**OpenHarmony竞赛训练营**

**应用生态题目赛题七：OpenHarmony图片加载弱网优化**

|  |  |
| --- | --- |
| 学校名称： | 华东师范大学 |
| 团队名称： | bug创造者团队 |
|  |  |
| 队长： | 张子彤 |

OpenHarmony竞赛训练营组委会

2025年8-9月

目 录

[1 简介 3](#_Toc147755079)

[1.1 背景 3](#_Toc147755080)

[1.2 目的 3](#_Toc147755081)

[2 设计描述 3](#_Toc147755082)

[2.1 总体设计 3](#_Toc147755083)

[2.2 实现思路 3](#_Toc147755084)

[2.3 系统结构 3](#_Toc147755085)

[2.3.1 模块划分 3](#_Toc147755086)

[2.3.2 系统架构说明 3](#_Toc147755087)

[2.3.3 文件结构 3](#_Toc147755088)

[2.4 模块功能描述 3](#_Toc147755089)

[2.4.1 模块/子系统1功能描述 4](#_Toc147755090)

[2.4.2 模块/子系统2功能描述 4](#_Toc147755091)

[2.4.3 模块/子系统3功能描述 4](#_Toc147755092)

[2.5 业务/实现流程说明 4](#_Toc147755093)

[2.5.1 用例1处理流程 4](#_Toc147755094)

[2.5.2 用例2处理流程 4](#_Toc147755095)

[2.6 接口描述 4](#_Toc147755096)

[2.6.1 调用接口 5](#_Toc147755097)

[2.6.2 提供接口 5](#_Toc147755098)

[2.7 UI设计 5](#_Toc147755099)

[3 其他 5](#_Toc147755100)

[3.1 成员分工 5](#_Toc147755101)

[3.2 困难与思考 5](#_Toc147755102)

[3.3 参考 5](#_Toc147755103)

# 简介

## 背景

## 随着移动互联网的快速发展，用户对应用体验的要求越来越高。在OpenHarmony系统中，图片资源加载是应用性能的关键指标之一。然而，由于无线网络及广域网的网络状态不确定性，当获取页面资源的网络吞吐性能无法得到满足时，用户页面刷新时就会出现如白块、卡顿等，严重影响用户体验。

## 当前，OpenHarmony系统弱网优化面临两大技术挑战：

## 网络状态预测的准确性：终端设备的网络状态变化不仅受用户行为模式影响，也受到复杂的无线环境干扰。在进入地铁、电梯等场景时，Wi-Fi/蜂窝信号常常发生非线性突变，传统静态阈值难以及时捕捉。

## （2）多网络切换决策的合理性：设备通常同时支持 Wi-Fi 与蜂窝网络。过早切换到蜂窝会带来流量浪费和用户舆情风险，而过晚切换则可能导致页面长期处于弱网环境，产生明显的加载卡顿与刷新失败。

## 为解决上述问题，本方案基于OpenHarmony提供的网络能力，设计并实现了一套高性能弱网优化方案，旨在图片资源加载场景下实现最短的图片下载完成时间，同时保证流量可控与用户体验平衡。

## 目的

## 本文档旨在详细描述OpenHarmony图片加载弱网优化方案的设计思路和实现方法，为代码实现与评测复现提供统一的设计依据。具体目的包括：

## （1）分析弱网环境下图片加载的技术挑战，提出系统性的解决方案

## （2）设计基于EWMA + CUSUM的网络状态预测机制与弱网检测器，提升弱网识别的稳定性与准确性

## （3）提出智能切换与分阶段迁移策略，确保网络切换时机合理，降低用户等待与流量开销

## （4）实现多路径并发下载与自适应调度，支持任务优先级划分和并发度动态调

## （5）集成断点续传与轻量探测机制，提升弱网恢复场景下的鲁棒性

## （6）给出完整的系统架构、模块划分、接口定义与业务流程，确保开发实现具有可操作性与可验证性

## （7）建立统一的评测与复现方法，保证有效性与可信度

# 设计描述

## 总体设计

## 本方案基于OpenHarmony系统提供的网络能力，设计了一套完整的弱网图片加载优化系统。系统通过“判弱检测 → 智能切换 → 分阶段迁移 → 并发自适应 → 断点续传 → 轻量探测 → 性能评测”的闭环，实现弱网场景下的最优加载体验。整体包括以下六个核心功能模块：

