

---

# 面向终端用户的 5G 切片系统及策略研究

---

所在赛道与赛项: A

作品关键词: 网络智能、软件定义网络、无线通信网络

2022 年 8 月 27 日

# 目录

<b>1</b>	<b>目标问题与意义价值</b>	<b>1</b>
1.1	作品背景	1
1.1.1	研究背景和趋势	1
1.1.2	面临的问题和挑战	2
1.2	国内外研究现状	3
1.2.1	5G 切片技术框架	3
1.2.2	切片流程及策略现状	4
1.2.3	云渲染应用及其切片挑战	5
1.3	作品研究的意义、内容和目标	5
1.3.1	作品研究的意义	5
1.3.2	作品研究的内容	6
1.3.3	作品研究的目标	6
<b>2</b>	<b>5G 切片系统需求分析与方案论证</b>	<b>8</b>
2.1	云渲染应用用户需求量化	8
2.2	关键技术分析	11
2.3	系统需求分析	13
2.4	可行性分析及方案论证	13
2.4.1	终端可行性及实现工具	14
2.4.2	服务器可行性及实现工具	14
2.4.3	切片策略服务器可行性及实现工具	14
<b>3</b>	<b>面向终端的 5G 逐用户切片系统设计</b>	<b>16</b>
3.1	系统总体设计	16
3.1.1	系统架构设计	16
3.2	策略方法设计	17
3.3	功能模块设计	18
3.3.1	终端模块设计	18
3.3.2	网络与服务器模块设计	19
3.3.3	策略服务器模块设计	20
<b>4</b>	<b>面向终端的 5G 逐用户切片系统实现</b>	<b>26</b>
4.1	终端功能模块实现	26
4.1.1	网络质量监测模块实现	26
4.1.2	用户需求分类模块实现	30
4.1.3	切片触发模块及协议实现	31

4.1.4	通信模块实现 . . . . .	32
4.2	服务器实现 . . . . .	32
4.2.1	云渲染数据通信模块实现 . . . . .	33
4.2.2	策略服务器通信模块实现 . . . . .	34
4.2.3	评价与预测模块实现 . . . . .	34
4.2.4	切片决策模块实现 . . . . .	34
4.3	服务器切片管理实现 . . . . .	35
4.3.1	切片部署实现 . . . . .	35
4.3.2	切片调整与释放实现 . . . . .	35
<b>5</b>	<b>性能测试与分析</b>	<b>36</b>
5.1	测试环境 . . . . .	36
5.2	功能测试 . . . . .	36
5.2.1	终端功能测试 . . . . .	36
5.2.2	通信功能测试 . . . . .	38
5.2.3	切片功能测试 . . . . .	39
5.3	切片框架性能测试 . . . . .	40
5.3.1	文件传输本地服务器测试 . . . . .	40
5.3.2	文件传输云服务器测试 . . . . .	42
5.3.3	视频渲染切片框架结果 . . . . .	44
5.3.4	视频渲染强化学习效果 . . . . .	46
<b>6</b>	<b>创新与特色</b>	<b>47</b>
6.1	作品创新点 . . . . .	48
6.2	作品特色 . . . . .	48

# 1 目标问题与意义价值

本章中首先介绍了当前移动网络通信技术面临的挑战和其发展趋势，然后对 5G 切片技术的概念及发展现状进行分析，详尽介绍了国内外在 5G 切片技术相关研究工作，并重点分析该技术在云渲染场景中的应用所需解决的调整，最后对本文的主要研究内容，目标、工作意义及文章的内容组织作出了具体说明。

## 1.1 作品背景

### 1.1.1 研究背景和趋势

随着移动数据业务流量的迅猛增长和各类新型移动应用场景的不断成熟，4G 及之前的移动通信技术逐渐难以满足其需要。为解决 4G 移动通信技术面临的挑战，应对日益多样化的网络服务需求，满足与日俱增的带宽通信需求，第五代移动通信技术（5G）被正式提出，并应用推广于社会生产和消费的方方面面。

针对各类终端应用的网络需求特性，5G 技术将其主要面向的业务划分成三大主流应用场景，分别为增强型移动宽带（Enhanced Mobile Broadband, eMBB）、超高可靠性低时延通信（Ultra Reliable Low Latency Communication, URLLC）和海量机器类通信（Massive Machine Type of Communication, mMTC），其典例和特性需求如图1.1所示。

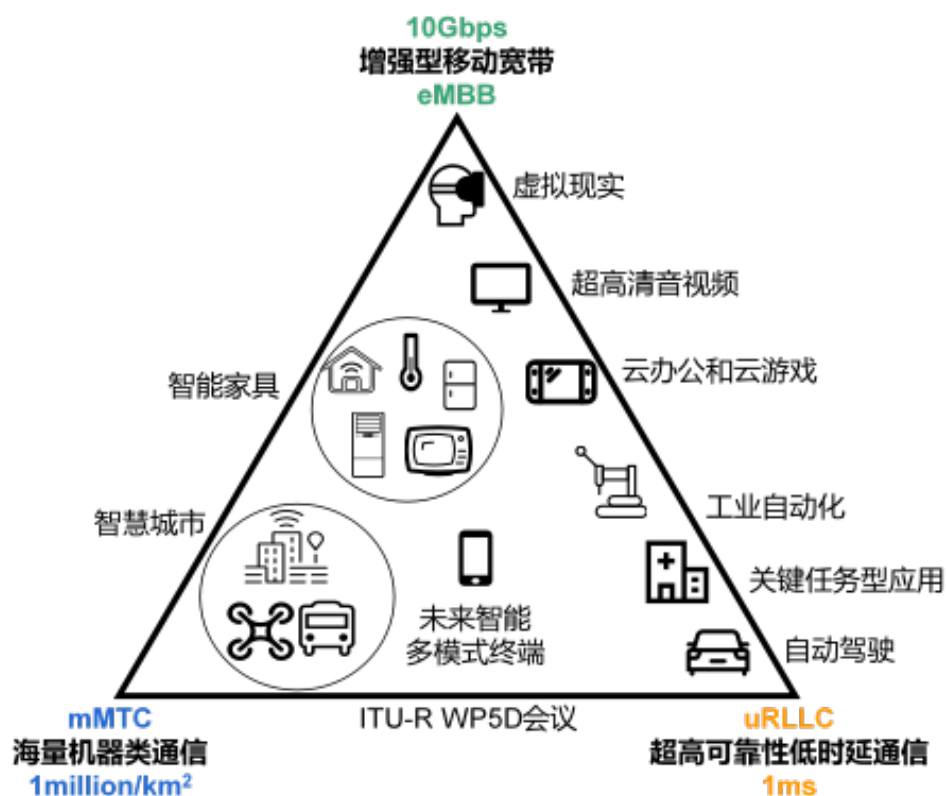


图 1.1: 5G 网络三大主流应用场景

作为新一代网络通信技术，5G 网络可提供高带宽，低时延和大连接容量的网络服务。在我国 IMT-2020(5G) 推进组推出的白皮书《5G 愿景与需求》和《5G 网络技术架构》中，对 5G 网络用户体验速率，端到端时延，空口时延和连接数密度等 5G 网络各方面关键性能指标均给出了相应规定。

由于 5G 多样化的应用场景与业务需求，在不同的应用场景下，上述指标并非固定不变。为在同一物理网络设施中充分利用网络资源，灵活地为多种应用提供定制网络服务，5G 引入了网络切片作为其核心技术之一。网络切片技术可运用网络功能虚拟化与软件定义网络技术，设计并实现了松耦合的新型网络架构，为各类应用服务部署专用的定制化端到端逻辑子网络。

在 5G 切片技术实际应用的过程中，其在工业领域的远程控制等固定网络、固定场景、固定物理环境的应用已有相对成熟的解决方案和行业实践。而在移动终端领域，5G 切片技术运用到游戏（特别是云游戏）、虚拟现实（虚拟演唱会）、自动驾驶等需求高带宽低时延的云渲染应用的过程中仍存在较多问题。一方面，如图 1.1 所示，云渲染应用受本身类型特性影响，兼有两大类应用场景特点，且受用户设置及终端设备性能影响，现有划分的三大类应用场景不足以准确描述用户需求。另一方面，粗粒度切片中仍存在大量用户乃至业务的动态接入，难以避免业务及用户间的资源竞争，无法保障云渲染类终端应用网络服务的稳定性。

### 1.1.2 面临的问题和挑战

目前，5G 切片技术正处于应用于固定终端和移动终端各类应用的发展阶段。其在真实移动终端场景下，面向云渲染类等高网络要求应用的 5G 切片服务系统仍未有较为完善的实践成果与解决方案，其尚待解决的问题与挑战可简要分为以下几点：

- (1) 网络服务封装问题。在云渲染类应用场景下，其应用流畅运行的保障往往需要网络各部分功能与资源共同协作。如何将应用所需的渲染算力服务，网络资源及网络功能较好地封装并整合，为切片决策提供可执行的功能与部署空间，是亟待解决的基础问题。
- (2) 网络状态测量问题。移动终端设备的网络状态复杂多变，相对于固定终端的工业类应用场景，其网络质量不确定性更强，应用网络需求变化快。若系统对此无法做出准确的感知乃至预测，切片建立时机过早或过晚，切片资源划分过于静态，则可能无法起到及时、准确、实时地改善网络质量的效果。
- (3) 切片策略效率问题。5G 引入切片技术的初衷便是为在同一物理网络中低成本且高效率地为差异化应用提供对应的虚拟专网服务。因此，如何将应用的服务需求转化为准确的网络资源，提供按需差异化的高效精准网络服务，是 5G 切片技术能否满足其初衷的核心问题。

- (4) 用户体验评估问题。在 5G 切片技术实际应用的过程中，一方面，切片框架与切片策略效果需从用户实际应用体验变化情况中得到反馈并优化；另一方面，切片技术需要可度量可感知的用户体验改善情况，以证明其实际价值，推动技术发展。如何针对切片的性能，切片前后用户体验改善情况进行评估，是 5G 切片技术切实投入实际应用所必须面对的效益问题。

## 1.2 国内外研究现状

### 1.2.1 5G 切片技术框架

自 5G 网络切片方案被提出之后，国内外的科研人员针对 5G 网络切片中的相关问题做出一系列的研究工作。

从网络范围的角度看，从用户终端到服务器的整个端到端网络切片可竖向划分为无线网，传输网和核心网三大网络部分，能提供效果可确定的网络服务。相比之下，非端到端的网络切片，剩余部分通过传统网络传输的切片形式开销相对较小，但其网络服务保障能力也会相对降低。

从切片网络结构的角度看，各研究综述中将网络切片架构横向划分为基础设施（物理设备）层，网络功能层，服务层和切片服务编排系统四大部分。基础设施层由网络硬件设备构成，在此基础上，运用软件定义网络（Software Defined Network, SDN）和网络功能虚拟化（Network Function Virtualization, NFV）技术对网络硬件设备所能提供的带宽等网络资源及所能实现的流量整形，时延敏感等技术进行封装，形成了网络功能层。而在服务层中，切片技术将结合网络功能层提供的网络资源与网络功能，针对用户需求，提供网络切片服务。具体如何针对服务请求与网络功能层提供的资源进行编排，则有切片服务编排系统进行决策。

在无线接入网的切片研究领域，由于其无线特性，基站各切片之间的干扰成为了一大挑战。如何动态划分频谱、功率、天线等资源形成足以保障服务的切片，如何最小化切片间的干扰，如何保证各用户的公平都是相关领域的研究重点，目前在这一领域方面，有采用启发式算法寻求近似解；利用强化学习与深度学习算法，基于历史流量进行预测，从而分配与调度无线资源等策略的研究。

在传输网和核心网一侧，软件定义网络（Software Defined Network, SDN）和（Network Function Virtualization, NFV）的技术已可对网络资源与网络功能进行较为成熟的封装。而针对如何将这二者技术运用于切片技术的研究工作中，大多聚焦于如何进行网络切片的映射部署。一者是讨论网络功能到物理设施的映射与部署，即如何根据网络的实时变化与不同需求，进行虚拟网络功能（Virtual Network Feature, VNF）结点与实际物理结点之间的映射部署与迁移，以达到更好的经济效益，降低能耗；另一者是将网络服务需求与网络功能之间形成映射，确认网络服务需要的 VNF 及相应资源，确保切片有效的同时尽可能减少网络资源浪费。

针对如何将接入网，传输网和核心网提供的资源与用户需求进行映射，学术界另一大切片技术研究方向是关于如何选取合适的网络整体资源的编排管理策略。其中，如何根据切片需求或预测应用流量，设计切片生命周期流程，如何在抽象出来的网络资源与功能层上制定相应资源编排与分配策略，是相关方向的研究重点。

综上，在 5G 切片网络框架基础上，5G 切片技术的主要研究重点可分层总结为：物理设施层中无线资源虚拟化的研究；网络功能层中功能封装粒度的选择研究；服务层中关于如何描述应用服务需求的探讨，并映射到对应切片；服务编排管理系统中编排资源和切片的策略，高效利用的同时满足服务需求。

### 1.2.2 切片流程及策略现状

切片的生命周期流程，可分为设计与发起，编排部署，运行时调整管理和回收释放四步。

在切片的设计与发起阶段，一些研究者选择根据预先对网络系统及服务应用的认知设计网络切片模板，即针对不同网络服务需求，按照经验值配置相应切片参数。同时，亦有学者提出需在切片发起阶段加入准入判断，根据预测流量和当前可用资源决定是否准许相应切片请求。

编排部署阶段本质是根据策略分配网络资源和网络功能并形成切片，是最重要的一环。目前主流的两种方案是采取按需求部署切片和根据预测流量切片。前者往往在设计阶段配置好切片模板，确认各应用服务需求对应的网络功能和资源。因此，该方案在编排部署阶段主要是对网络服务需求进行分类；基于流量预测的方案则利用时序模型或深度学习算法进行流量预测，并可选对预测结果粗略分类，再据此提前在网络中隔离对应的资源，降低网络切片实例化时间。

创建的切片实例的粒度亦是一个需要讨论的问题。大到整个网络、三大类划分下的应用场景、小到同类应用程序、每组用户、每个个人用户，甚至每个用户的不同类型数据都可以作为一种划分切片的粒度，它们开销和服务效果各不相同，往往取决于应用网络需求程度。在一个切片实例服务多个用户的粒度下，可采用切片选择问题来描述用户建立切片请求的过程，使用采用遗传算法选择切片。

建立好切片后，由于需求和网络状态的动态变化，先前划分的切片可能出现效果下滑的情况。因此，为保障性能并最大化网络资源利用率，可对已创建的切片实例进行调整。有学者尝试采用线性规划模型最大化可靠部署收益并最小化开销，深度强化学习方法也正被应用到这一领域的研究中。切片的回收释放阶段相对而言采用的方法都比较简单，基本根据用户服务需求的降级或运营商的策略直接解绑用户与切片实例的关联，将其回收至资源池即可。因此，也可将切片回收这一步骤归类到前一步的切片管理流程中。

### 1.2.3 云渲染应用及其切片挑战

云渲染类应用指需在服务器端进行计算与画面渲染，终端接收数据并发出操纵指令的应用。在云渲染应用的整个流程中，首先由云渲染用户发起请求与操作指令，与源流媒体服务器交互。而后，源流媒体服务器将源分发至渲染服务器（边缘），再重定向用户到渲染服务器（边缘）。完成该步骤后，渲染服务器将接收用户控制指令交互，汇聚多源流进行渲染重建，最后再将渲染音视频流分发给相应用户。

