策略来选择动作。

表2 A3C 算法参数

参数	取值
折扣因子 γ	0.99
视频信息数量	8
局部代理个数	16
Critic 网络的更新步长 α	0.001
Actor 网络的更新步长 ω	0.01
熵的系数 φ	0.2

5.2 性能评估

图4展示了一组用于测试的网络轨迹。从上到下,分别为异构终端1、2和3的网络带宽随时间变化的情况。从图4中可以看出,各异构终端的网络带宽变化趋势各不相同,并且在各自的时间域上波动较大。

在图4所示的信道带宽条件下,运用本文所提方法,其中 $(\mu, \beta, \lambda) = (0.5, 0.3, 0.2)$, 码率选择和分流情况如图5所示。其中,分流情况是将分片数量转换成比特率,即 R_n^i / T_c 。图4中的时间

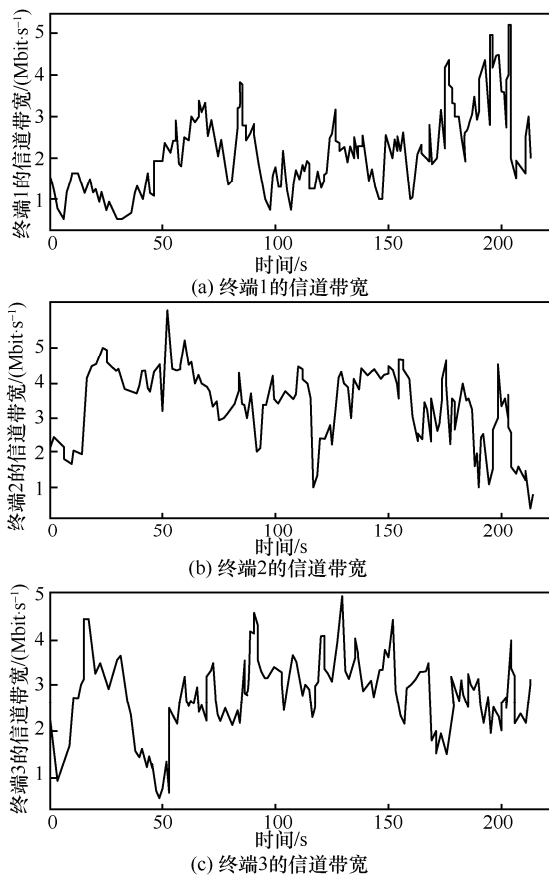


图4 各异构终端的信道带宽

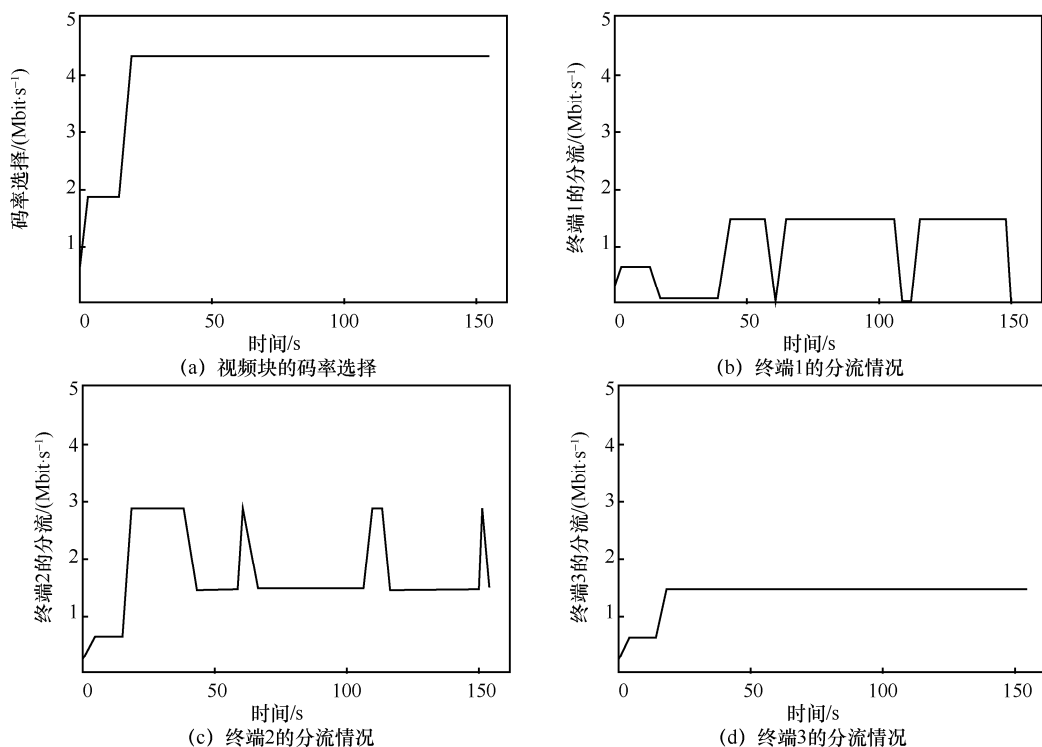


图5 码率选择和分流情况



与图 5 中的视频块时间序列并没有一一对应的关系,这是因为视频块的下载时快时慢,但变化的趋势是相同的。从整体来看,对于首个视频块,由于没有任何的反馈信息,本方法默认选择较低的码率;在随后的视频块中,主要选择了 1.85 Mbit/s 和 4.3 Mbit/s 两种码率,这是由于奖励值的指标中,视频质量的权重较大,为了获得最大的累计奖励,A3C 算法会先选择较低的码率,以维持缓冲区的占用率,再选择最高的码率。