## 网络状态感知与预测模块

## 基于 EWMA（指数加权移动平均）和 CUSUM（累积和）实现弱网趋势预测

## 引入失败率、多指标融合与置信度输出，提升预测的稳健性

## 智能网络切换决策模块

## 在检测到弱网时触发用户可感迁移，通过 UI 引导完成 Wi-Fi 到蜂窝的切换

## 设置门控条件与冷启动保护，避免误判与过早切换

## 切换过程中统计等待时间，保证评测口径公平

## 多路径并发下载模块

## 采用优先队列（小文件优先）+ 动态并发度调整，在弱网阶段收尾小文件，切换后提升并发度

## 支持 Wi-Fi 与蜂窝分阶段下载，记录字节占比与路径分布

## 断点续传与轻量探测模块

## 当服务器支持 Range 时，大文件采用分块续传，保证切换/断连后不中断

## 利用低频 Range 1B 探测 RTT，探测成本 <1%，用于预测增强

## 用户性能监控与统计分析模块

## 双口径统计：wallTime（含等待）与 totalTime（不含等待）

## 一致性校验：逐文件时长总和 ≈ 有效总时长

## 用户交互与模拟环境支持模块

## 提供简洁直观的界面，支持参数配置、模式选择、日志输出

## 集成网络领航员，模拟“地库/离家”等典型弱网场景，支持自动化对比测试

## 通过上述模块的协同运行，系统能够在弱网环境下有效缩短图片资源下载完成时间，降低用户感知延迟，并保持实验的公平性与可复现性。

## 2.2 实现思路

## 本方案的实现围绕“弱网识别 → 智能切换 → 并发调度 → 断点续传 → 轻量探测 → 公平评测”展开，关键思路如下：

## （1）弱网检测与预测

## 使用 EWMA 平滑速率趋势，结合 CUSUM 变点检测，识别网络性能持续下降

## 引入失败率等多维信号融合，输出 {isWeak, confidence}，保证判定稳健

## 智能切换机制

## 采用“用户可感迁移”模式：当弱网成立时，引导用户切换至蜂窝热点或更优 Wi-Fi

## 切换过程统计等待时间，用于区分真实耗时与用户等待

## 任务调度与并发自适应

## 在弱网阶段降低并发度（2–3）并优先完成小文件，避免尾部拖延

## 切换完成后提升并发度（6–8），加速大文件下载，整体缩短总时长

## （4）断点续传策略

### 对大文件启用 Range: bytes=offset- 请求，记录 offset 与校验信息

### 断网或切换后可继续下载剩余部分，保证任务不中断

## （5）轻量探测机制

### 采用 Range 1B 低成本请求周期性探测 RTT

### 在弱网概率升高时自动缩短探测间隔，探测成本控制在总流量的 <1%

### （6）性能监控与评测

### 双口径：输出 wallTime（含等待）与 totalTime（剔等待），并进行一致性校验

## 2.3 系统结构

### 2.3.1 模块划分

### 系统根据功能分为以下七个模块：

### （1）网络状态感知与预测模块（NetworkMonitor & WeakNetDetector）

### 实时采集速率、失败率等指标

### 基于 EWMA + CUSUM 算法完成弱网判定，输出置信度

### （2）智能网络切换决策模块（NetworkSwitcher）

### 根据检测结果触发用户可感迁移

### 管理切换过程，记录等待时长，保证公平统计

### （3）任务调度与并发下载模块（DownloadManager & PriorityPool）

### 采用“小文件优先 + 并发度自适应”策略

### 在弱网与切换后分别调整并发池大小，缩短整体下载时间

### （4）断点续传与轻量探测模块（RangeHandler & ProbeManager）

### 利用 Range 请求实现大文件续传

### 周期性发送 Range 1B 探测，成本 <1%，辅助预测

### （5）性能监控与统计分析模块（PerformanceMonitor）

### 记录 wallTime / totalTime、弱网触发点、切换时刻、流量占比等数据

### 输出标准化 JSON/CSV，保证公平性与可复现

### （6）用户界面与环境模拟模块（UI & NetworkNavigator）

### 提供一键运行、模式切换、结果展示功能

### 集成网络领航员，模拟“地库/离家”等典型弱网场景。

### （7）服务器图片模块（Nginx 静态服务）

### 部署在阿里云 Ubuntu 22.04，配置 Nginx 提供 155 张图片资源

### 目录结构固定：/var/www/images/

### 支持缓存与跨域，保证下载源稳定一致

### 为弱网实验提供统一数据集，保证评测公平

### 2.3.2 系统架构说明

系统整体分为客户端子系统和服务端子系统。服务端由 Nginx 图片服务组成，负责提供统一的图片下载源；客户端通过弱网检测、切换和调度实现优化。