基于云端渲染，终端操纵的特性，相比于传统网络应用而言，云渲染对网络的带宽，时延抖动等需求更高。然而，现有的尽力而为式网络难以满足该类应用的需求，网络能力已成为诸如增强现实（Augmented Reality, AR），虚拟现实（Virtual Reality, VR）等云渲染应用的发展一大瓶颈，需尝试将 5G 技术及 5G 切片技术应用于云渲染场景中，克服其网络瓶颈。

目前而言，学术界的 5G 切片策略多在理论仿真场景下进行实验，缺少实际场景下的数据采集与监测环节，对建立切片的实际用户体验效果评价亦没有相对成熟的方案。另一方面，云渲染网络指标及相应资源受应用类型与用户体验需求影响较大。若是在切片策略中依照三大主流应用场景进行切片类型设计与服务，切片种类粒度过大，难以较为精准地满足各云渲染应用网络需求。另一方面，预测策略大多在相对简单的局部网络场景与数据下，仅针对预先采集的带宽数据进行处理，并未在真实动态终端网络环境中进行实时性检验。

综上所述，若要将现有 5G 切片理论研究应用到云端渲染应用，目前存在尚未充分考虑移动终端网络动态变化，缺乏用户级及业务级粒度的资源预留与切片，切片策略针对云渲染类应用缺少一定的适配等问题，难以满足运行云渲染应用终端的移动用户需求。受困于这些问题，目前尚未有 5G 切片技术的实际应用实现。为此，本作品计划针对云渲染网络切片需求，基于已有分类与预测策略进行改进适配，在网络层和应用层组织网络资源与渲染网元，从用户粒度，业务粒度的角度进行按需切片，并给出相应的系统与方法的设计与实现。

## 1.3 作品研究的意义、内容和目标

### 1.3.1 作品研究的意义

云渲染应用正蓬勃发展，已成为诸多领域的一大发展方向。其场景包括且不限于云游戏、远程医疗（特别是远程手术）、VR、自动驾驶、远程工控、智慧园区等，它们均需要低延时、高带宽的稳定网络服务。

本作品将基于学术界 5G 切片技术研究，在提升服务质量和资源利用率的相关策略基础上进行适配与改进，针对云渲染场景下的移动终端，提出一种切实可行的 5G 切片机制系统及方法，针对逐个用户（简称：逐用户）在用户粒度进行网络切片的设计接入与部署，为用户提供基于其需求的精准网络服务，从而克服云端渲染本地展示的 VR，



AR 等高带宽差异化网络应用需求瓶颈，推动 5G 切片技术在云渲染场景应用中的实际使用。

总而言之，围绕云渲染终端场景开展的 5G 切片策略及机制研究工作，将有助于 5G 切片技术切实投入移动终端各类应用中，帮助各类应用克服网络瓶颈，具有 5G 切片策略研究的理论意义和 5G 切片技术实际运用的实践意义及商用价值。

### 1.3.2 作品研究的内容

本作品以云渲染应用中终端与服务器的通信为主要场景，5G 切片在其中的应用作为主要研究内容，探讨 5G 切片面向移动终端场景的机制策略，解决 5G 切片技术应用需克服的问题，并进行一定的实践。具体而言，有以下研究内容：

- (1) 调研分析云渲染应用网络需求，给出量化 QoE 指标，并选取云游戏作为云渲染应用典例进行深入阐述。从而基于云渲染应用需求基准，为用户需求进行建模，进而精确定位切片针对服务需求所需编排的相关网络资源，并对用户体验进行 QoE 量化评估。
- (2) 提出一种完整的面向移动终端的逐用户 5G 网络切片系统及策略方法。结合学术界最新切片策略研究成果及相应框架，该策略方法将分析云渲染应用对网络的需求情况，并尝试动态预测用户需求，以进行及时且精准的切片服务，提高切片策略效率，保障服务质量和资源高利用率。
- (3) 基于本系统在真实网络环境中搭建云渲染结合切片的实验系统。以安卓手机作为移动终端设备客户端，调研其网络质量数据可获取情况与相应获取接口，并对网络服务质量进行动态的监听，测量与建模；以 linux 系统为云渲染服务器，搭建可同时向多个 Android 或 Python 客户端按帧率逐帧传输音视频数据的云渲染实验环境。
- (4) 在实验系统中提出切片框架与策略进行实践，在手机终端，网络及服务器构成的云渲染应用运行场景中，在应用层实现结合带宽预测，用户按需切片，强化学习等切片策略，并通过相应技术工具实现网络资源的编排及切片部署。

### 1.3.3 作品研究的目标

本作品的研究目标如下：

1. 提出面向终端的 5G 逐用户切片系统及方法：该系统与方法将针对应用数据流特性，为各用户提供切片服务。系统与方法将考虑终端网络质量与实际用户需求，涵盖完整的切片流程，包括切片的发起、建立、部署和管理等部分，一定程度上解决 5G 切片技术应用到移动终端场景面临的问题。

2. 在真实 5G 网络环境中实现云渲染切片框架系统：基于手机终端安卓系统和服务器等可控网络资源，进行 Android 5G 终端与服务器的联合音视频服务器云渲染开发，模拟云渲染应用通信过程；在网络层和应用层实现所提出的切片框架及其策略，提供较好的云渲染应用网络服务。
3. 在实验系统中提出切片框架与策略进行实践：在应用层实现结合带宽预测，用户按需切片，强化学习等切片策略，并通过相应技术工具实现网络资源的编排及切片部署，按需精准将用户需求与网络资源映射，提供精准化高效的网络切片服务，改善云渲染用户网络质量。

## 2 5G 切片系统需求分析与方案论证

为解决 5G 切片技术应用于云渲染场景的问题，本节首先对云渲染应用的网络需求进行量化分析，并选取云游戏作为典例进行深入阐述，并以此为基准结合用户设置确认用户需求，进而对应设置相应网络资源切片需求。而后，文中对 5G 网络切片技术进行了进一步介绍，明确其对网络性能指标提升效果及方式和其应用缺陷。在此基础上，文中对将 5G 切片技术应用到云渲染场景的系统进行需求分析，并从一种可实现的真实网络实验环境出发，完成了可行性论证与采用工具选择。

### 2.1 云渲染应用用户需求量化

为解决 5G 切片策略效率问题，提高网络整体资源利用率，更好地保障用户网络服务，需对用户运行该应用的网络需求进行准确描述。为此，需先对各类云渲染应用进行一定的量化分析，并基于此建立用户需求的描述，完善需求到相应切片及资源的映射，实现服务描述到网络切片的转化。

基于云渲染类型应用云端渲染，终端操纵的服务特点，该类型应用需为音视频等渲染数据提供高带宽网络，为终端操纵指令提供低时延低抖动的强实时性网络服务。然而，不同种类云渲染应用的网络需求存在一定差异。应用类型特性对网络需求的影响会反应到网络切片的资源需求上。

在已有 5G 定义的三大类网络切片基础上，结合云渲染本身应用网络需求特性，可针对性将云渲染业务网络需求进行分类如表 2.1 所示：

表 2.1: 云渲染应用类型及其需求

云渲染应用类型	云渲染应用描述	云渲染应用网络需求
虚拟现实/增强现实	需较高带宽进行 3D 渲染结果传输	高带宽
远程工业控制	需保障关键任务指令的实时性	低时延低抖动
云游戏	受游戏类型影响，对各指标需求不同	对带宽、时延、抖动需求多样
远程医疗	受医疗类型影响，对各指标需求不同	对带宽、时延、抖动需求多样

通过以上应用需求的划分，可基于应用类型，设置用户需求的基准，确认其应用体验所需要的网络服务类型，进而定位切片所需资源，以精准改善网络服务。

其中，对云游戏，远程医疗这一类需求多样化的云渲染应用，其游戏类型、医疗服务类型较为丰富，影响对网络质量的需求。因此，该类应用需求分类结果涵盖云渲染应用的绝大多数需求，可作为云渲染典例。下面以云游戏为例，对云渲染应用的特点分类结果及相应网络需求进行进一步量化分析。

云游戏是以云计算为基础的游戏方式，本质为交互式的在线视频流。用户游玩云游

戏时，游戏内容的存储，计算和渲染都在云端，渲染完毕的音视频将在编码后通过网络发送给玩家。其网络架构如图2.1所示：

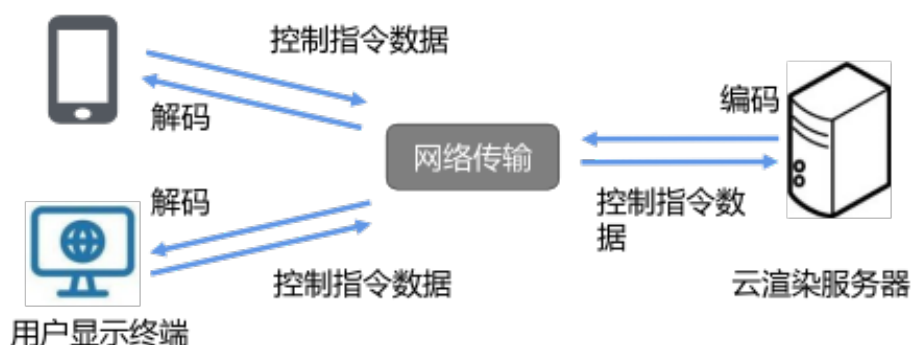


图 2.1: 云渲染应用网络拓扑

在手机显示终端上，基于视听体验和操作体验的需求，业界已给出以下云游戏需求指标：帧率：50-60fps、内容分辨率：1080P、建议码率：10-20Mbps、色深：8bit、操作响应时延：不高于 100ms。其余类云渲染应用亦对显示有相似需求。满足以上指标后，手机终端云游戏用户的内容清晰度，流畅度，色彩保真等效果较好，可确保终端用户拥有相对优质的云游戏体验。

为使云游戏渲染画面观感满足以上特征，对手机终端与渲染服务器的网络通信，可综合业界各实现的云游戏应用，给出以下云游戏基准网络 QoS 指标要求，如图2.2所示。

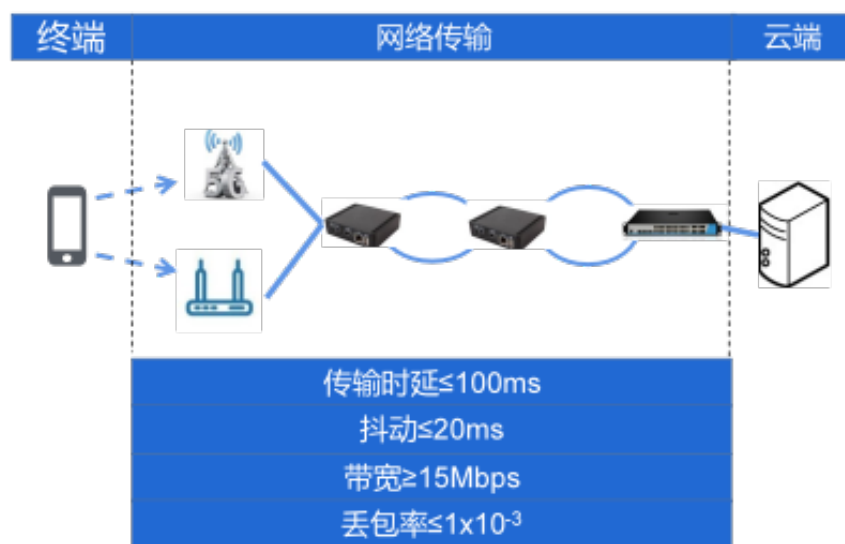


图 2.2: 云游戏基准网络需求

不同类型云游戏对画面，实时性，网络波动的要求不同，而如。因此，其在以上网络需求指标的基准上，会呈现出不同的容忍度，网络指标数值边界有相应变化。针对时延，带宽与抖动等网络特征，给出不同游戏类型特点需求如表2.2所示：

表 2.2: 云游戏类型及网络需求

云游戏类型	特点	网络需求
角色扮演/回合制、文字类、养成类	偏重剧情演出、故事渲染、只需简单选项操作	无
休闲类、闯关类	可能注重画质，需要一定带宽	高带宽
动作角色扮演、冒险类	些许抖动不会对体验带来直接影响	高带宽、低时延
射击类、竞速类、动作类、音游	对操作响应要求较为严格、时延需求不高于人都反应速度、对抖动敏感	高带宽、低时延、低抖动

另一方面，用户的终端应用设置所代表的需求差异也会对网络质量 QoS 指标需求产生影响。基于 xx 公司提供的云游戏应用云渲染脱敏数据，我们分别对用户设置为 720p, 30fps 低带宽高画质和 1080p, 60fps 高带宽高画质的实际云渲染码率及帧率进行了数据统计分析，实际测试结果分别如图2.3和图2.4所示。由测试结果可以看出，真实环境下 720p 画质云渲染的平均每秒传输帧数（FPS）约为 28.38，FPS 的众数为 29, 中位数为 29，码率均值为 3734.50，最大概率区间位于 3000 - 4000；而真实环境下 1080p 画质云渲染的 FPS 约为 44.79，FPS 的众数为 30, 中位数为 45，码率均值为 16071.90，最大概率区间位于 3000 - 4000。

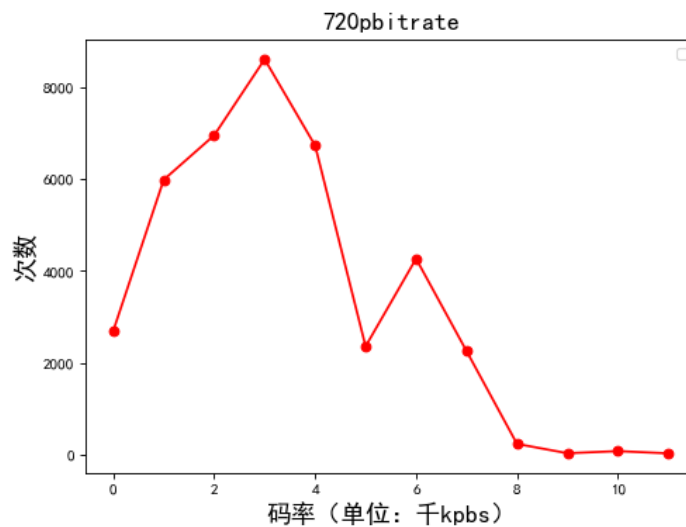


图 2.3: 真实环境下 720p 画质云渲染的码率和帧率需求

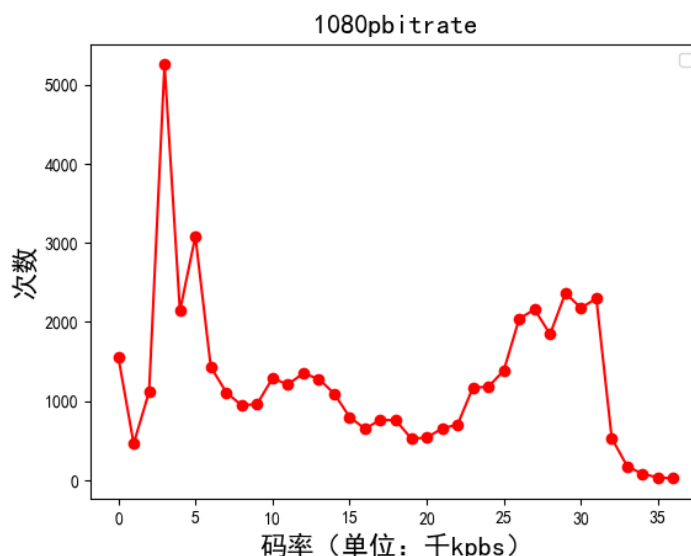


图 2.4: 真实环境下 1080p 画质云渲染的码率和帧率需求

基于以上网络需求分析结果,可在图2.2所给出的云游戏应用网络指标的基础上,在具体实施的过程中调整相应参数边界。如:将高带宽需求应用的下界调高至 15Mbps,时延抖动敏感型的时延上界设定为 70ms,抖动设定为 15ms 等。

