

# 多网聚合加速（腾讯云聚通） 操作指南



腾讯云

## 【 版权声明 】

©2013–2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

# 文档目录

## 操作指南

功能概述

设备管理（自有设备）

设备管理（厂商设备管理）

设备管理（动态设备/应用）

流量监控

流量用量统计

活跃设备统计

流量包相关

其他功能

物联网卡白名单配置操作指南

动态加速配置参考

# 操作指南

## 功能概述

最近更新时间：2026-06-04 11:14:06

本文为您介绍在多网聚合加速（腾讯云聚通）控制台进行设备全生命周期管理的操作流程，包括自有设备/厂商设备/动态应用的创建、授权、监控及流量管理等操作。

### 准备工作

- 已注册腾讯云账号并完成实名认证，有关实名认证的介绍请参见 [实名认证基本介绍](#)。
  - 如需注册腾讯云账号，请单击 [注册腾讯云账号](#)。
  - 如需完成实名认证，请单击 [实名认证](#)。
- 使用腾讯云账号登录 [控制台](#)。

### 设备管理

在设备管理页面，您可对设备进行新建、查看、修改及删除操作。

目前多网聚合加速（腾讯云聚通）允许三种设备类型进行绑定加速，根据需求选择以下设备类型进行界面操作。

设备类型	描述
自有设备	由客户自行拥有所有权、使用权的设备。
厂商设备	由设备制造商生产并直接提供的设备。这些设备通常具有标准化的设计和功能，面向市场销售，供不同企业和用户购买和使用。
动态设备/应用	对于海量设备接入的场景（例如手机应用），原来的一机一密不再适用，因此可以使用动态设备/应用的形式：一个密钥可以动态创建设备的集合，可以对应一个客户的应用。客户可以基于应用密钥，不需要创建一机一密的密钥，在客户端发起连接。

### 相关功能

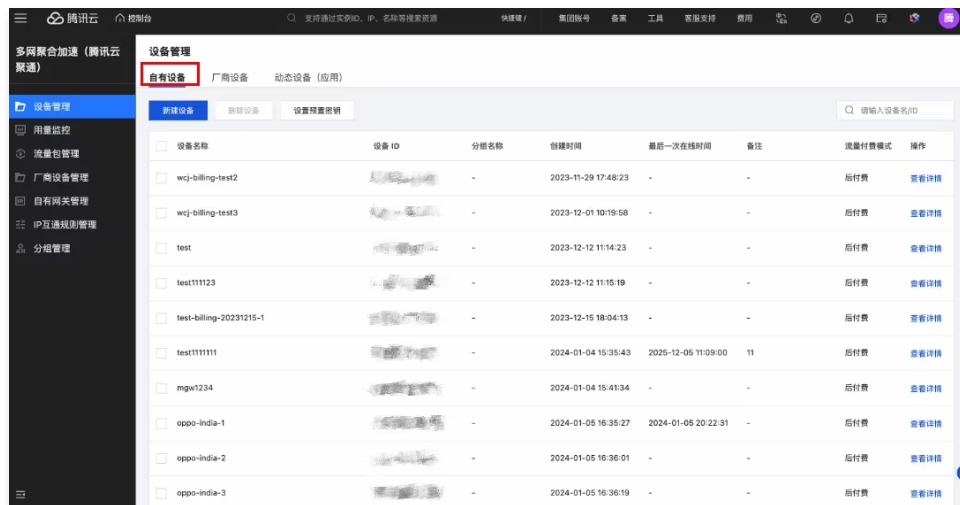
- [设备管理（自有设备）](#)
- [设备管理（厂商设备管理）](#)
- [设备管理（动态设备/应用）](#)
- [用量监控](#)
- [流量包相关](#)
- [其他功能](#)

# 设备管理（自有设备）

最近更新时间：2026-06-16 10:08:30

## 1. 新建设备

进入产品 [控制台](#)，在左侧菜单栏中进入[设备管理](#)，单击[自有设备](#)，您可以在该页面管理您的自有设备。



### 说明：

- **密钥：**建议优先选择使用提供的密钥。
- **License 授权有效期：**默认一次性授权，如需使用月度授权需要联系工作人员。
- **无流量包处理方式：**按量付费和截断加速。
  - 按量付费指流量包到期或用尽后继续使用，结算时用量按量计费，月底结算。
  - 截断加速指流量包到期或者用尽后直接阻断，无法继续使用。