客户端系统采用四层分层架构：

应用层：用户界面（Index.ets），负责参数输入、模式选择和结果展示

业务逻辑层：弱网检测、切换决策、任务调度、性能监控

网络适配层：封装 HTTP 请求、Range 支持、探测逻辑、网络领航员接口

系统服务层：依赖 OpenHarmony 提供的 HTTP 服务、连接管理与系统设置能力

模块交互流程：

（1）用户启动实验，配置参数

（2）DownloadManager 调用 NetworkMonitor，实时采样速率等数据

（3）WeakNetDetector 输出弱网判定，传递给 NetworkSwitcher

（4）NetworkSwitcher 触发切换，引导用户进入更优网络

（5）PriorityPool 管理下载队列，调度小文件与大文件

（6）RangeHandler 在切换中断后继续下载，ProbeManager 周期性探测

（7）PerformanceMonitor 收集完整数据，Index 界面展示结果

### 2.3.3文件结构

#### 整个NetBoost架构

/src/main/ets

  /pages

    - Index.ets                *// 主界面，入口页面*

  /common

    - HttpDownloader.ts        *// HTTP 下载封装*

    - Runner.ts                *// 实验执行器（弱网判定 + 切换串行逻辑）*

    - WeakNetDetector.ts       *// 弱网检测算法实现*

    - PriorityPool.ts          *// 并发调度池*

    - RangeHandler.ts          *// 断点续传封装*

    - ProbeManager.ts          *// 轻量探测管理器*

    - PerformanceMonitor.ts    *// 性能监控与日志输出*

  /navigator

    - NetworkNavigator.ets     *// 网络领航员封装*

#### 服务器端架构

/var/www/images/                        *# 图片资源存放目录（img\_001.jpg … img\_155.jpg）*

/etc/nginx/sites-available/images.conf  *# Nginx 图片服务配置文件*

/etc/nginx/sites-enabled/images.conf    *# Nginx 启用的站点软链接*

/var/log/nginx/images\_access.log        *# 图片服务访问日志*

/var/log/nginx/images\_error.log         *# 图片服务错误日志*

## 2.4模块功能描述

### 2.4.1 服务器图片模块（Nginx 静态服务）

#### **功能**

#### 该模块的核心作用是为弱网优化实验提供统一的图片下载源。通过在云服务器上搭建基于 HTTP 的图片服务，向 HarmonyOS 客户端提供 155 张图片资源，保证数据来源稳定一致，便于后续对比实验。

#### **实现**

#### 服务器环境：阿里云 Ubuntu 22.04，2 vCPU / 2GB 内存 / 200 Mbps 带宽。

#### 软件：使用 Nginx 提供静态资源服务，开启目录索引与缓存控制，支持跨域访问。

#### 目录结构：图片存放于 /var/www/images/，Nginx 配置文件位于 /etc/nginx/sites-available/images.conf。

#### **部署步骤**

### 登录服务器：通过 SSH 远程进入云主机。 （2）安装并启动 Nginx：使用 apt 安装并配置为开机自启。 （3）上传图片：将本地 155 张 JPG 图片传输至服务器并移动到 /var/www/images/ 目录下。 （4）配置 Nginx 站点：新增独立的 images.conf 配置文件，设置静态目录、缓存策略与 CORS 头。 （5）验证：浏览器访问 http://139.224.130.188/images/，能够看到 img\_001.jpg … img\_155.jpg 的列表。