在对 5G 逐用户切片系统所需服务的云渲染应用特征,以及用户的需求特征,做出准确的建模需求描述后,可通过获取云渲染应用类型,结合用户设备设置的方式,精准描述用户网络需求,进而确认其所需相应网络资源,从而进行按需精准分配资源形成切片,以实现高效 5G 切片策略,为网络状况不足以满足用户需求的云渲染终端用户改善网络服务。

## 2.2 关键技术分析

随着云渲染技术的不断发展,云渲染应用对网络的需求不断提高,现有 4G 网络技术难以满足其对带宽,时延等参数的需求,网络已成为制约其发展的主要瓶颈之一。5G 作为一种新型移动通信网络技术,其技术本身及其引入的网络切片技术可成为解决问题瓶颈的主要工具。

5G 端到端网络切片技术能基于一个 5G 网络虚拟出多个具备不同特性的逻辑子网,而每个端到端切片均由核心网、无线网、传输网子切片组合而成,并将通过端到端切片管理系统进行统一管理。下面对这数个部分各自的组成与作用进行介绍。

### (1) 切片控制系统

切片控制系统 (Network Slice Manager, NSM) 负责管理全局切片,需完成设计模板、实例化、运行时质量保证、回收切片等功能。

一种实现切片控制系统的方式是将 NSM 划分为两个部分。一为切片支持系统 (Slice Support System, SSS)，负责统一模板设计和服务需求指标拆解。前者根据网络容量、租户功能和性能需求来划分资源，设计切片模板；后者则管理切片实例的生命周期，包括切片与用户的绑定、运行时管理与销毁。另一个部分是域切片支撑系统 (Domain slice Support System, DSS)，旨在分网络子领域落实指标需求。它包括了接入域、核心域和传输域各自的子域切片管理工具 (Domain Slice Managers, DSM) AN-DSM、CN-DSM、TN-DSM。每个子域的 DSM 对于各域提供子域划分资源、供应、运行时保障、销毁的能力，并保证服务质量独立与闭环资源控制。

## (2) 接入网协同切片

接入网 (Access Network, AN) 侧需要满足访问资源逻辑独立和差异化访问两种特性，前者使得 AN 资源可被隔离且逻辑上资源复用，资源分配可根据每个网络切片实例流量调整；后者旨在分别利用非实时性功能来应用云化和编排技术，利用实时性功能保证资源利用率和确保网络切片实例的安全和隔离性。其落实方案可有：

1. 对 MAC 层与物理无线资源配置不同参数 (传输点、频谱、时间等) 实现无线资源的逻辑切分；
2. 通过定制化协议栈的方式，减少不必要功能带来的延迟和浪费。

## (3) 传输网协同切片

传输网包括前传网，回传网，城域网，骨干网等。在传输网中，需协调数据路径上不同网络段使用的协议，并支持其不同切片类型程度的隔离。在链路层与物理层实现中，将使用灵活以太网等技术保障切片的部署。

## (4) 核心网协同切片

核心网侧切片需要实现为用户提供控制平面，用户平面等的网络功能 (Network Function, NF)，并进行相应切片管理。即：针对不同网络切片实例分配诸如 AMF, PCF, UDN 等 NF，控制 NF 的复用性与独立性。

综合以上介绍，不难看出，网络切片技术的本质是：在利用网络乃至通信客户端，服务器等封装实现的可控制软件化资源的基础上，通过切片控制系统，对相应应用用户的需求编排资源，预留形成切片，而后部署到网络各可控部分中。

然而，由于在第三代合作伙伴计划 (3rd Generation Partnership Project, 3GPP) 的 R15 阶段 (2017 年 8 月) 前后，网络切片才被纳入 3GPP 的定义中，目前业界对于切片自动化编排的流程与接口均未统一，端到端网络切片技术难以应用；另一方面，网络资源与功能本身复杂多样，现有标准实现较为繁琐。如何更具可实现性地将虚拟网络功能与资源纳入切片技术的切实可控制范畴并实现切片仍存在一定挑战。

在本 5G 逐用户切片系统所面向的云渲染应用通信网络中，将对源流媒体服务器与渲染服务器的算力渲染资源以网络功能的形式进行封装。进一步地，系统将基于此针对云渲染应用的关键可控资源，对其进行切片，对网络中可控资源实现相对简化的 5G 切片技术，将网络资源与服务进行整合编排，从而面向用户应用需求进行高效快速地精准切片。

## 2.3 系统需求分析

根据以上对云渲染应用本身特性的分析，为将 5G 切片技术应用到以云渲染为典例的终端应用中，系统需涵盖原有的终端——网络——渲染服务器通信架构，感知云渲染应用本身特性及网络状态，并具有相应切片机制的发起与管理能力。

下面依次对系统涉及部分，需求功能及运行方式进行简要分析：

### (1) 系统涉及范围：

系统需包括终端设备、5G 无线接入网、由传输网与核心网组成的网络、由边缘服务器和数据中心服务器组成的渲染服务器、切片策略服务器等。

### (2) 系统需求功能：

系统应检测用户的网络质量，感知用户需求，并可触发建立 5G 切片，对此进行可控网络资源上的切片决策与部署实施，而后对切片的生命周期进行管理。

### (3) 系统运行方式：

系统应在云渲染类型应用启动后伴随运行，不需要独立的 UI 界面或是接收用户的相应操作。在整个云渲染应用的运行过程中，系统需通过切片技术保障云渲染应用网络质量，并在这一过程中保证用户无感。

## 2.4 可行性分析及方案论证

为满足系统的上述需求，贴近真实云渲染终端应用场景，使系统具有更强的实际应用价值，确认目前 5G 切片技术实际可应用情况，下面将针对本系统在真实 5G 网络环境中的实现，进行功能可行性分析及方案论证，并依次对终端实现工具，服务器侧实现工具和策略服务器侧实现工具进行分析与选择。



### 2.4.1 终端可行性及实现工具

针对云渲染类运行应用，本课题选取手机作为移动终端设备。在手机终端中，Android 系统已对 5G 网络具有一定支持能力。考虑到 Android 后续版本对网络质量探针接口的更新与废弃频率，为延长系统使用生命周期，降低系统受操作系统版本更替的影响，本系统在终端选取最新的 Android 12 版本的操作系统进行相应开发实现，并尽可能贴近真实应用环境。

Android 具有封装较为完善的网络质量探针以及应用层数据接口，可以获取网络连接方式，无线侧信号强度，预估第一跳带宽等网络数据。此外，Android 对手机流量亦有相关类的实现，可以获取当前手机上下行流量速率，并借助 icmp 报文侦测时延与抖动。

在 Google 于 2021 年 10 月发布的 Android 12 版本操作系统中，Android 官方声称支持了 5G 切片功能。然而经过详尽的调研与测试，可发现，Android 仅在企业对员工的工作设备进行管理这一应用场景提供了一种切片支持，需在 NetworkCapabilities 类中设定 NET\_CAPABILITY\_ENTERPRISE 以表示切片，并借助 Android Enterprise 方面提供的企业移动管理在企业端进行管控工作文件。

而对 5G 标准中的带宽，时延等网络切片设定，Android 于 NetworkCapabilities 类中设定了两个整型标识：NET\_CAPABILITY\_PRIORITIZE\_LATENCY 和 NET\_CAPABILITY\_PRIORITIZE\_BANDWIDTH，可对网络应用的需求特性进行设置，但该功能仍在开发中，计划在 Android 13 版本中推出，无法应用到本 5G 逐用户切片系统的开发中。另一方面，安卓官方并未给出其标识对应 5G 切片的使用流程与相应网络设备对此标识的资源分配支持情况，因此，目前将计划采用有传输预先协定的相应字符串数据的方式建立切片，并在系统中涵盖网络资源与网元的分配调度。

综上所述，本系统将在 Android 12 操作系统中采用 Java 语言实现终端侧功能，并借助相应 socket 方法传递切片建立请求数据。

### 2.4.2 服务器可行性及实现工具

在本系统中，服务器将架设于电脑设备的 Linux 操作系统下，采用 Python 实现相应 socket 通信功能。对云渲染音视频数据的传输模拟，可选取适当大小的音视频文件进行多次传输，使终端模拟运行云渲染应用。

### 2.4.3 切片策略服务器可行性及实现工具

在切片策略服务器中，受限于真实通信环境中基站，有线网的开放情况，本系统主要针对 5G 终端及 5G 网络应用特性，5G 流量特性，对可控的服务器通信资源进行网络切片的决策与部署。

具体而言，策略服务器可借助流量控制（Traffic Control）工具，对 Linux 下实现的

服务器的网卡通信带宽与优先级进行设置,切分,从而完成切片策略控制的效果。Traffic Control 是 Linux 内核提供的流量限速、整形和策略控制机制。它通过在输出端口处建立一个队列,以 qdisc-class-filter 的树形结构来实现对流量的分层控制。其中,QDisc 是 queueing discipline 的缩写,设置接口的队列规则。class 是在 QDisc 基础上设置的更深入排队规则,为进入队列的数据包进行排队。Filter 是过滤器,主要对数据包进行标记分类,决定它们将依据何种 QDisc 与 class 进入队列中。

采用流量控制工具进行网卡带宽设置时,需首先对网络物理设备(如以太网卡)绑定一个队列 QDisc,而后在绑定的 QDisc 队列上建立分类 class,并针对每一分类 class,设置相对应的基于路由的过滤器 filter,最后与过滤器相配合地建立特定路由表。

综上所述,通过 Traffic Control 这一工具,切片策略服务器具有控制服务器端网络资源,编排实现切片的能力。

### 3 面向终端的 5G 逐用户切片系统设计

本节基于系统需求分析结果，可行性分析与方案论证中对工具的选择分析，针对 5G 切片技术应用到云渲染终端应用的几大问题，设计了一种面向终端的 5G 逐用户切片系统，提供了使用此系统对云渲染终端用户提供 5G 切片服务的一种方法流程。

#### 3.1 系统总体设计

##### 3.1.1 系统架构设计

基于前文制定方案，为针对用户需求高效进行切片决策，实时感知网络状态，准确评估用户体验，本文设计了一种面向移动终端用户的 5G 逐用户切片系统，该系统涵盖终端，网络资源与网络功能构成的网络与服务器及切片策略服务器三大部分，其拓扑图如图3.1所示。

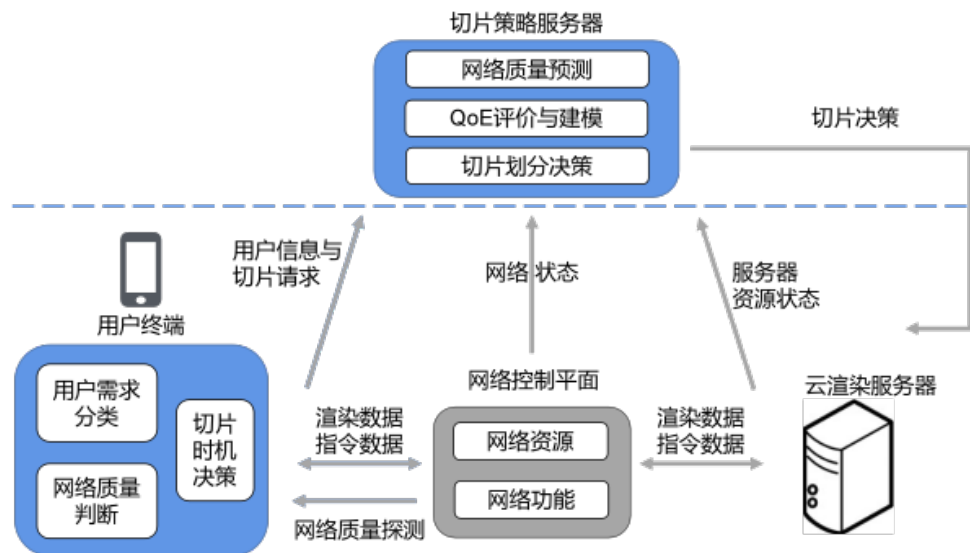


图 3.1: 面向终端的 5G 逐用户切片系统拓扑图

此系统主要由以下三部分组成：

##### (1) 终端部分：

该部分的实体设备是运行云渲染应用的各类终端，包括电脑设备 python 客户端，移动设备 5G 终端等。

在整个系统中，各移动用户的终端会保持与服务器进行音视频数据与控制数据的传输，同时，基于一定的网络状态探测功能和切片协议，终端会在需要建立切片时向切片策略服务器发送相应切片请求。

##### (2) 网络及服务器部分：

该部分的实体设备由网络设施与服务器组成。本系统设计的云渲染通信网络包括了 5G 接入网，传输网和核心网，及网络层中控制平面基于 SDN 与 NFV 技术封装的网络功能与网络资源。服务器包括了云渲染应用商的数据中心及边缘服务器，并以网络功能的形式封装。

在本系统的实现中，基于资源与功能的可切片操作情况，初步以服务器网卡通信资源作为可切片部分，向切片策略服务器提供相应感知与操纵接口，以验证提出切片框架和策略的有效性。

### (3) 切片策略服务器部分：

切片策略服务器将处理各移动终端的云渲染应用请求，感知网络与算力资源，及系统中的各切片情况，并以优化全局服务保障质量，提升系统整体资源利用率为目标，进行整个系统的资源编排决策，为各用户创建，接入相应 5G 网络切片，并管理各切片。

## 3.2 策略方法设计

在前述设计的系统上，结合 5G 切片技术的应用方法及云渲染应用的差异化需求，面向移动用户终端云渲染应用通信的实际场景，设计 5G 网络切片从触发，创建到生命周期管理的整体方法流程如图3.2所示：