单击**新建设备**，进入新建设备页面。在页面上依次填充信息，单击**保存**即可成功新建设备。

### 新建设备 ×

**ⓘ** 同一个产品下，设备名称需要保证唯一性

设备名称 \*   
支持英文，数字，@，.，:，下划线，-的组合，最多不超过 48 个字符

备注   
支持中文，英文，数字，下划线，-的组合，最多不超过 16 个字符

密钥  使用提供的密钥  使用自定义密钥

是否进行二次加密  是  否

接入环境 \*  公有云网关  自有网关  公有云网关和自有网关

License 授权有效期 \*  月度授权  一次性授权

无流量包处理方式 ⓘ \*  按量付费  截断加速

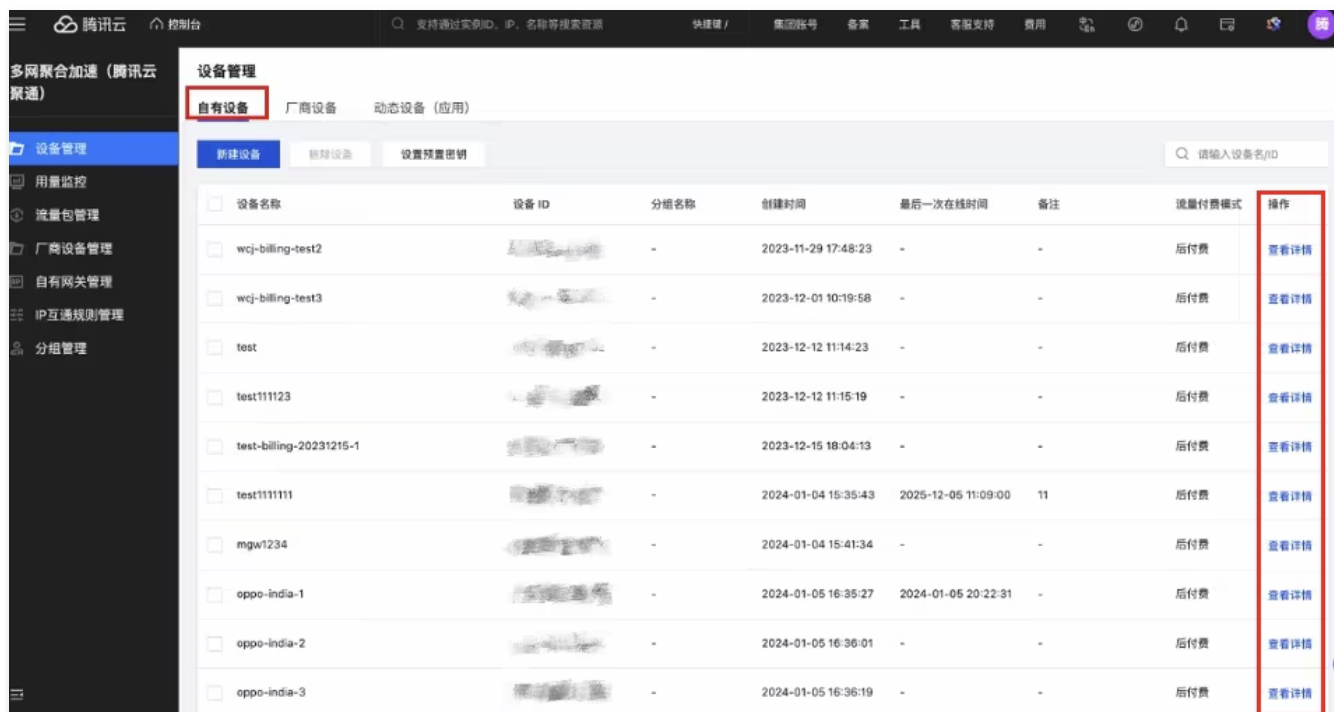
分组