#### **效果与作用**

#### 为 HarmonyOS 端实验提供真实的 HTTP 下载环境

### 支持缓存与跨域，保证测试过程的稳定性

### 2.4.2 网络状态感知与预测模块

#### **目的**

#### 实时感知网络质量，预测是否进入弱网，为切换决策提供依据。

#### **功能**

#### 收集下载速率、失败率等核心指标

#### 使用 EWMA 算法平滑速率趋势，过滤短时抖动

#### 使用 CUSUM 检测持续下降趋势

#### 结合失败率输出 {isWeak, confidence} 判定结果

#### 触发弱网预警事件

### 2.4.3 智能网络切换决策模块

#### **目的**

#### 在弱网判定成立时，决定是否切换网络，并管理切换过程。

#### **功能**

#### 接收 WeakNetDetector 的判定结果和置信度

#### 当满足切换条件时，引导用户打开系统设置，完成 Wi-Fi → 蜂窝或更优网络迁移

#### 记录切换前后 netId 的变化，统计等待时间（pausedMs）

#### 保证切换过程对下载任务平滑衔接

#### 提供兜底策略（切换超时回退到基线模式）。

### 2.4.4 任务调度与并发下载模块

#### **目的**

高效管理图片下载任务，提升整体下载速度。

#### **功能**

管理任务队列，按“小文件优先、大文件延后”策略排序

提供并发池，支持动态调整并发度（弱网时 2–3，切换后 6–8）

每个下载任务失败时支持重试，避免整体中断

下载进度实时反馈给性能监控模块。

### 2.4.5 断点续传与轻量探测模块

#### **目的**

增强系统鲁棒性，保证在网络切换或中断时任务不中断。

#### **功能**

对大文件启用 Range: bytes=offset- 请求，记录 offset 与校验信息

切换后可直接续传未完成部分，避免重复下载

使用 Range 1B 请求进行 RTT 探测，周期性评估网络质量

自适应调整探测间隔，在弱网风险升高时加快探测

### 2.4.6 性能监控与统计分析模块

#### **目的**

提供公平、可复现的性能数据，作为优化效果的量化依据。

#### **功能**

记录 wallTime（含等待）、totalTime（不含等待）、pausedMs 等核心指标

统计下载总字节数、弱网触发点、切换时刻、失败率

输出统一格式的 JSON/CSV，便于对比和二次分析

校验一致性：逐文件耗时总和 ≈ totalTime

### 2.4.7 用户界面与环境模拟模块

#### **目的**

提供用户交互和弱网环境复现能力。

#### **功能**

UI 界面支持参数配置（服务器地址、文件数）、模式选择（Wi-Fi Only / Auto Switch）、一键运行

实时展示 5 轮测试的结果数据

支持日志滚动查看，便于开发与调试

集成网络领航员，模拟“地库/离家”等典型弱网场景

支持在相同条件下复现实验结果

## 2.5 业务/实现流程说明

### 2.5.1 基础用例处理流程

#### **用例描述**

在弱网环境下完成图片资源下载，对比基线模式（Wi-Fi Only）与优化模式（Auto Switch），统计下载总时长和流量占比。

#### **流程说明**

（1）初始化阶段

用户通过 Index 页面配置服务器地址与下载数量，选择模式（Wi-Fi Only / Auto Switch）

系统初始化各模块：NetworkMonitor、WeakNetDetector、NetworkSwitcher、DownloadManager、PerformanceMonitor