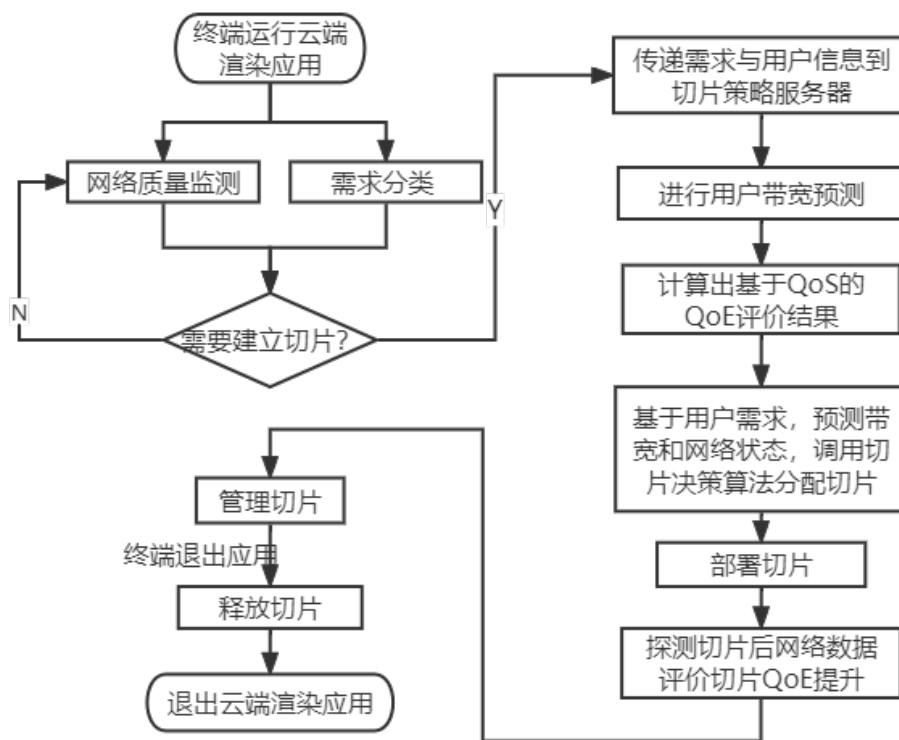


图 3.2: 5G 逐用户切片系统方法流程

本方法流程可分为以下六步：

- (1) 在终端侧对云渲染应用网络需求和移动用户网络质量进行采集，并分别进行用户需求量化分类与网络质量矩阵构建，并基于一定的切片触发策略，在需要建立切片时将相应结果上传至切片策略服务器。
- (2) 获取网络侧的网络整体服务资源状态报告，感知网络中云渲染所需网络资源与功能可用情况、已存在的切片实例使用情况及相应表现，并上传至切片策略服务器。
- (3) 在切片策略服务器中，基于网络质量预测和 QoE 评估模块，对个用户的终端侧数据进行网络带宽质量预测与 QoE 评估，得到更全面的，具有预测性的用户网络质量。
- (4) 基于所有用户的网络质量评价结果、请求切片类型、系统当前全局可支配资源，以保障网络中用户网络服务质量，提升系统整体资源利用率为目标，进行算法决策，确认各用户的切片请求允许与否及其实际分配资源类型。
- (5) 将切片策略服务器中做出的策略下发部署到网络与终端中，借助实现工具对资源进行编排，实例化切片；
- (6) 根据移动用户的云端渲染应用运行生命周期与网络质量探针获取的切片服务用户网络数据变化情况，给出切片后的 QoE 网络质量评价，从而评价切片效果。

### 3.3 功能模块设计

在 3.2.1 节系统设计的基础上，结合本系统的应用方法流程，对终端，网络与服务器和策略服务器三大系统组成部分进行模块划分与设计，分别给出其具体功能设计如下。

#### 3.3.1 终端模块设计

为使 5G 切片对终端应用的网络服务资源更具有针对性，建立更具有实时性，终端模块需要承担起量化用户需求，检测网络质量，保持与服务器的应用通信，并基于一定的策略触发向策略服务器请求切片的功能。具体而言，可分为以下四个部分：

##### (1) 用户需求建模与分类

该模块实现了用户需求测量与量化描述功能。系统将基于获取的云渲染应用类型，结合预先设定的对应类型需求网络指标上下界进行量化，再结合移动用户设备能力与用户需求设置进行调整，量化得到整体用户需求离散值（亦可称为 QoE 保障等级），作为需求分类结果。

##### (2) 网络质量探针

该模块将借助移动终端为应用层提供的网络测量接口，利用相应网络质量探针进行较为全面的网络通信质量的测量与描述，并以矩阵的形式存储，作为用户的网络质量描述。

### (3) 网络通信

该模块将借助 socket 通信机制，保持与云渲染服务器的长时间数据传输，进行带宽资源划分与优先级设定，精准控制各优先级下的带宽速率。在实际商业应用场景中，该模块对应云渲染应用部分，将接收到的视频帧渲染并显示。

### (4) 切片请求准入判断

针对实际应用场景中的海量移动终端用户，为减少切片策略服务器所需决策数据量，提高系统的整体利用率与服务保障率，终端应对切片的触发请求进行一定的控制与筛选，在网络质量已较为优越，足以满足云渲染等应用运行需求的情况下不发起资源切片请求，仅当网络质量不满足时触发切片。

为此，系统将基于网络质量探针给出的描述，与用户需求离散化的整型值所对应的需求进行综合判断，当一维或多维网络质量数据低于预设定的用户此类需求指标时，将向切片策略服务器发起切片请求。

进一步地，为确保切片策略服务器能准确感知各终端情况和用户需求，传递的切片请求中应包含用户信息，需求类型，网络质量等数据，并与策略服务器约定相应协议，确保数据的有效传递。

## 3.3.2 网络与服务器模块设计

在网络与服务器部分，系统将感知控制平面层的网络资源，网络功能以及网络切片实例情况，对其进行建模描述，上报于切片策略服务器，为切片策略服务器提供决策动作空间，解决切片网络资源与服务封装问题。其功能模块划分为以下部分：

### (1) 网络测量报告

该模块实现了网络测量和切片感知的功能。具体而言，系统将获取网络中云渲染应用所需的网络资源，网络服务功能并进行表示、获取网络中存在切片实例容量与使用情况并进行表示。通过相应的接口和预定通信协议，切片策略服务器可快速获取网络资源情况与切片使用情况。在本文中，基于网络资源和服务器资源的可控情况，主要采用服务器层的网卡出端口带宽作为整体网络及服务器资源，并作为切片策略所分配的资源。

### (2) 切片部署

该模块将切片策略服务器下发的需求指令，感知系统可使用的网络资源或已创建的切片实例，选择合适的切片实例资源与相应应用服务进行绑定。若已有类型且有空载的资源切片实例，则可将相应需求接入；若暂无适合切片实例，则调用已有的资源管理接口创建新的切片实例并接入用户，从而完成切片的部署。

本文将采取 Linux 系统下的 Traffic Control 工具进行流量管控，从而完成云渲染服务器网卡通信资源的调度，达成控制网络与服务器资源形成切片的功能。

### 3.3.3 策略服务器模块设计

切片策略服务器是本系统的核心部分。在策略服务器中，系统需要感知网络资源与切片情况，处理大量终端发起的切片需求，改善网络质量不佳的用户体验，提高系统资源的利用率。其功能模块可划分为以下部分：

#### (1) 网络测量接收

该模块主要针对终端模块、网络与服务器模块上传的数据进行接收与存储，为决策算法提供需求数据与可分配数据。

#### (2) 网络质量预测

为使切片系统能提前预报用户可能出现的网络带宽拥塞，从而预先进行相应切片处理，提前回避网络质量下降，解决从网络质量下降到切片建立的一段时间的较差体验问题，该模块借助时序回归算法，对用户带宽流量数据进行时序建模，本身得到历史带宽模型。

由于云渲染类应用本身带宽时间序列不具有明显趋势且不含有季节性，可采用自回归滑动平均模型（Auto-Regressive and Moving Average）作为时序回归算法模型。在此时序模型基础上，预测短时间内下一时间段的网络带宽，从而监测是否出现带宽下降，对未来一段时间内的质量进行评估，辅助切片决策算法。

#### (3) 用户 QoE 评价

该模块将对用户应用服务体验给出归一化的间接评价。考虑到 QoE 主要由服务层面（包括网络状态），环境层面（即运行运用终端实际环境）及用户层面（期望，身心状态等）三维度影响因素决定，而本系统主要目的是提升网络服务层面对 QoE 的保障能力，因此，系统需可针对应用网络质量参数，基于 QoS 与 QoE 的关联模型，对用户网络 QoE 进行评价。

视频 QoE 的评价方法分为主观评价和客观评价两种。主观评价方法有：平均意见分法（MOS），失真平均意见分法（DMOS），单刺激连续质量标度方法（SSCQS），双刺激连续质量标度（DSCQS）等。主观评价方法的优点是准确和易于理解，缺点是实施条件苛刻，步骤复杂，难以在实验室的环境中实施。

在对 QoE 进行主观评测时，一般使用 MOS（Mean Opinion Score）值来衡量体验质量的好坏。MOS 值分为 5 个等级，分别为 1 至 5，其含义如表 3.1 所示。

表 3.1: MOS 值含义

MOS 值	含义
5	非常好
4	好
3	一般
2	差
1	非常差

基于网络视频领域的相关 QoS 与 MoS 映射关系研究与实验，可将丢包率、抖动对 QoE 的影响视为相互独立的函数。设丢包率对视音频 QoE 造成的损伤为  $drop$ ，抖动对视音频 QoE 造成的损伤为  $jitter$ 。考虑到在不同的抖动值下，丢包率对 QoE 造成的损伤是不同的；同样，在不同的丢包率下，抖动对 QoE 造成的损伤也是不同的。因此，QoS 到 QoE 的映射关系如公式 3.1 所示。

$$MOS = a - b * (1.8 * jitter + delay) - c * droprate \quad (3.1)$$

其中，使用 MOS 对 QoE 进行定量的度量。 $a$  表示 MOS 值上限， $b$  和  $c$  为待定常数。 $delay$  表示时延， $drop$  和  $jitter$  分别表示丢包率、抖动对 QoE 造成的影响。

由于上述公式及论文研究主要针对 IPTV 等网络视频领域，我们将其结合今年来各大应用厂商和国家广播电视总局中白皮书云渲染相关应用需求，考虑到云渲染应用本身特性，带宽、时延、抖动等均对云游戏等云渲染应用影响重大，故设置其时延抖动丢包率的 QoS 到 QoE 映射关系如公式 3.2 所示。

$$MOS_d = 4.995 - 0.008 * (1.8 * jitter + delay) - 0.231 * droprate \quad (3.2)$$

具体而言，考虑到云渲染应用对网络波动的容忍程度随应用类型变化，用户设置不同而存在一定差异，其渲染数据流量大，对时延抖动丢包率均敏感。因此可综合其带宽，时延，抖动，丢包率数据，可准确反应应用当前渲染数据能否及时，稳定地传输到终端，保障用户 QoE。基于此，策略服务器将根据用户终端上传的网络测量数据，对时延，带宽，抖动，丢包率四个维度进行计算，通过 QoE 间接评价公式得出用户当前网络 QoE 计算结果，评价用户服务体验。

另一方面，云渲染应用对带宽的需求随应用类型变化，用户设置不同而存在一定差异，其渲染数据流量大，带宽需求高，且更多的在当前切片可控资源与功能中呈现，因



此引入带宽对 MoS 的影响公式，为尽量保证策略使应用带宽落在符合其需求适当区间内，给出带宽对 MoS 的影响如公式3.3所示：

$$MoS_{band} = \frac{5}{1 + e^{k_1 * (b - k_2)}} \quad (3.3)$$

综上，综合其带宽，时延，抖动，丢包率数据，可准确反应当前网络服务质量对应用需求的保障程度。为此，策略服务器将根据用户终端上传的网络测量数据，对时延，带宽，抖动，丢包率四个维度进行计算，通过 QoE 间接评价公式，将用户网络 QoS 指标转化成相应 MoS 评分。基于此，我们对各个网络质量参数影响的 QoE 进行加权求和，统构造一 QoE 评估模型，如公式3.4所示：

$$QoE = k_3 * bqoe + k_4 * djpqoe \quad (3.4)$$

根据前文中不同用户运行云渲染应用类型对网络需求的侧重不同，对时延、带宽等的敏感程度不同， $k_1, k_2, k_3, k_4$  四个参数的取值可进行不同的表述。此外，由于不同需求用户的 QoE 计算结果数量级难以统一，为保证各个用户的 QoE 评价结果权重无差距，我们基于获取数据的码率分布和白皮书等参考设计 QoE 评价公式中的各个参数，以确保 QoE 的评价结果能针对用户当前需求衡量网络质量是否满足，且我们通过调整参数保证评价结果均会落在  $[0, 5]$  区间中。

#### (4) 切片策略

##### a. 整体架构

决策模型主要分为两个部分：用户的需求预测 + 基于强化学习框架的部署。我们基于用户过往的网络质量和历史带宽，使用自回归滑动平均模型为用户的需求带宽做出合理的预估。并基于该预测以及用户的需求类型和当前 qoe 进行合理的资源分配。决策模型的示意图如图3.3所示。

经过多轮学习之后，我们的模型就能在实验场景下做出合理的切片决策。然后使用预先训练好的模型，模型会根据  $n$  个应用请求以及所处的网络环境进行网络切片实例的分配，从而达到总体的最佳用户体验。

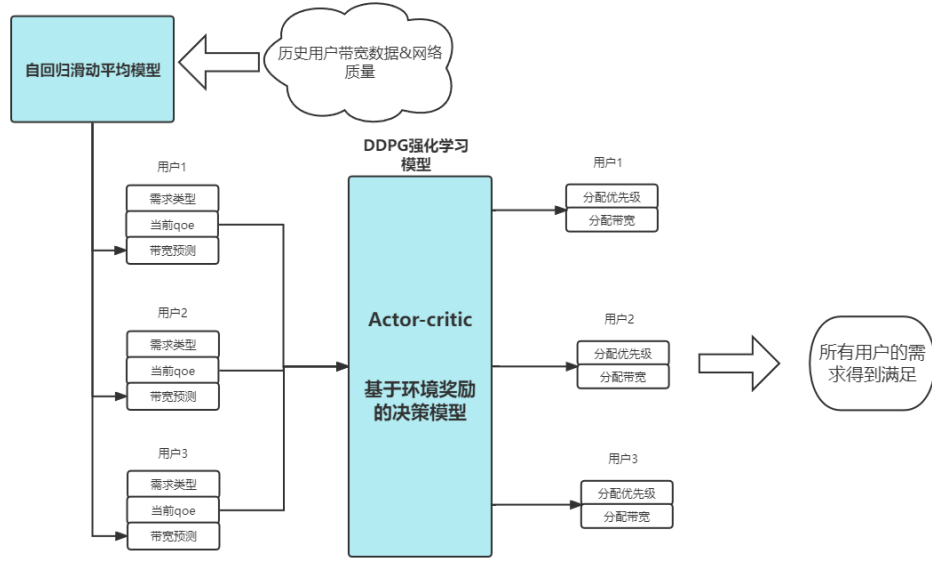


图 3.3: 决策模型示意图

在终端侧，采用较为全面的切片触发筛选机制，减轻策略服务器需处理的终端切片请求数量；在策略服务器侧，首先采用预测算法和 QoE 评价算法对网络质量数据进行进一步预处理，并基于预测和 QoE 计算结果，对更需要切片的用户请求进行筛选；而后基于用户需求和网络资源及切片实例，采用深度学习模型进行切片决策，尽可能提升用户移动应用网络质量体验。

## b. 相关定义

该模块将基于 QoE 指标和网络资源情况构建多目标优化决策，编排网络资源形成切片。在对终端数据，网络与服务器数据进行感知和处理，得到各用户 QoE 结果，需求类型，可使用网络资源和当前网络中切片实例后，模块以优化整体用户 QoE 评价结果，全局网络资源与功能利用率为优化目标进行决策。本文中，切片策略以多层切片请求准入筛选机制结合深度学习模型的形式实现。