#### ⚠ 注意：

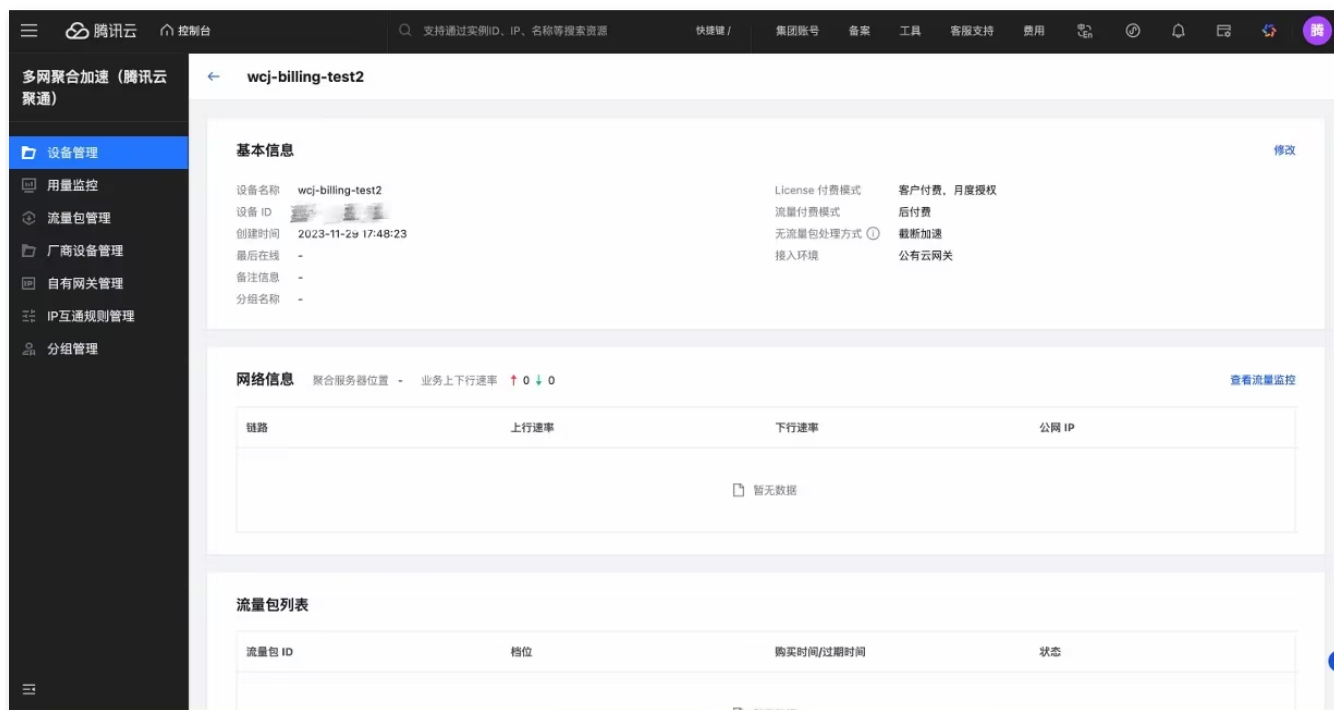
- 一次性授权的设备，需要付费后才能启动使用。
- 创建设备成功后，页面会返回该设备的密钥 dataKey；密钥需要妥善保存记录，后续若丢失或遗忘需要咨询工作人员查询。
- 多网聚合加速（腾讯云聚通）要求密钥一机一密，请保证每一个物理设备使用独立的密钥。

## 2. 查看设备详情

单击查看详情，进入该设备的详情页面。

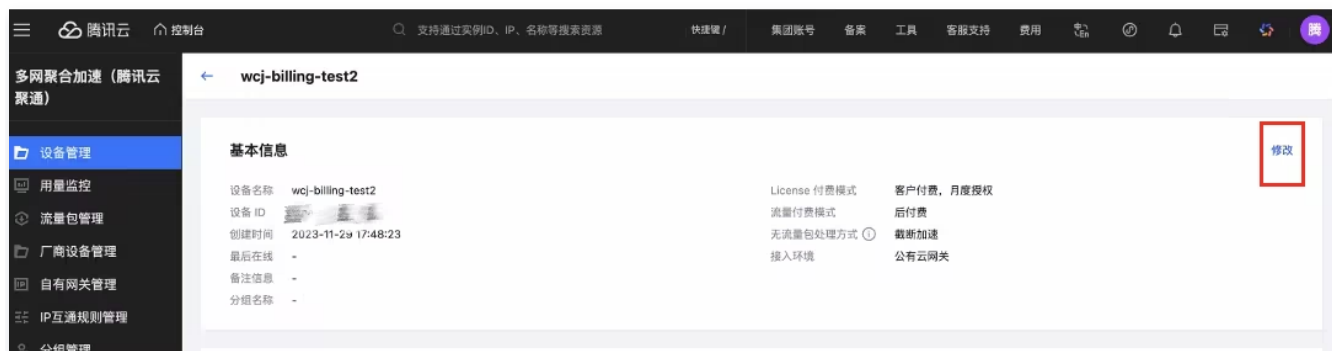


设备详情页面展示设备信息和设备相关的网络情况，展示聚合服务器节点和上下行速率统计，同时展示流量包列表。在设备管理 > 网络信息，单击查看实时监控，将跳转至该设备的实时流量监控页面。