从各终端的分流情况来看,当信道带宽差异不大时,各协同终端的分片数量大多为(1,1,1)。但在 10~40 s,信道 1 的带宽远远小于信道 2 的带宽,而信道 3 的状况介于二者之间。因此,算法选择了(2,0,1)的分配方案。这样的协同方式能够避免信道带宽波动带来的影响。

缓冲区占用状况和系统整体下载速率如图 6 所示。对于缓冲区占用率,一开始选择的码率较低,下载的时间较短,即 $(T_c + B_n - (t_n^e - t_n^s)) > 0$,此时缓冲区的占用率增加;随后,系统选择了最高的码率,下载的时间较长,即 $(T_c - (t_n^e - t_n^s)) < 0$,此时缓冲区的占用率开始减少;最后,虽然系统仍然选择了最高码率,但是信道状况良好,协同终端的整体下载速率很高。因此,缓冲区的占用量反而增加了,即 $(T_c - (t_n^e - t_n^s)) > 0$ 。从图 6 中可以看出,缓冲区没有出现上溢和下溢的情况,避免了视频播放中断。对于系统整体下载速率,其中最低速率也达到了 2 Mbit/s,说明在多流并发环境下,即使某一个终端出现了网络中断的情况,也不会影响用户观看视频的流畅性。图 6 中水平的横线表示平均下载速率维持在 5~6 Mbit/s,与单链路相比具有更快的下载速率。

为了评估本文所提方法的性能,本文首先与单链路的 Pensieve^[12]方法和 MPC^[10]方法进行比较。分别将上述方法应用到 62 组网络轨迹中,每一组网络轨迹均包含 3 条信道,评估它们的平均 QoE 和累计 QoE。评估结果如图 7 所示。