（2）清单获取与检测阶段

客户端从服务器获取图片列表

NetworkMonitor 收集速率与失败率，WeakNetDetector 进行 EWMA+ CUSUM 判弱

若弱网成立 → 传递判定结果给 NetworkSwitcher

（3）切换与迁移阶段（仅在 Auto Switch 模式下触发）

NetworkSwitcher 引导用户进入系统设置，完成 Wi-Fi → 蜂窝切换

记录切换等待时间，保证评测公平

（4）下载执行阶段

DownloadManager 创建任务队列

PriorityPool 管理并发度：弱网时 2–3，并优先完成小文件；切换完成后提升到 6–8，并迁移大文件

RangeHandler 对大文件启用断点续传，避免切换过程中的中断

ProbeManager 定期发起 Range 1B 请求，辅助预测

（5）结果统计与展示阶段

PerformanceMonitor 记录 wallTime、totalTime、pausedMs、弱网触发点、流量占比等指标

数据写入 JSON/CSV 文件，Index 页面实时显示 5 轮结果

用户可对比 Wi-Fi Only 与 Auto Switch 的平均耗时

### 2.5.2 挑战用例处理流程

#### **用例描述**

#### 利用 OpenHarmony 网络领航员功能，模拟“地库/离家”等典型弱网场景，验证优化方案效果，并对比自定义预测策略与基线性能。

#### **流程说明**

#### （1）初始化阶段

#### 用户在 Index 页面选择“挑战测试”

#### 系统初始化 NetworkNavigator、NetworkMonitor、WeakNetDetector、NetworkSwitcher、DownloadManager

#### （2）地库场景模拟

#### NetworkNavigator 构建地库弱网环境

#### NetworkMonitor 与 WeakNetDetector 监控并预测弱网趋势

#### NetworkSwitcher 在预测到弱网时触发切换

#### DownloadManager 执行下载，记录切换过程中的性能指标

#### （3）离家场景模拟

#### NetworkNavigator 构建 Wi-Fi 信号渐弱的离家场景

#### NetworkMonitor & WeakNetDetector 持续预测

#### NetworkSwitcher 再次进行切换决策

#### DownloadManager 继续调度与下载

#### （4）结果统计与展示

#### PerformanceMonitor 收集完整实验数据

#### 系统输出 JSON/CSV 文件，包含不同场景下的 wallTime、totalTime 与流量占比

#### Index 页面展示对比结果，用户可查看优化效果在不同场景下的差异

## 2.6 接口描述

### 2.6.1 调用接口

### 系统主要调用 OpenHarmony 提供的网络相关接口，包括：

### **HTTP 服务接口**

### http.createHttp(): HttpRequest —— 创建 HTTP 客户端实例

### HttpRequest.request(options: HttpRequestOptions): Promise<HttpResponse> —— 发起请求

### 支持 Range 请求头，实现断点续传与轻量探测

### **连接管理接口**

### connection.getAllNets(): Promise<Array<NetHandle>> —— 获取所有可用网络

### connection.getNetCapabilities(handle: NetHandle): Promise<NetCapabilities> —— 获取网络能力

### 用于判断蜂窝/Wi-Fi 可用性

### **网络领航员接口**

### navigator.simulateScene(scene: string): void —— 模拟弱网场景（地库/离家等）

### navigator.stopSimulation(): void —— 停止模拟

### 2.6.2 提供接口

### 系统内部封装并向上层提供以下接口：

### 弱网检测器 (WeakNetDetector)

interface WeakDecision {

  isWeak: boolean;     *// 是否判定为弱网*

  confidence: number;  *// 判定置信度 [0,1]*

}

### 网络预测器 (NetworkPredictor)

interface PredictionResult {

  willBeWeakNetwork: boolean;

  confidence: number;

  estimatedTime: number;

  recommendedAction: NetworkAction;

}

enum NetworkAction {

  KEEP\_CURRENT, SWITCH\_TO\_CELLULAR, SWITCH\_TO\_WIFI, LOAD\_BALANCE

}

### 下载管理器 (DownloadManager)

interface DownloadTask {

  id: string;

  url: string;

  priority: number;