### 3. 修改设备信息

进入设备管理，在基本信息页面中，单击修改，即可修改设备名称和备注字段。



## 4. 相关 SDK 接口

### 4.1 Android SDK

#### 说明:

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

#### setDataKey

设置设备密钥，用于自有设备认证鉴权。

```
public static void setDataKey(String uuid, String dataKey)
```

#### 参数说明:

参数	类型	描述
uuid	String	设备唯一标识（可复用云 API 返回的 deviceId）。
dataKey	String	设备密钥（每个 dataKey 对应一个设备，不能重复使用）。

### 4.2 iOS SDK

#### 说明:

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

#### setupDataKey

设置公有云注册参数。

```
func setupDatakey(_ datakey: String, uuid: String)
```

参数	描述
datakey	设备密钥
uuid	设备唯一标识

## 4.3 Linux SDK

### 说明：

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

调用如下接口配置加速的各项参数（仅配置参数不启动加速）：

```
curl -X 'POST' 'http://127.0.0.1:9801/api/v2/client/mp-speeder' -H
'accept: */*' -H 'Content-Type: application/json' -d '{
  "dataKey": "get this devicekey from tencent and replace it here",
  "interfaces": ["usb0", "usb1", "eth0"],
  "scheduleMode": "bonding"
}'
```

本接口传入 Client 实体，定义如下：

名称	类型	说明
dataKey	string	必选设备 dataKey。
interfaces	[string ]	必选。参与多网聚合的接口列表。支持指定优先级，范围0~255，数值越小，优先级越高，优先级与网卡以冒号间隔。如果不传入优先级，则默认为64。注：redundant 模式下不支持传入优先级。["eth1:100", "eth2"]。
scheduleMode	string	必选。默认加速模式。"bonding", "redundant", "rtc"。

## 4.4 Windows SDK

### 说明：

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

调用如下接口配置加速的各项参数（仅配置参数不启动加速）：

### datakey 模式

```
curl -X POST "http://127.0.0.1:9801/api/v2/client/mp-speeder" ^
-H "accept: */*" ^
-H "Content-Type: application/json" ^
-d "
{"serviceMode":0,"dataKey":"xxxxxxx","scheduleMode":"rtc","re
gisterEnv":-2,"tunInterfaceName":"mp_tun0","t2Probe":true}"
```

名称	类型	说明
dataKey	string	必选，设备 dataKey。

## 4.5 HarmonyOS SDK

### ⓘ 说明：

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

### setDataKey

用于自有设备，设置设备密钥，认证鉴权。

```
static setDataKey(uuid: string, dataKey: string)
```

参数	描述
uuid	设备唯一标识（可复用云 API 返回的 deviceId）。
dataKey	设备密钥（每个 datakey 对应一个设备，不能重复使用）。

# 设备管理（厂商设备管理）

最近更新时间：2026-06-16 10:08:30

腾讯云聚通产品的接入网关方式有以下两种，涉及三方角色：厂商、运营方、腾讯云。

接入方式	角色关系	认证方式
一机一密	设备厂商与运营方为同一角色，且具备将 DataKey 写入设备的能力。	DataKey 密钥认证
厂商设备 SN 号	<ul style="list-style-type: none"><li>设备厂商与运营方为不同角色（可选择由谁付费）。</li><li>设备厂商与运营方为同一角色，但不具备将 DataKey 写入设备的能力。</li><li>设备厂商与运营方为同一角色，希望通过 SN 号方式管理设备。</li></ul>	SN 号认证

当厂商和运营方为同一角色，且具备在硬件设备中写入 DataKey 的能力时，每台设备分配唯一密钥（一机一密），通过 DataKey 接入腾讯云多网聚合网关。

当厂商希望通过 SN 号方式接入网关时，适用以下场景：

- 厂商和运营方为不同角色（例如：T-Box 类设备），可选择由厂商或运营方独立付费；
- 厂商和运营方为同一角色，但不具备将 DataKey 写入设备的能力；
- 厂商和运营方为同一角色，具备写入能力，但主动选择 SN 号方式接入。