由于系统为用户分配相近的资源情况时，用户将得到相近的网络应用体验，我们将其作为策略依据，用公式3.5描述如下：

$$\|QoE_{R1} - QoE_{R2}\| \leq L \|R1 - R2\|^\alpha \quad (3.5)$$

我们从带宽、时延、抖动、丢包率四个方面描述各用户的需求，如公式3.6所示：

$$Q_m = \{d_{max}, j_{max}, l_{max}, b_{min}\}, \in [1, 4] \quad (3.6)$$

其中第  $m$  类用户的需求为  $Q_m$ ，用户数量为  $n_m$  个。

引入  $b_{max}$  作为足以优质满足该类应用需求的带宽，视作某种上界，则具体约束用公式3.7描述为：

$$\begin{cases} Q_m \cdot b_{min} \leq B_{mi} & \forall i \in [1, n_m], m \in [1, 4] \\ 0 \leq D_{mi} \leq Q_m \cdot d_{max} & \forall i \in [1, n_m], m \in [1, 4] \\ 0 \leq J_{mi} \leq Q_m \cdot j_{max} & \forall i \in [1, n_m], m \in [1, 4] \\ 0 \leq L_{mi} \leq Q_m \cdot l_{max} & \forall i \in [1, n_m], m \in [1, 4] \end{cases} \quad (3.7)$$

### c. 强化学习策略

切片的具体策略为：连接  $n$  台设备，每台设备（应用）有自己的需求（req），切片策略服务器会根据这个需求来进行切片的初始化。

当检测到网络拥塞或者平均  $qoe$  指标较低时，说明当前的切片策略不能满足该场景下各用户应用的需求了，我们需要启用切片策略来对当前的各用户带宽与优先级标签进行重新的分配和调整。

为了得到最佳的切片策略的分类结果，我们综合  $n$  个用户的各自需求和当前网络的资源，为每一个用户做出合理的网络资源分配从而使每个用户的需求都能得到满足。特别的，我们这里把网络资源的分配具象化为每一个用户分配一个网络优先级  $priority$  和一定量的带宽资源  $bandwidth$ 。

由于网络状况和质量的多变性，以及用户数量的多变性，我们手动地为各种网络场景和不同需求产生测试集并做标签是一个很难完成的任务。所以我们在这里考虑使用无监督学习的算法-强化学习。

我们使用 *DeepDeterministicPolicyGradient* 算法，可以做出连续性的动作决策，对于我们的切片来说，*DDPG* 算法可以通过环境来学习出满足用户需要的一个具体的带宽值。*DDPG* 算法示意图如图3.4所示：

我们首先定义  $n$  个用户，*DDPG* 算法观测到的环境状态为当前每个用户的需求类型、 $qoe$  值以及通过预测模型预测到的该用户可能使用到的带宽，*action* 为对某个用户分配的资源优先级和带宽值。奖励  $reward$  为  $n$  个用户的  $qoe$  平均值，其中为了让决策模型更加兼顾全局的效果，我们使用一个超参数  $\lambda$ ，对于已经实施了分配的用户的  $qoe$  值做额外的奖励。奖励  $reward$  的计算公式如公式3.8所示：

$$reward = \begin{cases} qoe * (1 + \lambda) \\ qoe \end{cases} \quad (3.8)$$

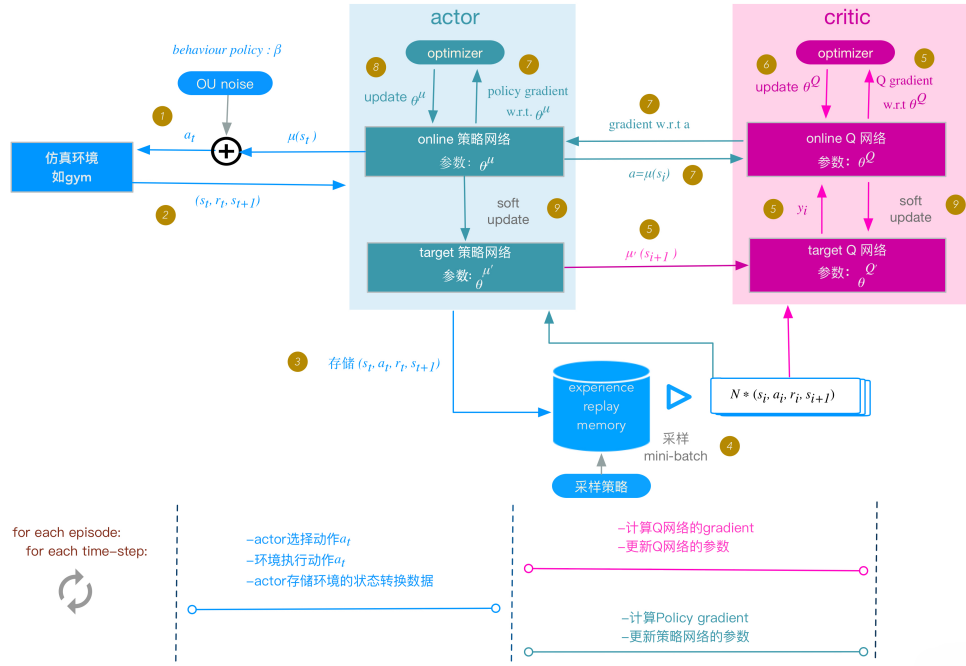


图 3.4: DDPG 算法示意图

#### (5) 策略下发

该模块实现了将切片决策传递给网络与服务器部分的功能。具体而言，策略服务器会借助网络与服务器侧提供的函数，依据预选设定的通信内容协议，将用户标识和对应切片类型下发，从而在网络与服务侧将切片实例与相应用户标识（如用户的 ipv4 地址）绑定，实现资源的切片划分。

## 4 面向终端的 5G 逐用户切片系统实现

本节将基于前文中设计的面向移动终端的 5G 逐用户切片系统，以云游戏为应用典例，给出云渲染应用场景下的一种系统实现方案，从终端，网络及服务器和策略服务器三个部分出发，对系统设计内容进行实现。

### 4.1 终端功能模块实现

在实现系统终端侧功能时，本课题主要采用 Android Studio 2021.1.1 开发工具进行实现，语言采用 java 1.8 版本。

值得一提的是，我们同样在 Python3.6 中实现了具有相同功能（包括且不限于网络质量监测，音视频通信，切片协议通信等功能）的 python 客户端，用于弥补实验手机数量相对不足的问题。

#### 4.1.1 网络质量监测模块实现

本模块基于 Android 提供的网络层接口，实现了监测网络连接方式；监听无线信号强度，无线信号功率，基站预估上下行带宽功能；测量终端实时网速的功能；测量与指定服务器的通信时延，丢包率并计算抖动的功能。下面针对该模块涉及实现的类进行详细阐述

（1）NetTypeListener 类。

该类负责监测网络连接方式，需继承 ConnectivityManager 的 NetworkCallback 子类。UML 如图4.1所示：

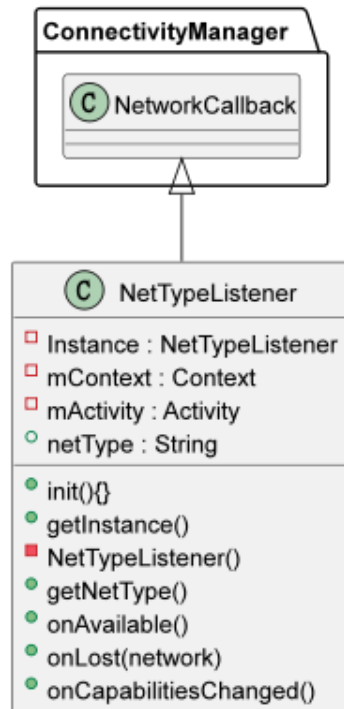


图 4.1: 网络连接方式类 UML 图

在此类中，本系统重写了 `onAvailable`, `onLost` 和 `onCapabilitiesChanged` 三个函数。以上三个函数以安卓本身抽象的 `Network` 类型变量作为参数，分别在网络连接到新的指定网络，网络断开连接以及网络能力发生变化时被调用。

对前两个函数，选择在控制台或日志进行相应提示信息的输出即可，此处略去具体内容。

对 `onCapabilitiesChanged` 而言，需对当前网络的网络能力进行判断，即对类型为 `NetworkCapabilities` 的变量 `networkCapabilities`，数次调用网络能力类实现的函数进行判断。

具体而言，首先通过 `hasCapability` 函数判断当前网络连接是否能正常上网，即 `networkCapabilities` 是否具有 `NET_CAPABILITY_VALIDATED` 属性。

若能正常上网，则调用 `hasTransport` 函数判断当前网络是 Wifi 连接还是移动数据连接，即网络能力具有 `TRANSPORT_WIFI` 属性还是 `TRANSPORT_CELLULAR` 属性。

而针对移动数据连接的情况，需要基于 Android 的当前 context，获取 `TelephonyManager` 类变量 `telephonyManager`，并通过封装的 `getDataNetworkType` 函数获得相应网络类型的整型变量。值得注意的是，该函数调用需要 `READ_PHONE_STATE` 这一权限，而该权限在较新版本的 Android 系统中属于高危权限，需要借助 `ActivityCompat.requestPermissions` 函数进行动态申请。

在获取该整型变量后，通过与 `TelephonyManager` 类中的整型常量比较，即可判断

移动网络的 2-4G 类型。

在实现 NetTypeListener 类，并实例化该类型变量 ntl 后，只需获取当前应用的 ConnectivityManager 网络连接管理器，并调用 registerNetworkCallback 函数传入 ntl 参数即可完成注册工作。

## (2) SigStrengthListener 类

该类具有监听无线信号强度功能，实现与 NetTypeListener 类类似，需继承 TelephonyCallback 类并实现该类的接口 SignalStrengthsListener。其 UML 图如图4.2所示：

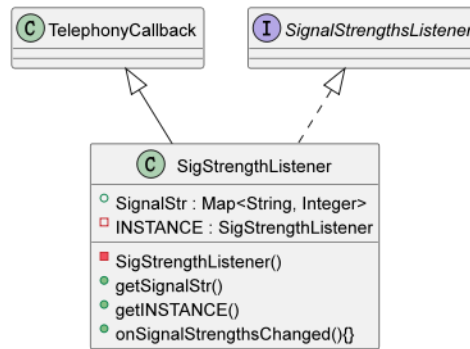


图 4.2: 无线信号强度监听类 UML 图

类中主要对 onSignalStrengthsChanged 方法进行了重写，其参数为 SignalStrength 类型的变量 ss。在该方法中，将通过类型的 getCellSignalStrengths 函数，传入指定参数 CellSignalStrengthNr.class，获取相应的 5G 基站信号强度列表。

通过对该列表进行操作，获取其首位的当前连接基站，可以获得类型为 CellSignalStrengthNr 的变量 css，并通过该遍历获取无线信号强度 dbm 和无线信号功率 Rsrq。另一方面，在类中监测到无线信号强度较差这一监听事件时，在实现该类并实例化变量 ssl 后，需借助当前应用的移动管理器 TelephonyManager，调用 registerTelephonyCallback 函数，以新开线程的方式实现监听注册。

## (3) NetWorkMatrixUtils 类

该类将用于获取当前应用的带宽速率，到服务器的时延，抖动和丢包率数据，构成网络矩阵信息。

带宽测量借助 Android 提供的 TrafficStats 类在 showNetSpeed 方法中进行实现。该类在网络层测量流量并存储器统计总量。因此，只需在类中存储上一时刻的数据包量与时刻，计算出当前数据包和上一时刻差值即可获取带宽速率。

对时延,抖动,丢包率的测量则通过 detect 函数实现。借助 Runtime.getRuntime().exec 函数，以 ping 指令的形式对服务器 ip 进行 icmp 报文探测，从而实时获得时延与丢包率，并计算出抖动。

通过 java 的 Timer 类定时器，设置每秒运行一次 showNetSpeed 与 detect 方法，并存储在 ArrayDeque 双端队列中，即可完成网速矩阵的测量与存储。其 UML 图如图4.3所示：

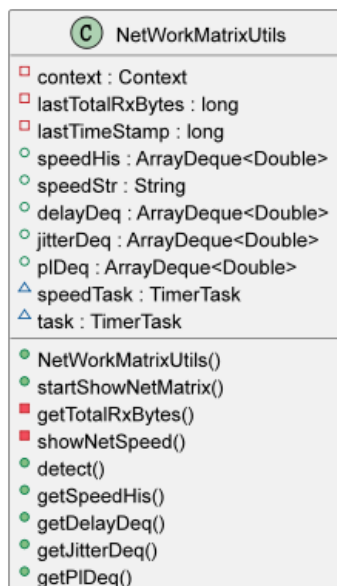


图 4.3: 网络矩阵工具类 UML 图

#### (4) NetworkUtil 类

该类是对以上三个类进行了合适的封装，从而使终端的主函数中可以统一化的使用各探测功能，并实现了测量指定服务器的通信时延与丢包率、获取基站上下行理论带宽的功能。其 UML 图如图4.4所示：

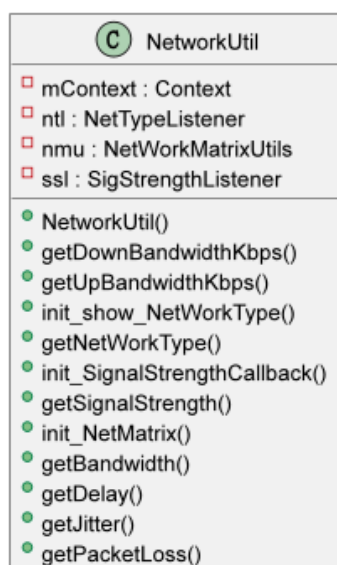


图 4.4: 网络工具类 UML 图



首先,类中实现了 `init_show_NetWorkType` 方法,在该方法中获取应用当前的 `ConnectivityManager` 类型变量,初始化了 `NetTypeListener` 并获取其单例实例 `ntl`,注册网络连接方式的监听。并在此基础上借助 `ntl` 实现了获取网络连接方式的 `getNetWorkType` 方法。

其次,该类实现了 `init_SignalStrengthCallback` 方法,在该方法中获取应用当前的 `TelephonyManager` 类型变量,并获取 `SigStrengthListener` 实例 `ssl`,调用相应方法对进行注册监听。并在此基础上借助 `ssl` 实现了获取网络连接方式的 `getSignalStrength` 方法。

最后,类中实现了 `init_NetMatrix` 方法,获取 `NetWorkMatrixUtils` 类型变量 `nmu`,并借此提供了带宽,时延,抖动与丢包率的获取方法。

此外,该类实现了 `getDownBandwidthKbps` 和 `getUpBandwidthKbps` 方法,分别借助 `networkCapabilities` 的相应函数获取终端与当前基站的理论上下行带宽。

#### 4.1.2 用户需求分类模块实现

将应用类型需求作为用户需求的基准,则可先针对云渲染云游戏应用类型,基于 2.1 节分析,从带宽,时延,抖动与丢包率四个维度评价业务的网络需求指标,并将需求映射到单个整型变量中。

具体而言,在类 `SimUser` 中首先获取云渲染应用类型需求。与表 2-2 相对应,定义了四大静态常量标识应用需求类型,分别对应默认需求、高带宽需求、低时延需求、高带宽,低时延且低抖动需求。在此基础上,将表示应用类型的字符串一一映射到相应需求中。

而后,在 `get_req_model` 方法中,采集包括终端设备型号,分辨率,帧率,码率设置等需求,基于图2.3和图2.4,对应用需求进行进一步准确调整。如:帧率分辨率过低时,则可相应降低应用需求对带宽的要求等级。若帧率分辨率较高,则需提升需求的带宽等级。最终得到调整后分类结果,即用户需求。

通过以上方式,可以实现综合考虑应用特性需求,终端设备实际性能与用户设置,对用户需求进行全面量化表示的功能,指导切片决策时的资源分配。其 UML 图如图4.5所示:

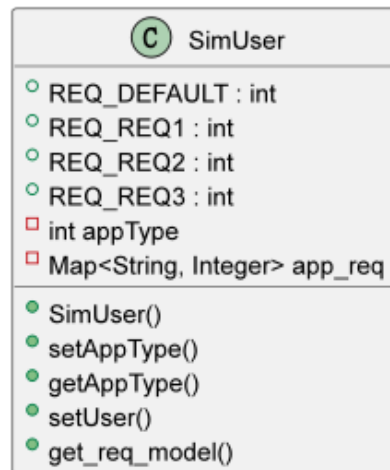


图 4.5: SimUserGame 类 UML 图

#### 4.1.3 切片触发模块及协议实现

在本系统中，在主类 NetSlice 中实现了 `judge_slice_build` 函数和 `judgeDeq` 函数，用于判断是否触发切片。

`judgeDeq` 函数将 `ArrayDeque` 类型的变量作为需要判断的列表参数，将整型 `standard` 变量作为其标准值。通过遍历列表数据，与标准值比较，可统计判断出低于标准或远低于标准的数据数量，并基于网络不佳的数据比例进行真假值的返回，给出该项网络参数是否满足对应标准需求的判断。

在 `judge_slice_build` 函数中，其参数有包括带宽、时延、抖动、丢包率数据的网络矩阵数据 `NetData`，无线信号强度数据 `SigData` 和用户需求类型结果 `req`。其返回值为是否建立切片的 `boolean` 值。

该函数将本方法根据 `req` 的取值，设置相应用户需求应有网络质量值，并对网络质量矩阵中相应网络数据进行判断。例如，对高带宽用户需求，调用 `judgeDeq` 函数时主要针对带宽数据；对低时延应用则针对时延抖动和丢包，设置 `judgeDeq` 的时延标准为 100ms，并依据判断值给出返回结果。

实现以上方法后，在正式应用时，首先需检测网络连接方式。若是 5G 网络连接方式，可借助定时器机制实现定时监测，在定时器线程中每分钟进行一次是否触发建立切片的网络质量判断。当 `judge_slice_build` 方法返回值为真时触发建立切片。

对所传递的建立切片信息，采用 `hashmap` 存储用户网络需求类型，网络质量，上报信息类型（是发起切片还是更新网络质量）及其余可扩展信息。借助通信模块向本函数提供的发送数据方法进行传输。

#### 4.1.4 通信模块实现

该模块主要借助已有的 java 网络层 socket 工具，通过定义一个实现 Runnable 接口的 ConnectSer 类，重写 run 方法，对服务器与切片策略服务器对应 ip 进行 socket 绑定，而后通过相应 io 类进行输入输出处理即可。

### 4.2 服务器实现

服务器包括了切片策略服务器和云渲染服务器，分别承担做出与客户端通信并切片决策，和客户端保持视频按帧率逐帧传输模拟云渲染应用的功能。

本系统选择在 VMWare 15.5.6 虚拟机的 Ubuntu18.04 环境下，采用 python3 语言和 Traffic Control 命令脚本实现。

项目中包含有：切片策略服务器 sli\_fine.py、云渲染服务器 service.py、预测带宽模块 ARMA\_forecast.py、QoE 计算模块 QoE\_cal.py、结果可视化模块 pltres.py、工具函数模块 sliceUtil.py 和切片策略相关的 py 文件；切片工具初始化脚本 fine\_init\_tc.sh，切片部署脚本 fine\_slice.sh，切片调整脚本 fine\_change.sh，切片释放脚本 fine\_delFilter.sh 共五个 sh 文件。

对切片策略服务器而言，接收到各终端的切片请求时，切片策略整体处理流程如图4.6所示：

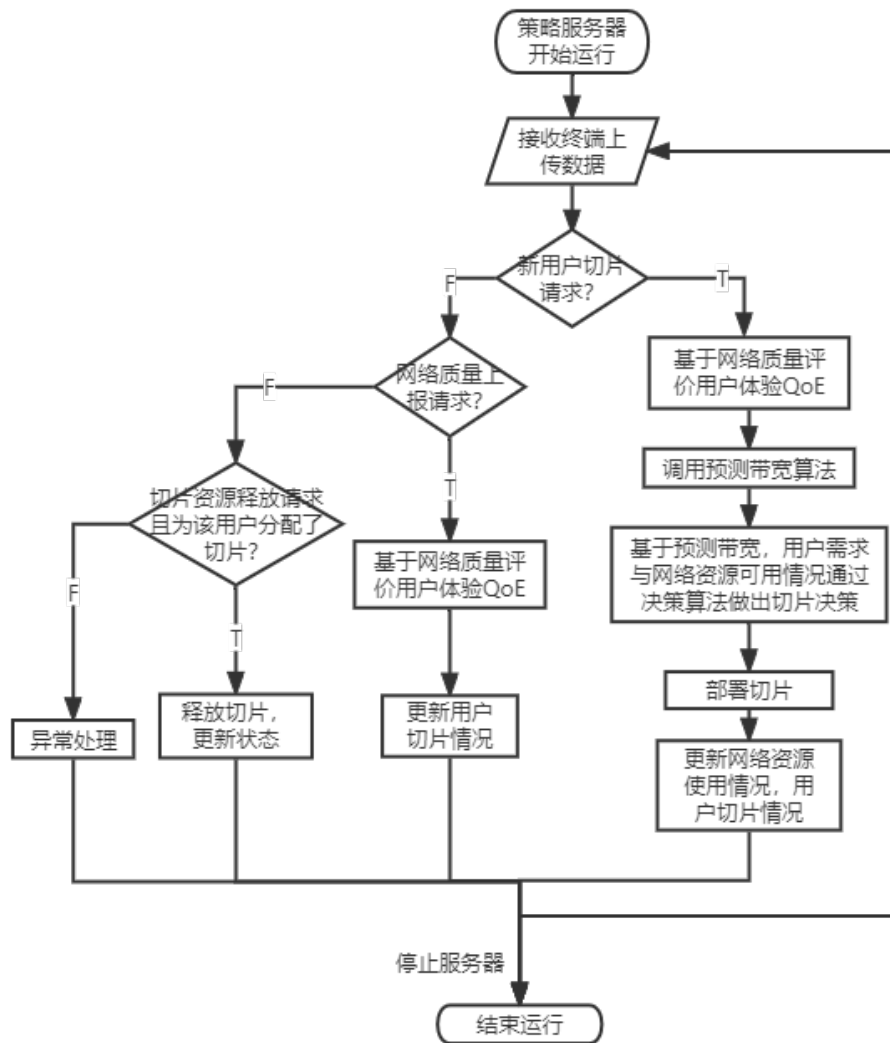


图 4.6: 切片策略服务器流程图

下面依据功能模块的划分，重点对切片策略服务器进行介绍。

#### 4.2.1 云渲染数据通信模块实现

在云渲染服务器中，采用依据帧率按帧传输视频文件的方式模拟云渲染数据的实时传输效果。具体而言，渲染服务器中需首先进行 socket 端口的初始化，以监听可能的 udp 客户端请求。当与终端建立通信后，可通过多线程与多个终端分别按帧率逐帧传递音视频，确保实验在无限靠近云渲染环境

其依据帧率逐帧传输的方法主要借助 opencv 的 python 版实现。

首先，通过 `get(cv2.CAP_PROP_FPS)` 读取视频帧率，开一个线程逐帧读取视频并存放在该用户的帧队列中，而在传输线程中从帧队列中读取，并在每发送视频帧率次文件后检查用时是否超过 1 秒，基于此调整读取，渲染发送速率。

#### 4.2.2 策略服务器通信模块实现

在 `sli_fine.py` 中, 首先进行 socket 端口的初始化。当接收到终端 socket 的连接请求后, 为该用户新建线程, 并调用处理报文的 `dealReq` 函数。处理报文时, 首先要对相应报文进行解析。系统在终端处将带宽, 时延, 抖动和丢包率数据, 需求类型和报文类型发送给策略服务器。在策略服务器收到数据后, 首先解析报文类型, 判断是切片请求, qoe 更新请求还是释放切片请求, 而后调用 `sliceUtil` 类中的 `parsemsg` 函数对报文解析, 形成带宽序列列表 `band`, 丢包率 `pack`, 时延 `delay`, 抖动 `jitter` 等。对切片请求, 将进入切片策略模块。对另外两种请求, 将更新系统变量记录值, 并对释放切片请求执行释放切片操作。

#### 4.2.3 评价与预测模块实现

该模块主要在 `QoE_cal.py` 和 `ARMA_forecast.py` 文件中实现, 主要对用户上传网络数据进行预测和综合评估处理, 从而更好的辅助切片决策。

##### (1) 预测带宽

在得到带宽列表 `band` 后, 调用 `ARMA_forecast` 文件中实现的 `forecast` 函数, 借助 `statsmodels` 库对应实现的 ARMA 实例化时序模型, 设置其 `p`, `q` 参数, 对历史带宽数据进行分析, 而后调用其内置实现的 `forecast` 函数, 传入指定长度的预测区间, 得到进一步的预测结果。在本系统的此种实现中, 设置其预测时间长度区间与传入带宽数据时间长度相同。

##### (2) QoE 评价指标

在终端中, 手机设备触发切片模块已对网络质量进行过判断。在策略服务器中, 为综合衡量用户的云渲染应用网络服务质量体验, 可借助第三节中 QoE 公式实现的 QoE 计算函数 `MosToQoE`, 将四项网络质量的列表和需求 `req` 作为参数, 并根据不同用户需求 (即是否是高带宽需求, 是否是低时延需求) 设置 QoE 公式各参数, 计算出历史网络质量的综合 QoE 结果。用于切片策略及切片效果评价。

#### 4.2.4 切片决策模块实现

本模块主要在 `sli_fine.py` 文件中进行决策, 并通过调用一系列 `sh` 文件中的 `tc` 命令对 linux 服务器的网卡带宽资源进行编排划分。

首先, 决策算法需要用户需求, 用户当前 qoe 评价, 用户带宽, 系统当前网络可切片资源与功能的使用情况等作为决策依据及动作空间。

在通信模块处理报文中, 评价与预测模块获取用户的预测带宽和 QoE 网络指标后, 算法基于此进行切片决策, 基于用户需求是否时延敏感设置优先级, 基于用户需求与预测带宽设置带宽资源, 以达到精准有效提升网络服务质量的的目的。

目前,策略服务器的一种策略主要依据前面的功能模块实现的基于用户需求和预测带宽的切片策略,即:基于历史经验数据,将预测带宽与用户需求结合,以用户需求为主设置对应 ip 数据流优先级,以处理后的预测带宽为主设置该 ip 可使用的网络带宽资源,并在此基础上基于用户需求对带宽进行适当调整。另一种是基于用户需求,预测带宽的 DDPG 强化学习算法,即:将前述方法的设置过程完全交由强化学习模型决定,通过训练及 qoe 反馈得到相关参数,令算法学习经验数据,并基于此对各用户切片请求进行处理。

### 4.3 服务器切片管理实现

服务器部署于与切片策略服务器相同的环境中,将与终端保持通信,并落实对切片策略服务器的决策。下面分别对其切片部署,切片调整和释放功能进行阐述。

注:考虑系统可控网络资源与网元功能。本系统实现中即服务器网卡资源设备,共有 100Mbps 带宽。

#### 4.3.1 切片部署实现

在实现时,本系统于 `fine_init_tc.sh` 脚本中运用 `traffic control` 命令进行带宽资源划分与优先级设定,创建父 class。

当切片决策模块为用户分配相应资源时,则为其分配 `flowid` 作为切片 id,传递切片划分的该用户数据流优先级 `prio`,分配带宽 `bandwidth` 和用户 ip 给 `fine_slice.sh`,创建子 class,并建立 filter 将发往对应 ip 的数据包标记为对应的类别,从而控制其出口带宽速率和优先级。

#### 4.3.2 切片调整与释放实现

当切片决策模块为用户调整或释放切片资源时,则基于策略服务器存储的变量信息,在策略服务器中读取其 `flowid` 与原先分配的资源情况,并连通新的分配资源决策,传递给 `fine_change.sh` 或 `fine_delFilter.sh`,以完成切片的调整或释放。

## 5 性能测试与分析

本章中针对前文设计实现的 5G 逐用户切片系统各组成部分进行了较为详尽的功能评测，包括终端侧的网络与无线信号监听功能、质量探测，服务器和策略服务器的通信与资源切片功能，并在此基础上设计了合适的实验方案，在拥塞网络环境下，对系统改善用户服务质量的性能进行了综合评测。

### 5.1 测试环境

下面分别给出终端和服务器的设备型号及测试运行环境：

表 5.1: 实验设备环境

型号/性能	终端设备	本地服务器	云服务器
设备型号	红米 K50 电竞版 小米 11 OPPO Find X5	虚拟机 VMware Workstation 15.5.6	阿里云云服务器 ECS
操作系统版本	Android 12	Ubuntu 18.04	Alibaba Cloud Linux release 3
内存	8GiB	5GiB	4GiB
处理器数	8	4	2
带宽	/	100Mbps	1Mbps

### 5.2 功能测试

下面依次对终端的网络各质量监听功能，终端和服务器的通信传输功能，切片策略服务器的切片功能进行测试。服务器部署于本地，其配置如表5.1所示。

#### 5.2.1 终端功能测试

##### 1. 网络连接监听测试：

首先将 5G 手机采用 Wifi 连接，而后断开，切换移动数据连接，观察 Android Studio 控制台输出结果如图5.1所示：

```
I/System.out: on Available 网络已连接
I/System.out: 移动数据连接中
I/System.out: on Available 网络已断开
I/System.out: on Available 网络已连接
I/System.out: Wifi连接中
I/System.out: Wifi连接中
```

图 5.1: 网络连接监听功能测试

借助以上功能，可以确认网络连接监听功能可以及时感知连接方式变化，确保系统运行于 5G 无线连接环境下。

## 2. 无线信号强度监听测试:

令终端保持对 5G 无线网络连接，较大幅度移动手机终端，观测控制台是否有输出相应 Change 提升信息，如图5.2所示:

```
: dbm: -92
: ssRsrp:-92
: ssRsrq:-12
: NetType: 5G
: UpBandWidth: 60000
: DownBandWidth: 145000
: SignalStrength: {ssRsrp=-92, ssRsrq=-12, dbm=-92}
: Signal Strength Change:
7483647 csiCqiTableIndex = 2147483647 csiCqiReport =
}]
: dbm: -87
: ssRsrp:-87
: ssRsrq:-12
: NetType: 5G
: UpBandWidth: 60000
: DownBandWidth: 145000
: SignalStrength: {ssRsrp=-87, ssRsrq=-12, dbm=-87}
```

图 5.2: 无线信号强度监听测试

可以看出，无线信号强度发生变化后，确实会触发相关监听函数，并且其新结果值与原有信号强度有一定变化差距。

## 3. 网络质量探测:

在手机终端探测渲染服务器的端到端时延与丢包率如图5.3所示:



```

In detect:PING 192.168.1.108 (192.168.1.108) 56(84) bytes of data.
In detect:64 bytes from 192.168.1.108: icmp_seq=1 ttl=64 time=142 ms
In detect:64 bytes from 192.168.1.108: icmp_seq=2 ttl=64 time=18.4 ms
In detect:64 bytes from 192.168.1.108: icmp_seq=3 ttl=64 time=17.6 ms
In detect:64 bytes from 192.168.1.108: icmp_seq=4 ttl=64 time=31.8 ms
In detect:
In detect:--- 192.168.1.108 ping statistics ---
In detect:4 packets transmitted, 4 received, 0% packet loss, time 3005
丢包率:0%
In detect:rtt min/avg/max/mdev = 17.677/52.645/142.615/52.247 ms
时延:52

```

图 5.3: 探测服务器时延丢包测试

打开云游戏软件运行云游戏，同时运行系统，检测 5G 网络下终端设备带宽速率，其输出结果如图5.4所示：

```

System.out: Tag:网络质量 吞吐率:37681.957 kb/s
System.out: Tag:网络质量 吞吐率:34135.570 kb/s
System.out: Tag:网络质量 吞吐率:38565.175 kb/s
System.out: Tag:网络质量 吞吐率:36973.53 kb/s
System.out: Tag:网络质量 吞吐率:35684.480 kb/s
System.out: Tag:网络质量 吞吐率:50575.175 kb/s
System.out: Tag:网络质量 吞吐率:46280.800 kb/s
System.out: Tag:网络质量 吞吐率:40729.1027 kb/s
System.out: Tag:网络质量 吞吐率:45396.556 kb/s
System.out: Tag:网络质量 吞吐率:48185.525 kb/s
System.out: Tag:网络质量 吞吐率:41160.600 kb/s
System.out: Tag:网络质量 吞吐率:38684.112 kb/s
System.out: Tag:网络质量 吞吐率:38918.820 kb/s

```

图 5.4: 带宽探测功能测试

基于此，可验证系统能较为全面的探测网络连接方式变化，无线信号强度变化，带宽速率，时延抖动丢包率等数据，实现了全面获取用户网络质量的功能。

### 5.2.2 通信功能测试

在终端手机上运行 5G 逐用户切片系统客户端，与虚拟机 Ubuntu 切片策略服务器通信，其客户端显示结果如图5.5所示：

```

I/System.out: 发送切片请求
I/System.out: 正在与切片策略控制器建立连接: 222.20.103.130 5050
I/System.out: 222.20.103.130 5050 Connect success!