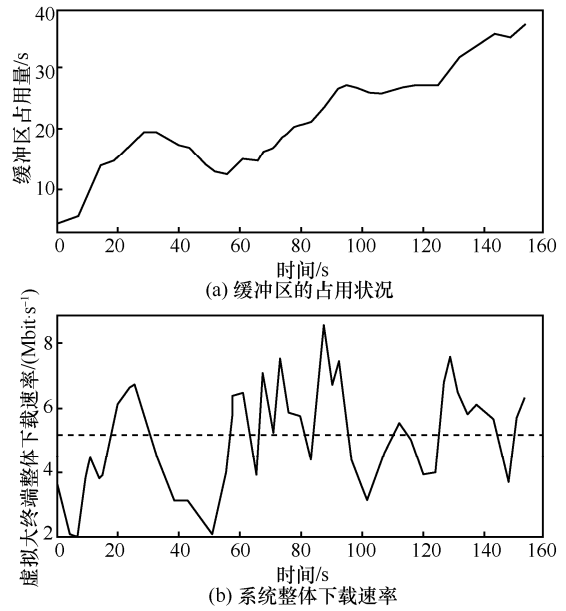


图 6 缓冲区占用状况和系统整体下载速率

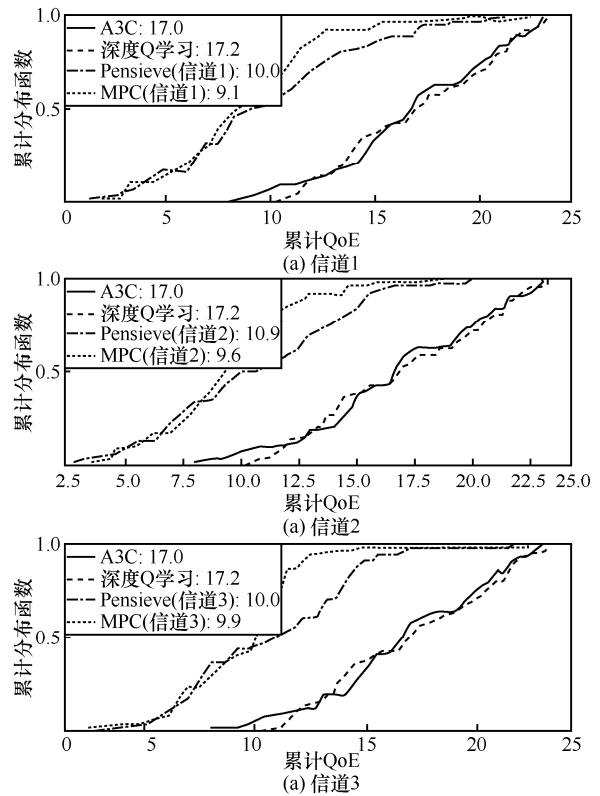


图 7 本文所提方法与单链路的 Pensieve 和 MPC 的性能比较

图 7 比较了上述两种单链路方法与本文所提方法的平均 QoE 和累计 QoE 的分布。从图 7 中可以看出,本文所提方法的平均 QoE 远大于单链路的方法,3 个终端协同的视频流控制方法相比于单

链路方式性能提升了 54.5%，说明多流并发的方式很好地提升了用户 QoE。同时，从累计 QoE 的分布函数中可以看出，采用多流并发技术，可以有效避免获得较低的累计 QoE，同时提供更大的累计 QoE，从而提升视频的质量和流畅度。

对于所提多流并发场景，本文与 Q 学习^[16]算法和单个决策代理的 A3C 算法进行比较，以评价 3 种算法在多流并发环境下的性能。性能评估结果如图 8 所示。

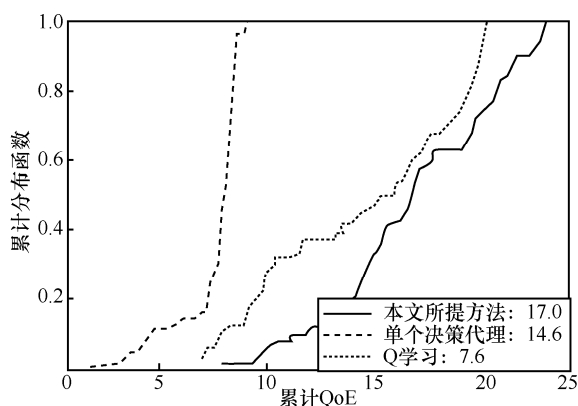


图 8 本文所提方法、单个决策代理的 A3C 算法和 Q 学习的性能比较

图 8 展示了 3 种强化学习算法所获 QoE 的累计分布函数。从图 8 中可以看出，Q 学习算法表现很差，表现甚至不如单链路的方法。这是因为，Q 学习主要通过创建 Q 表来存储状态—动作对，但对于视频流的播放环境，状态—动作对的维度是巨大的，显然会给 Q 表带来维度灾难，使得 Q 学习算法无法获取准确的网络信息。对于单个决策代理的 A3C 算法，失去多线程的局部代理后，训练速度很慢，并且样本间的相关性也大大增加，使得网络参数很难收敛，性能相比多线程的 A3C 算法较差。

6 结束语

本文主要研究了在多流并发场景下的 ABR 算法和分流策略。针对无线异构网络的特点，本文提出了一种基于 A3C 算法的自适应视频流传输

控制方法。首先，建立了视频流动态规划模型，对视频码率和分流策略进行联合优化。但是，该模型的求解依赖于精确的信道估计，于是将该动态规划问题改进为强化学习任务，利用 A3C 算法求解。最后，通过实测的网络数据对本文所提方法进行评估。与单链路传输的 Pensieve 和 MPC 优化方法，以及多流并发场景下的 Q 学习、单个决策代理的 A3C 算法相比，本文所提方法更好地提升了用户 QoE。

所提强化学习任务中，视频块的长度是固定的，对于可变的视频块长度未做分析。并未对存在较强信号干扰的一般化传输场景进行分析^[17-18]。同时，考虑与新型的视频编码方案结合^[19-21]。希望在今后的工作中，改进优化方法，将强化学习算法与传统的优化方法相结合，以扩展算法的适用范围。

参考文献:

- [1] Cisco. Cisco visual networking index: global mobile data traffic forecast update 2016–2021[R]. 2018.
- [2] SUN L, TIAN H, SUN Q Y, et al. Traffic allocation scheme with cooperation of WWAN and WPAN[J]. IEEE Communications Letters, 2010, 14(6): 551-553.
- [3] AKHSHABI S, BEGEN A C, DOVROLIS C. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP[C]//Proceedings of Second ACM SIGMM Conference on Multimedia Systems. New York: ACM Press, 2011.
- [4] SUN Y, YIN X, JIANG J, et al. CS2P: improving video bitrate selection and adaptation with data-driven throughput prediction[C]//Proceedings of the 2016 ACM SIGCOMM Conference. New York: ACM Press, 2016.
- [5] JIANG J C, VYAS S, HENRY M, et al. CFA: a practical prediction system for video qoe optimization[C]//Proceedings of USENIX Conference on Networked Systems Design & Implementation. [S.l.:s.n.], 2016.
- [6] WANG C R, TIAN H, MIAO J. Dynamic traffic allocation scheme for optimum distribution in heterogeneous networks[C]//Proceedings of IEEE Vehicular Technology Conference. Piscataway: IEEE Press, 2011.
- [7] ZHOU B, WANG J, ZOU Z, et al. Bandwidth estimation and rate adaptation in HTTP streaming[C]//Proceedings of IEEE ICNC. Piscataway: IEEE Press, 2012: 734-738.
- [8] HUANG T Y, JOHARI R, MCKEOWN N, et al. A buffer-based



- approach to rate adaptation: evidence from a large video streaming service[J]. Computer Communication Review, 2014, 44(4): 187-198.
- [9] YIN X, JINDAL A, SEKAR V, et al. A control-theoretic approach for dynamic adaptive video streaming over HTTP[J]. Computer Communication Review, 2015, 45(4): 325-338.
- [10] ZHOU C, LIN C W, ZHANG X, et al. A control-theoretic approach to rate adaption for dash over multiple content distribution servers[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2014, 24(4): 681-694.
- [11] GUO Y, YU R, AN J, et al. Buffer-aware streaming in small scale wireless networks: a deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2019, 68(7): 6891-6902.
- [12] MAO H, NETRAVALI R, ALIZADEH M. Neural adaptive video streaming with pensieve[C]//Proceedings of the Conference of the ACM Special Interest Group. New York: ACM Press, 2017.
- [13] ALLMAN M, PAXSON M, BLANTON E. TCP congestion control: RFC5681[S]. 2009.
- [14] KIMURA H, KOBAYASHI S. An analysis of actor/critic algorithms using eligibility traces: reinforcement learning with imperfect value function[C]//Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998). [S.l.:s.n.], 1998.
- [15] MNIH V. Asynchronous methods for deep reinforcement learning[C]//Proceedings of International Conference on Machine Learning. [S.l.:s.n.], 2016.
- [16] CHIARIOTTI F. Online learning adaptation strategy for dash clients[C]//Proceedings of the 7th International Conference on Multimedia Systems. New York: ACM Press, 2016.
- [17] LI C G, YANG H J, SUN F, et al. Multiuser overhearing for cooperative two-way multiantenna relays[J]. IEEE Transactions on Vehicular Technology, 2016, 65(5): 3796-3802.
- [18] LI C G, ZHANG S, LIU P, et al. Overhearing protocol design exploiting inter-cell interference in cooperative green networks[J]. IEEE Transactions on Vehicular Technology, 2016, 65(1): 441-446.
- [19] 董振江, 张东卓, 黄成, 等. 虚拟现实视频处理与传输技术[J]. 电信科学, 2017, 33(8): 45-52.
- DONG Z J, ZHANG D Z, HUANG C, et al. Technology of virtual reality video processing and streaming[J]. Telecommunications Science, 2017, 33(8): 45-52.
- [20] 周建同, 杨海涛, 刘东, 等. 视频编码的技术基础及发展方向[J]. 电信科学, 2017, 33(8): 16-25.
- ZHOU J T, YANG H T, LIU D, et al. Trends and technologies of video coding[J]. Telecommunications Science, 2017, 33(8): 16-25.
- [21] 马思伟, 罗法蕾, 黄铁军. AVS2 视频编码标准技术特色及应用[J]. 电信科学, 2017, 33(8): 3-15.
- MA S W, LUO F L, HUANG T J. Kernel technologies and applications of AVS2 video coding standard[J]. Telecommunications Science, 2017, 33(8): 3-15.

[作者简介]



罗志强 (1997-), 男, 南京邮电大学通信与信息工程学院在读, 主要研究方向为无线异构网络视频流自适应传输。



王伟 (1997-), 男, 南京邮电大学通信与信息工程学院硕士生, 主要研究方向为无线异构网络视频流自适应传输。



朱晓荣 (1977-), 女, 博士, 南京邮电大学通信与信息工程学院教授、博士生导师, 主要研究方向为下一代无线网络、异构网络。