以上场景中，设备通过 SN 号接入多网聚合网关，可实现与 DataKey 接入完全相同的效果。

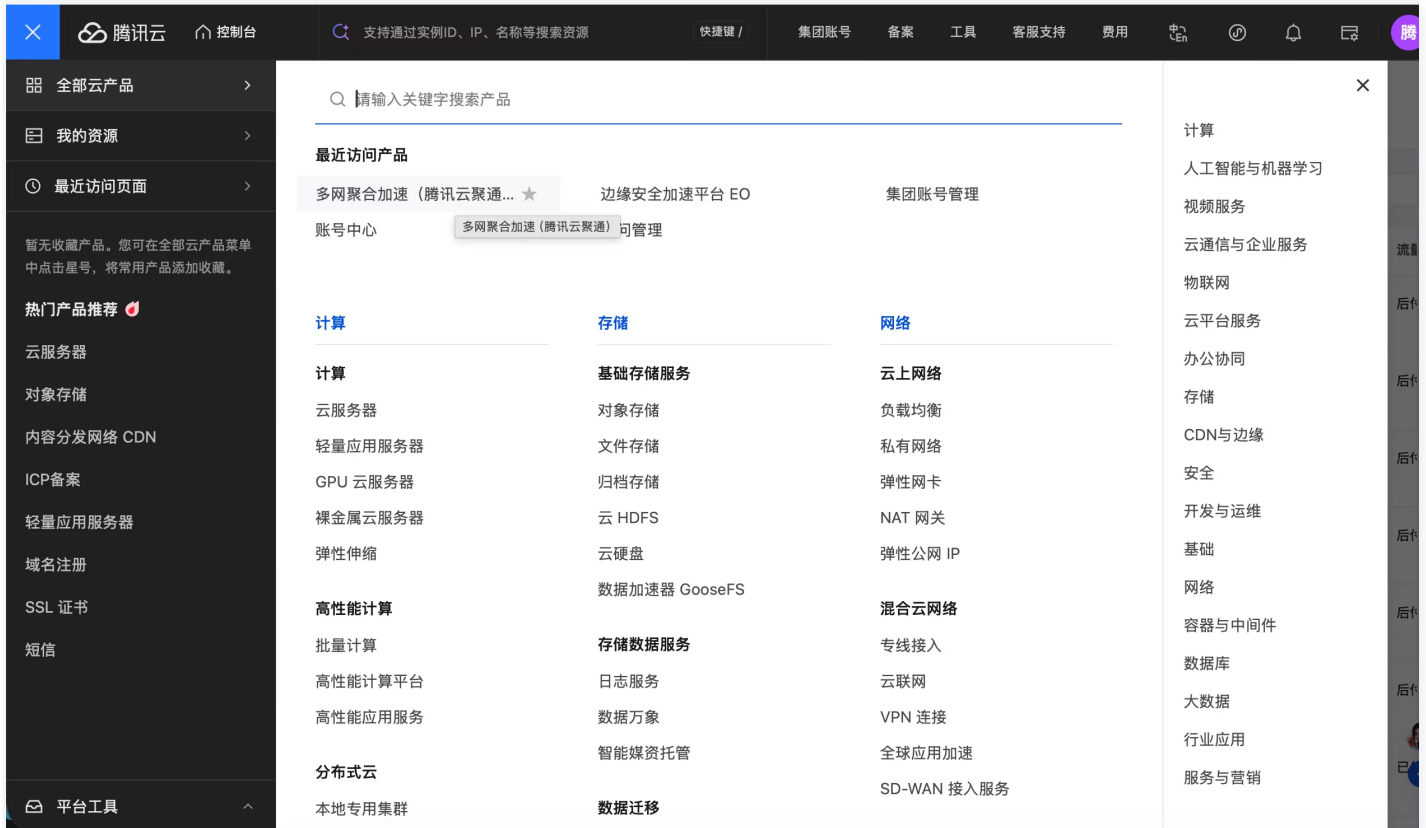
下文描述了如何在多网聚合控制台中，通过 SN 号完成厂商设备的注册与激活流程。

## 前置条件：

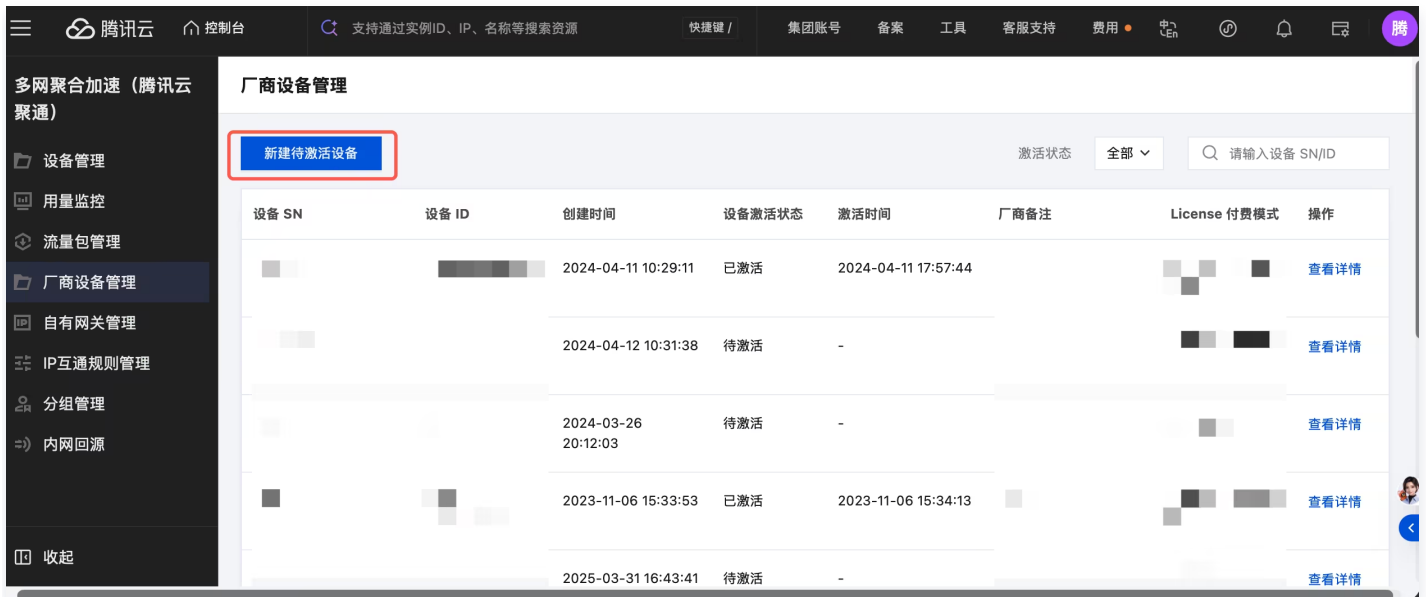
- 已获取厂商设备的 SN 号（设备序列号）。
- 已开通多网聚合控制台账号。
- 厂商提前注册备案。
- 厂商设备已内置我方 SDK 且处于可联网状态。