```

图 5.5: 通信成功控制台输出

服务器端连接成功信号如图5.6所示：

```
root@tby-virtual-machine: /home/tby/Desktop/python_service#
python sliStra.py
hostip: 192.168.226.130
切片策略服务器初始化成功
Traffic control 初始化成功
```

图 5.6: 通信成功服务器输出

可以认为成功建立 socket 通信。

### 5.2.3 切片功能测试

本功能测试首先主要针对 Traffic Control 限制带宽的能力进行测试，对比限制前后传输同一文件的用时，从而论证系统具有资源切片功能。

令虚拟机上运行的服务端向主机 windows 环境中的 Python 客户端传输 169MB 的 csv 文件 20 次，在第 11 次前令客户端向切片策略服务器发起请求，切片策略服务器通过 tc 命令采用 htb 队列设置各类资源带宽上限，并建立过滤器 filter 来部署对该用户应接入切片的决策，最后统计带宽切片前后用时变化。

如图5.7实验结果所示，限制前的平均用时为 2665.06ms，限制资源后平均用时为 3670.18ms，同比用时变为先前的 1.37 倍。从数据可以看出，该通信可使用的网络带宽资源成功被流量控制工具限速，可起到切片效果。

```
渲染服务器初始化成功
Connected by ('10.21.195.145', 59338)
第1次发送完毕,用时3095.041533999847ms
第2次发送完毕,用时2722.5946309999927ms
第3次发送完毕,用时2510.5217790001007ms
第4次发送完毕,用时2481.3793180001085ms
第5次发送完毕,用时2481.6475900001933ms
第6次发送完毕,用时2496.0710459999973ms
第7次发送完毕,用时3164.117688999795ms
第8次发送完毕,用时2440.3677909999715ms
第9次发送完毕,用时2474.687170999914ms
第10次发送完毕,用时2784.1255289999935ms
第11次发送完毕,用时3881.3220510000974ms
第12次发送完毕,用时3768.078539999806ms
第13次发送完毕,用时3885.525463000249ms
第14次发送完毕,用时3411.0825249999834ms
第15次发送完毕,用时3599.8909509999066ms
第16次发送完毕,用时3662.5146940000377ms
第17次发送完毕,用时3757.3488199996063ms
第18次发送完毕,用时3723.1154870000864ms
第19次发送完毕,用时3552.1449950001625ms
第20次发送完毕,用时3671.9123080001737ms
切片前平均用时: 2665.0554077999914
切片后平均用时: 3670.1793092222233
```

图 5.7: 服务器传输文件时延

## 5.3 切片框架性能测试

为验证本系统的切片技术对 5G 终端网络服务质量的保障能力,说明本系统对包括且不限于云渲染的应用场景均具有改善网络质量的效果,我们分别使用了传输文件的服务器系统和按帧率逐帧传输渲染音视频数据的服务器系统进行实验,将这两种服务器与策略控制器共同部署在同一主机上。在此基础上,我们分别在主机处于局域网中的 Wifi 环境,和主机位于云端数据中心的 5G 环境下设置实验,对我们设计的切片框架性能进行测试。

### 5.3.1 文件传输本地服务器测试

在终端 wifi 连接网络的环境下,将服务器部署于本地服务器,配置如表5.1所示。系统的终端,网络及服务器,策略服务器整体处于局域网环境下。

令服务器接入终端数量为 5。其中 4 个为 windows 主机运行的 python 客户端,用于模拟服务器网络拥塞与波动。1 个为需保障网络服务的安卓 12 手机客户端应用,设其需求分类结果为高带宽低时延。

为每个终端创建通信线程,分别对 python 客户端持续传输大文件,对安卓终端传输大小为 1080p60fps 的音视频以模拟云渲染应用运行实况。在传输十次文件后触发切片,对比切片前后网络性能变化,服务器端实验结果如图5.8所示:

```

tby@tby-virtual-machine:~/Desktop/python_service$ python service.py
[hostip: 192.168.110.129
渲染服务器开始运行
Connected by ('10.21.194.151', 59191), 对模拟客户端的传输实时用时不予显示
传输文件大小为: 377899146
Connected by ('10.21.194.151', 59198), 对模拟客户端的传输实时用时不予显示
传输文件大小为: 377899146
Connected by ('10.21.194.151', 59226), 对模拟客户端的传输实时用时不予显示
传输文件大小为: 377899146
Connected by ('10.21.194.151', 59251), 对模拟客户端的传输实时用时不予显示
传输文件大小为: 377899146
第1次向移动终端传输数据完毕,用时11527.885325998795ms
第2次向移动终端传输数据完毕,用时6363.033724999696ms
第3次向移动终端传输数据完毕,用时6159.049845000482ms
第4次向移动终端传输数据完毕,用时6514.859484001136ms
第5次向移动终端传输数据完毕,用时7896.35646899842ms
第6次向移动终端传输数据完毕,用时15706.865927000763ms
第7次向移动终端传输数据完毕,用时20660.52535400013ms
第8次向移动终端传输数据完毕,用时21981.780981001066ms
第9次向移动终端传输数据完毕,用时18341.58329999991ms
第10次向移动终端传输数据完毕,用时15246.296031000384ms
第11次向移动终端传输数据完毕,用时9222.53208300026ms
第12次向移动终端传输数据完毕,用时6639.832552998996ms
第13次向移动终端传输数据完毕,用时6146.43424899867ms
第14次向移动终端传输数据完毕,用时8005.188007999095ms
第15次向移动终端传输数据完毕,用时6483.665834000931ms
第16次向移动终端传输数据完毕,用时5637.015921000057ms
第17次向移动终端传输数据完毕,用时8012.560714001665ms
第18次向移动终端传输数据完毕,用时7681.6401980013325ms
第19次向移动终端传输数据完毕,用时8167.499018998569ms
第20次向移动终端传输数据完毕,用时6842.266364999887ms
切片前平均用时: 13039.823644200078
切片后平均用时: 7068.455873444489

```

图 5.8: 切片前后服务端传输文件用时

从图5.8的实验结果中可以看出,在服务器端吞吐量较大,通信资源相对紧缺时,切片前安卓终端的传输用时波动范围极大,在 [6000,22000] 区间内剧烈波动,用时统计标准差为 5842ms。

而借助切片技术对网卡带宽资源按类划分，并接入用户后，可以使向安卓移动终端的线程的传输平均用时由 13839.82ms 下降到 7068.46ms，平均吞吐量由 1.45MB/s 提升为 2.59MB/s，用时统计标准差降低至 1049.35ms，波动幅度大幅度减小，网络服务质量稳定。

对安卓终端服务效果体验，观察 Android Studio 控制台结果如图5.9所示：

```
Tag:性能测试 正在与渲染服务器建立连接: 10.21.194.151 5051
Tag:性能测试 10.21.194.151 5051 Connect success!
Tag:性能测试 传输文件大小: 19812731 B
Tag:性能测试 Use time: 12604ms
Tag:性能测试 Use time: 7372ms
Tag:性能测试 Use time: 7157ms
Tag:性能测试 Use time: 7616ms
Tag:性能测试 Use time: 8929ms
Tag:性能测试 Use time: 16790ms
Tag:性能测试 Use time: 21774ms
Tag:性能测试 Use time: 23034ms
Tag:性能测试 Use time: 19347ms
Tag:性能测试 Use time: 16309ms
Tag:性能测试 发送切片请求
Tag:性能测试 正在与切片策略控制器建立连接: 10.21.194.151 5050
Tag:性能测试 10.21.194.151 5050 Connect success!
Tag:性能测试 Use time: 10255ms
Tag:性能测试 Use time: 7766ms
Tag:性能测试 Use time: 7226ms
Tag:性能测试 Use time: 9003ms
Tag:性能测试 Use time: 7533ms
Tag:性能测试 Use time: 6675ms
Tag:性能测试 Use time: 9147ms
Tag:性能测试 Use time: 8684ms
Tag:性能测试 Use time: 9172ms
Tag:性能测试 Use time: 7842ms
```

图 5.9: 切片前后终端接收文件用时结果

对切片前后手机终端接收同一音视频文件进行统计分析，如图5.10所示。从带宽角度，在为手机终端分配切片资源前，文件传输用时区间在 [7000,23000]ms 之间波动。其平均用时为 14093.2ms。在为该手机 ip 分配资源形成切片后，文件传输用时稳定在 [6500,10000]ms 内，平均用时为 8330.3ms，网络吞吐率由 1.34MB/s 提升至 2.27MB/s，提升约 69.4%。

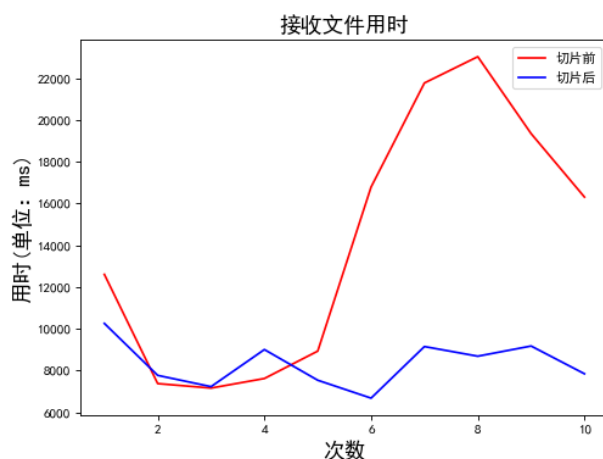


图 5.10: 切片前后终端接收文件用时统计

从时延与抖动的角度，绘制其文件接收用时变化曲线如图5.11所示，通过接收文件的用时变化情况间接反应抖动数据的变化，文件接收用时标准差由 5855.2867ms 缩小至 1039.6928ms。终端接收文件用时波动平均值由 3116.56ms 缩小至 1320.78ms，缩减至原先的 43.38%，大幅提高终端网络质量的稳定性。

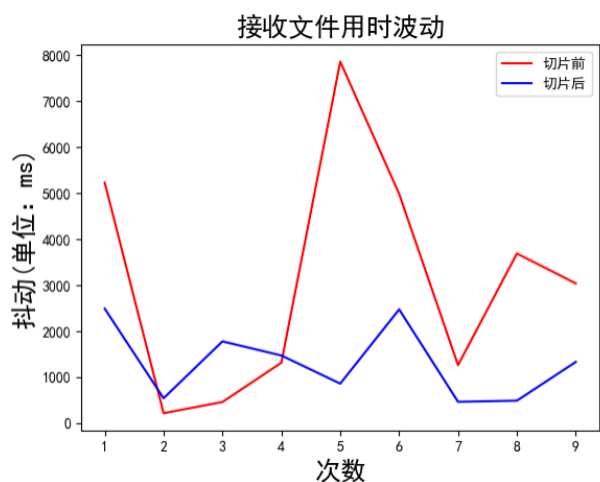


图 5.11: 切片前后终端接收文件用时波动

从可接入数量的角度来看，根据本实现时设计的切片实例，其接入用户容量可满足约 22 台终端应用设备通信网络需求。通过对切片实例的设计，调整和划分，其接入数量亦可进一步增加，杜绝其余拥塞流量对需保障应用的干扰。

### 5.3.2 文件传输云服务器测试

进一步地，在手机终端接入 5G 移动数据的环境下，将服务器部署于阿里云云服务器 ECS 上，其配置如表5.1所示，所处地域为深圳。该服务器上的切片实例设置同表??所示，带宽单位由 Mbps 改为 Kbps。



实施时，在 windows 主机上运行三个 python 客户端与服务器进行通信，传输 15.2MB 大小文件，模拟服务器的拥塞与波动环境。在 5G 安卓 12 手机上运行系统，受限于设备带宽资源，反复传输分辨率较低的音视频文件，模拟云渲染应用运行场景。在传输 5 次文件后触发切片，对比切片前后网络质量变化，其服务器端实验结果如图5.12所示：

```

/System.out: Tag:性能测试 正在与渲染服务器建立连接: 112.74.104.115 5051
/System.out: Tag:性能测试 112.74.104.115 5051 Connect success!
/System.out: Tag:性能测试 传输文件大小: 1380251 B
/System.out: Tag:性能测试 Use time: 39569ms
/System.out: Tag:性能测试 Use time: 35764ms
/System.out: Tag:性能测试 Use time: 34576ms
/System.out: Tag:性能测试 Use time: 34416ms
/System.out: Tag:性能测试 Use time: 37682ms
/System.out: Tag:性能测试 发送切片请求
/System.out: Tag:性能测试 正在与切片策略控制器建立连接: 112.74.104.115 5050
/System.out: Tag:性能测试 112.74.104.115 5050 Connect success!
/System.out: Tag:性能测试 Use time: 12881ms
/System.out: Tag:性能测试 Use time: 12860ms
/System.out: Tag:性能测试 Use time: 13280ms
/System.out: Tag:性能测试 Use time: 12839ms
/System.out: Tag:性能测试 Use time: 13057ms
/System.out: Tag:性能测试 切片前平均用时: 36401.4
/System.out: Tag:性能测试 切片后平均用时: 12983.4

```

图 5.12: 切片前后终端接收文件用时结果

从图5.12可以看出，客户端接收文件平均用时由 36401.4ms 减少到 12983.4ms，降低 64.34%，带宽由原本的 288kbps 提升到了 807kbps，可说明终端所使用的带宽从与三个 python 客户端均分，转变为依据切片分配而使用资源。绘制客户端接收文件用时及变化如图5.13所示：