}

interface DownloadStatus {

  taskId: string;

  status: DownloadState;

  progress: number;

  speed: number;

}

enum DownloadState { PENDING, DOWNLOADING, COMPLETED, FAILED }

### 性能监控器 (PerformanceMonitor)

interface PerformanceReport {

  wallTimeMs: number;      *// 含等待总耗时*

  totalTimeMs: number;     *// 剔除等待总耗时*

  successRate: number;

  networkSwitchCount: number;

}

## 2.7 UI 设计

### 本系统 UI 遵循简洁直观原则，主要包含以下区域：

### **标题区域**

### 文本：弱网图片加载优化实验

### 显示当前模式（基线 / 自动切换）

### **配置区域**

### 输入框：服务器地址（默认 http://<ip>/images/）

### 开关：是否启用预测切换、是否优先蜂窝

### **控制区域**

### 按钮：运行基线测试、运行预测切换测试

### 状态变化：点击后禁用按钮，显示“运行中…”

### **结果展示区域**

### 总耗时、总字节数、弱网触发点、切换次数

### 实时日志输出

### **视觉风格**

### 采用系统默认主题，整体风格与 OpenHarmony 保持一致

### 重要结果加粗，高亮显示

### 控件间距适中，界面紧凑整洁

### 日志使用较小字体，避免占用过多空间

# 3 其他

## 3.1 成员分工

### 一个人全部完成

## 3.2 困难与思考

## 在项目实施过程中遇到的主要困难与解决思路：

## 弱网预测准确性不足

## 困难：复杂无线环境下网络指标波动大，传统阈值法容易误判

## 思考：通过 EWMA 平滑 + CUSUM 变点检测结合失败率信号，提升判定稳健性；未来可引入 ML 模型进行预测增强

## 网络切换时机难以把握

## 困难：切换过早浪费流量，过晚导致页面卡顿

## 思考：采用“用户可感迁移”策略，引导用户确认；并通过置信度量化与门控机制降低误切概率

## 并发调度与尾部拖延问题

## 困难：弱网环境下大文件拖慢整体完成时间

## 思考：采用“小文件优先 + 动态并发池”策略，切换前收尾小文件，切换后提高并发下载大文件

## 真实弱网环境难以复现

## 困难：测试场景依赖复杂环境，如地库、电梯，难以保证稳定性

## 思考：利用 OpenHarmony 网络领航员模拟弱网环境，并通过脚本批量测试，保证结果可重复、可复现

## 系统性能与资源消耗平衡

## 困难：弱网检测、轻量探测与监控会额外消耗 CPU/流量

## 思考：采用轻量级 Range 1B 探测，探测流量占比控制 <1%；监控采用自适应采样频率，平衡准确性与开销

## 3.3 参考

### **官方文档**

#### OpenHarmony Network Kit 子系统介绍

#### <https://developer.huawei.com/consumer/cn/doc/harmonyos-guides/network-kit>

#### Network Boost Kit 子系统介绍

#### <https://developer.huawei.com/consumer/cn/doc/harmonyos-guides/network-boost-kit-guide>

#### 网络领航员功能介绍

#### <https://developer.huawei.com/consumer/cn/doc/harmonyos-guides/network-navigator>

#### 依赖代码仓

#### <https://gitee.com/openharmony/communication_netstack>

### **技术论文**

### "Network-Aware Application Adaptation for Mobile Computing"，IEEE Transactions on Mobile Computing

### "Predictive Network Switching for Improved User Experience in Heterogeneous Wireless Networks"，ACM MobiCom

### "Adaptive Resource Allocation for Multi-Homed Mobile Devices"，IEEE ICC

### **开源项目与标准**

### OpenHarmony 网络管理相关示例

### HTTP/1.1、HTTP/2 协议规范

### TCP/IP 协议栈相关标准

### **行业实践**

### 主流移动应用的弱网优化经验

### 多网络环境下的资源调度与负载均衡案例