## 1. 厂商新建待激活设备（用厂商腾讯云账号）

登录多网聚合加速（腾讯云聚通）[控制台](#)。

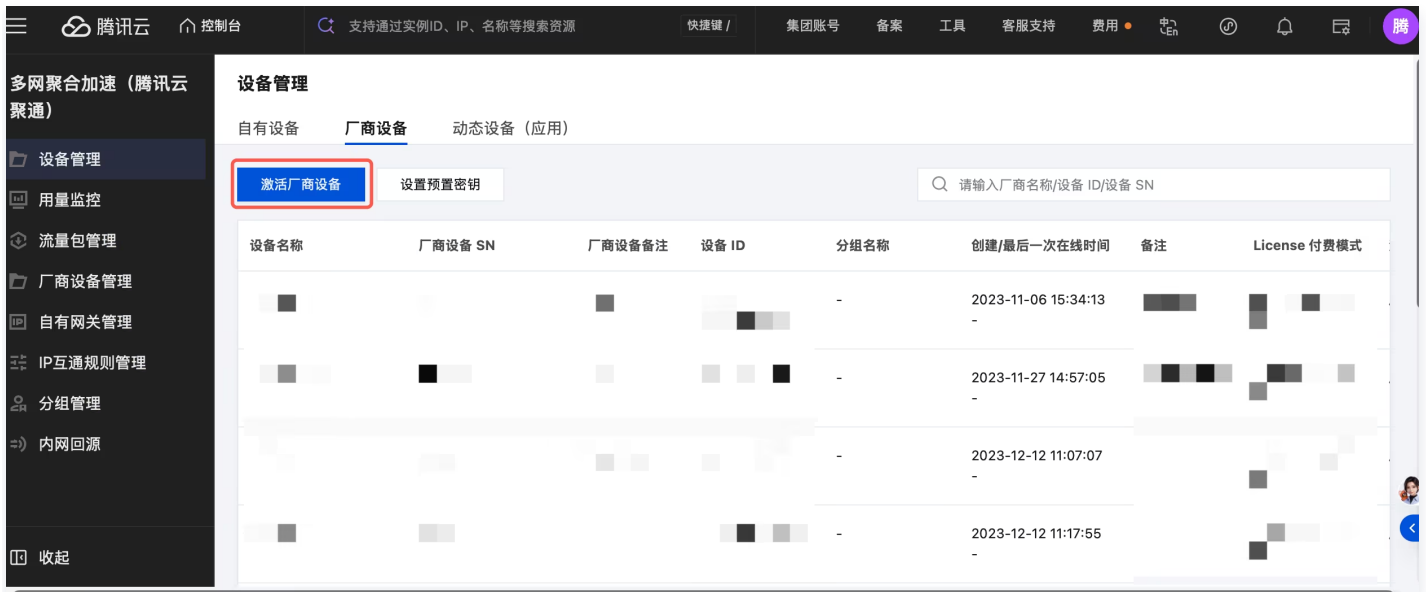


在左侧导航栏中，进入厂商设备管理模块，单击新建待激活设备。

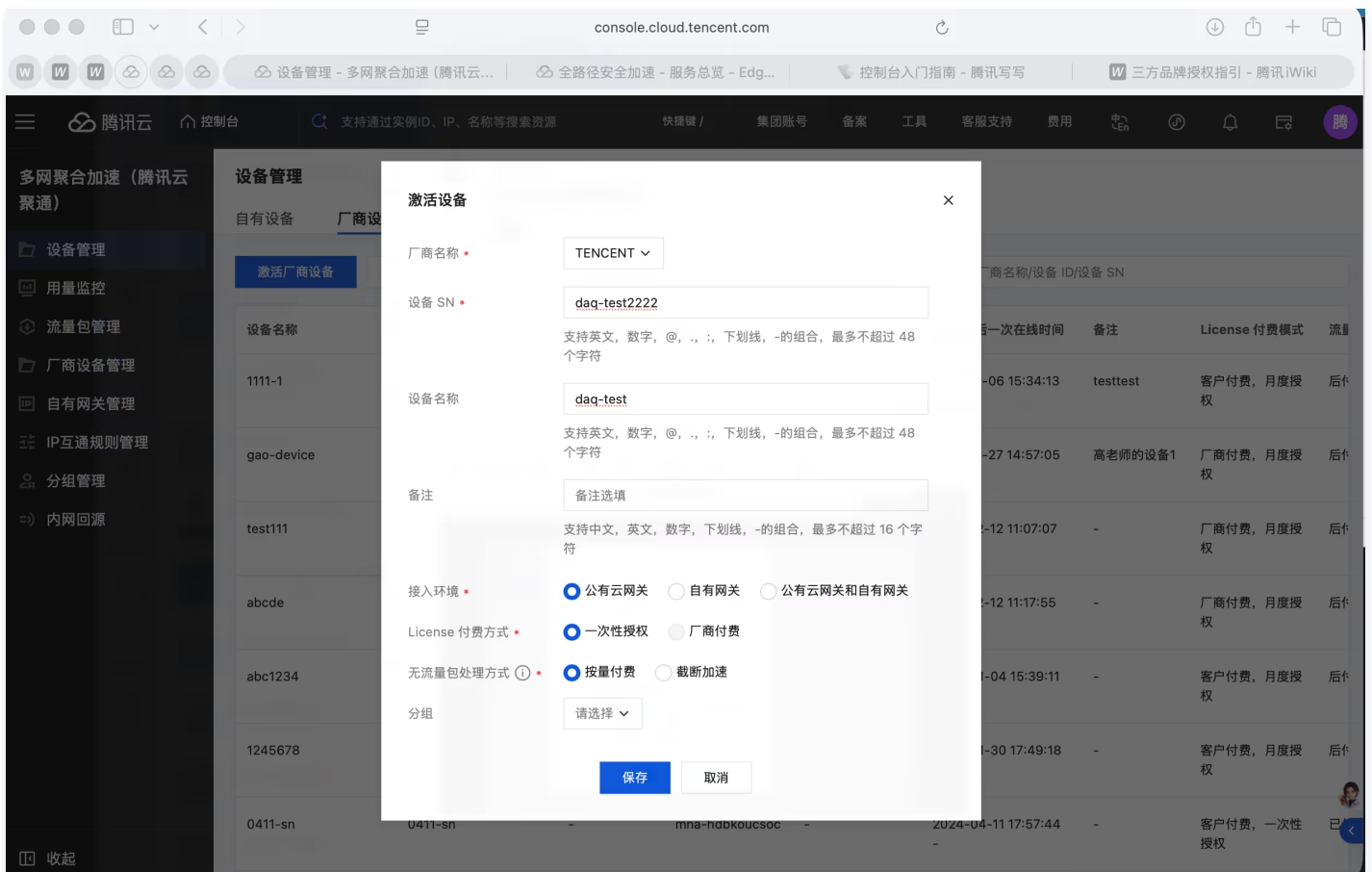


在弹出的表单中，填写以下信息（图为示例）。





在弹出的窗口中选择对应的厂商名称，输入设备 SN 号，系统将自动校验 SN 号的正确性，您可以自命名您的设备名称等，将带星号的内容选择完毕后，单击保存，激活成功，厂商可在控制台看到设备状态变更为“已激活”。