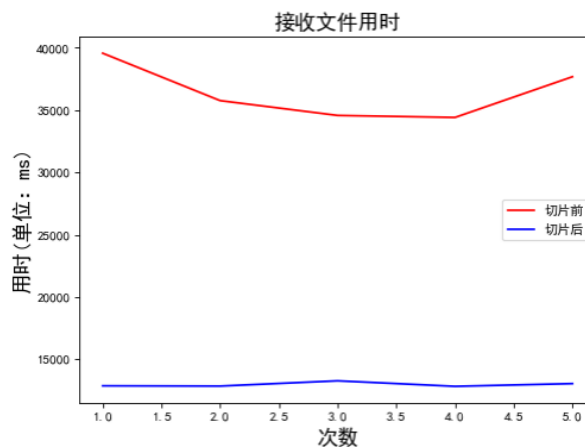


图 5.13: 切片前后终端接收文件用时统计

通过接收文件的用时变化情况间接反应抖动数据的变化如图5.14所示，文件接收用

时标准差由 1255.82ms 缩小至 244.75ms。终端接收文件用时波动平均值由 1419.22ms 下降至 469.37ms，缩减至原先的 34.97%，大幅提高终端网络质量的稳定性。

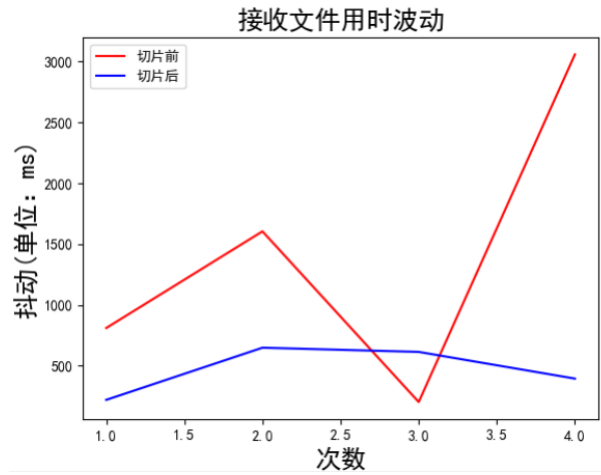


图 5.14: 切片前后终端接收文件用时波动

综上所述，基于手机终端连接 Wifi 环境和 5G 环境的实验说明，本面向 5G 终端的网络切片系统，可大幅度改善拥塞网络环境下对终端用户可感知的网络服务质量，并在吞吐量，时延与抖动等多个维度网络性能指标中体现，且让网络整体能容纳更多用户，提升系统整体资源的有效利用率。

### 5.3.3 视频渲染切片框架结果

在前文实验环境基础上，我们将整体实验环境进一步完善，实现了按帧率逐帧传输音视频的云渲染环境。

在该环境中，我们在 Android 客户端上运行需求为 1080p, 60fps 的云渲染应用，并启动相应的 Python 客户端运行需求为 720p, 30fps 的视频应用。运行一段时间后，客户端们向服务器上报切片请求或网络质量数据（每 15s 上传一次），策略服务器调用 ARMA 算法预测带宽并进行需求分类，在此基础上为各用户划分带宽和优先级。一段时间后，得到安卓客户端监测的带宽变化趋势如图5.15所示：

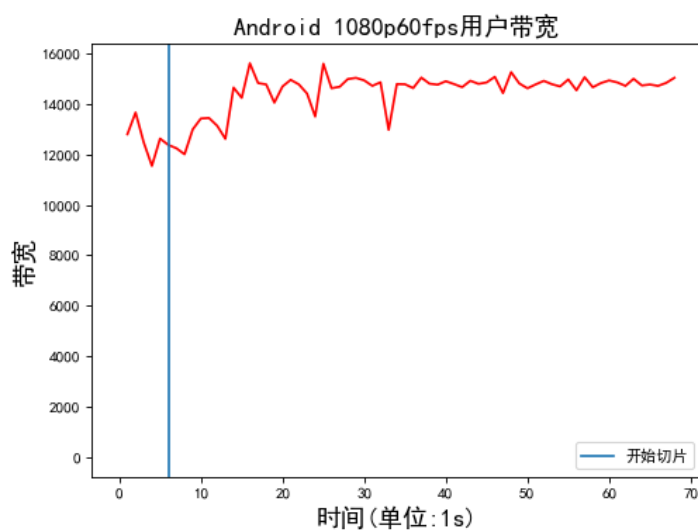


图 5.15: 切片前后 Android 客户端带宽变化

在服务器端，得到该 1080P 60fps 安卓客户端的每 15s 的平均 QoE 评价结果如图5.16所示：

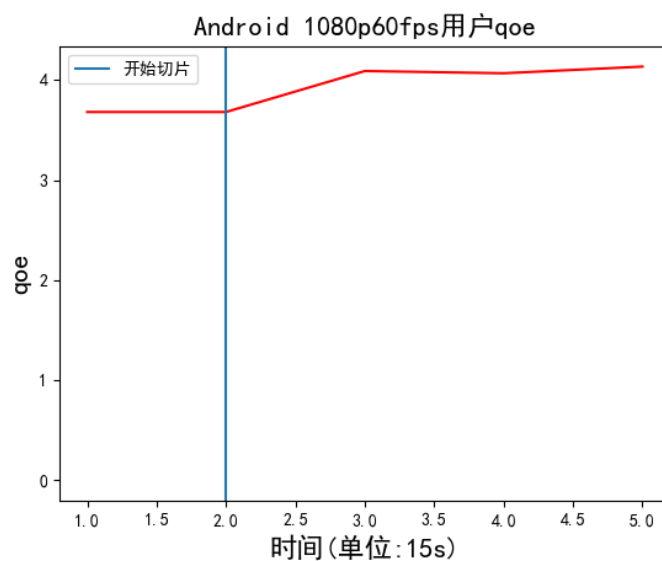


图 5.16: 切片前后 Android 客户端 QoE 变化

与此同时，在服务器端检查 python 客户端的 QoE 变化情况如图5.17所示：



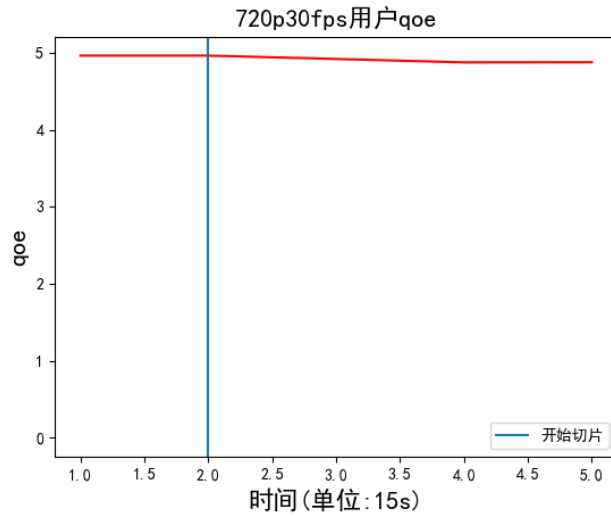


图 5.17: 切片前后 720p30fps 客户端 QoE 变化

根据以上结果，可以看出，在并未付出其余用户 QoE 评价明显降低的代价的情况下，本切片系统框架能针对用户的需求特点和历史带宽数据，进行精准的资源切片，能改善特定网络质量较差用户，将其 QoE 评价 MoS 值由约 3.6 提升至约 4.1。综合来看，本系统及切片框架能较好的保障云渲染应用用户网络需求。

### 5.3.4 视频渲染强化学习效果

强化学习策略主要结合切片策略服务器一同运行。我们的强化学习策略采用了在线学习的方法，训练时间较长，且需要一定数量的终端不断的接入，退出，以确保系统学习到尽可能全面的历史数据，并对不同数量终端接入情况决策均有效果。在前文所述本地实验环境中，较为拥塞的网络环境下，其初步实验结果如图5.18所示：

```
强化学习切片策略实施前平均qoe: 2.7399999999999998
成功运行切片策略服务器,等待qoe中
C:\Users\24168\anaconda3\envs\Qt\lib\site-packages\qym\spaces\box.py:128: UserWarning: WARN: Box bound precision lowered by casting to float32
  logger.warn(f"Box bound precision lowered by casting to {self.dtype}")
获取action执行后的state,current_step: 0 action_type 6 action[1] 6.666666666666667
成功运行切片策略服务器,等待qoe中
获取action执行后的state,current_step: 0 action_type 2 action[1] 13.333333333333334
成功运行切片策略服务器,等待qoe中
获取action执行后的state,current_step: 0 action_type 2 action[1] 20.0
成功运行切片策略服务器,等待qoe中
获取action执行后的state,current_step: 0 action_type 2 action[1] 26.666666666666668
成功运行切片策略服务器,等待qoe中
获取action执行后的state,current_step: 0 action_type 2 action[1] 33.33333333333333
强化学习切片策略实施后平均qoe: 3.891944816977204
Process finished with exit code 0
```

图 5.18: 切片前后 720p30fps 客户端 QoE 变化

可以确认的是，在较为拥塞的网络环境下，训练后的强化学习策略能对各个终端部署较为合适的不同的切片，以提升系统接入的所有用户的平均 QoE

## 6 创新与特色

5G 作为新一代网络通信技术，为用户带来高带宽，低时延和大连接的移动通信体验。为充分利用网络资源，灵活地为应用提供多样化定制网络服务，5G 引入网络功能虚拟化与软件定义网络技术，利用网络切片构建松耦合、易扩展的新型网络架构，以高效满足如云渲染等应用的网络需求。

5G 切片在工业远程控制等固定网络、场景和应用已有相对成熟的方案和实践。但移动场景网络状态变化条件下，采用 5G 切片满足虚拟现实、云游戏、自动驾驶等高带宽低时延云渲染应用需求仍存在较多问题亟待解决。一方面，现有分类需求描述难以精细描述业务逐流特征和需求。另一方面，按类粗粒度切片难以避免移动动态网络环境下用户和业务间的资源竞争与网络质量及需求的动态变化，无法为云渲染类终端应用提供稳定的服务保障。

作为 5G 的核心技术之一，5G 切片技术可针对应用网络需求，在同一物理网络上利用软件定义网络技术和网络功能虚拟化技术对资源进行切片，形成逻辑子网络，以虚拟专网的方式为应用提供网络服务。通过 5G 切片技术针对用户需求对网络进行资源编排与切片，可以更好地保障 5G 应用对网络服务的高质量需求，提升用户体验，提高网络整体资源利用率。

云渲染类应用由于其云端渲染，终端操作的特性，对网络需求尤为高，需要应用 5G 切片技术提升其网络能力。进一步地，为了将 5G 切片技术应用到实际云渲染场景中，突破其网络瓶颈，需回答切片策略与部署问题，填补解决其实际应用流程目前的缺陷，实现网络质量监测功能，用户体验评估问题等本作品主要工作如下：

- (1) 深度调研云端渲染应用网络需求，给出整体量化 QoE 指标。并基于云端应用渲染需求基准，对用户需求进行建模，来实现切片的精准划分以及网络资源合理调配。最终验证于云渲染模拟环境。
- (2) 针对面向移动终端的逐用户 5G 网络我们提出了一种完整的网络切片策略方法。结合学术界最新的研究成果，该策略方法将深入分许云端渲染是是网络环境，通过不断自学习，动态预测用户需求以及实施精准的切片划分服务。实验证明，此切片策略有效地提高了切片效率，网络整体质量以及资源利用率。
- (3) 基于本系统在真实网络环境中搭建云渲染结合切片的实验系统。以安卓手机作为移动终端设备客户端，深入调研其网络数据质量并获取其相应接口，对实际云渲染网络环境进行监听，测量与建模。以 Linux 系统作为云渲染服务器，搭建可同时向多个 Android 或 Python 客户端按帧率逐帧传输音视频数据的云渲染实验环境。
- (4) 针对本系统的切片策略，网络建模以及云端渲染系统进行了广泛深入的验证与实践。在手机端，网络服务器构成的真实云渲染应用场景中，我们实现了在应用层

带宽预测，用户按需切片，强化学习等切片策略。并通过相应的网络技术工具编排网络资源以及切片的部署。

## 6.1 作品创新点

云渲染应用正蓬勃发展，已成为诸多领域的一大发展方向。其场景包括且不限于云游戏、远程医疗（特别是远程手术）、VR、自动驾驶、远程工控、智慧园区等，它们均需要低延时、高带宽的稳定网络服务。

本作品基于学术界 5G 切片技术研究，在提升服务质量和资源利用率的相关策略基础上进行适配与改进，针对云渲染场景下的移动终端，提出一种切实可行的 5G 切片机制系统及方法，针对逐个用户（简称：逐用户）在用户粒度进行网络切片的设计接入与部署，为用户提供基于其需求的精准网络服务，从而克服云端渲染本地展示的 VR，AR 等高带宽差异化网络应用需求瓶颈，推动 5G 切片技术在云渲染场景应用中的实际使用。本作品创新点如下：

1. 深入分析云端应用网络需求，给出量化 QoE 指标，并在云渲染应用上进行深入阐述。同时基于云渲染应用需求，为用户需求进行建模，进而精确定位切片针对服务需求所需编排的相关网络资源，并对用户体验进行 QoE 量化评估。
2. 在系统中实现基于定时测量与监听事件回调的网络测量机制、基于用户网络质量的间接 QoE 评估机制，以深度学习与强化学习为基础，针对实际渲染网络建模，提出一种完整的面向移动终端的逐用户 5G 网络切片系统及策略方法。横向对比学术界最新框架，进行了广泛深入的实验与验证。
3. 在真实的 5G 场景云端渲染应用下对切片策略与渲染系统进行实践。横向对比学术界最新研究框架，在手机端与服务器组成的云端渲染应用环境中进行广泛的实验。大量实验证明，我们的策略可以及时且精准的切片服务，提高切片策略效率，保障服务质量和资源高利用率。

总而言之，围绕云渲染终端场景开展的 5G 切片策略及机制研究工作，将有助于 5G 切片技术切实投入移动终端各类应用中，帮助各类应用克服网络瓶颈，具有 5G 切片策略研究的理论意义和 5G 切片技术实际运用的实践意义及商用价值。

## 6.2 作品特色

本作品以云渲染应用中终端与服务器的通信为主要场景，5G 切片在其中的应用作为主要研究内容，探讨 5G 切片面向移动终端场景的机制策略，解决 5G 切片技术应用需克服的问题，并进行一定的实践。本作品特色如下：

- (1) 提出面向终端的 5G 逐用户切片系统及策略：该系统与方法针对应用数据流特性，为各用户提供切片服务。系统与策略考虑了终端网络质量与实际用户需求，涵盖完整的切片流程，包括切片的发起、建立、部署和管理等部分，能够适应各种网络应用需求，并在实际任务中解决 5G 切片技术应用到移动终端场景面临的问题。
- (2) 在真实 5G 网络环境中实现云渲染切片框架系统：基于手机终端安卓系统和服务器等可控网络资源，进行 Android 5G 终端与服务器的联合音视频服务器云渲染开发，模拟云渲染应用通信过程；在网络层和应用层实现所提出的切片框架及其策略，提供较好的云渲染应用网络服务。
- (3) 在实验系统中提出切片框架与策略进行实践：在应用层实现结合带宽预测，用户按需切片，强化学习等切片策略，并通过相应技术工具实现网络资源的编排及切片部署，按需精准将用户需求与网络资源映射，提供精准化高效的网络切片服务，改善云渲染用户网络质量。