**注意：**

- 激活设备时填写的设备名称需要唯一，不可与已激活的设备同名。

- 付费模式：如果厂商创建设备时选择客户付费，则 License 付费方式只可选择一次性授权。如果厂商创建设备时已付费，则运营方在选择 License 付费方式时只可选择厂商付费。

付费成功后，设备即可通过 SN 号正常接入多网聚合网关。

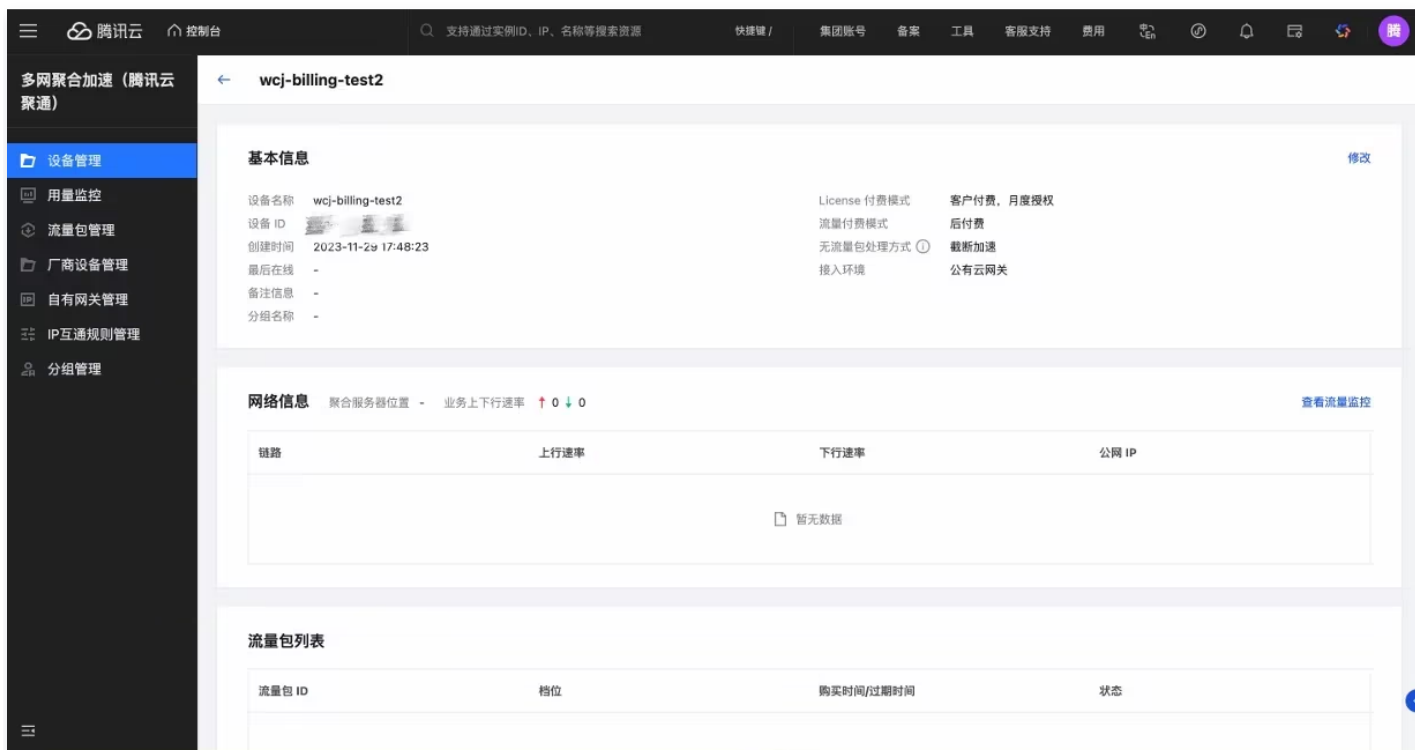
### 3. 查询厂商设备详情

在设备管理 > 厂商设备中选择设备，单击查看详情，可以进入该设备的详情页面。

设备名称	厂商设备 SN	厂商设备备注	设备 ID	分组名称	创建/最后一次在线时间	备注	License 付费模式	流量付费模式	操作
1111-1				-	2023-11-06 15:34:13		客户付费, 月度授权	后付费	查看详情
gao-device				-	2023-11-27 14:57:05		厂商付费, 月度授权	后付费	查看详情
test111				-	2023-12-12 11:07:07		厂商付费, 月度授权	后付费	查看详情
abcde				-	2023-12-12 11:17:55		厂商付费, 月度授权	后付费	查看详情
abc1234				-	2024-01-04 15:39:11		客户付费, 月度授权	后付费	查看详情
1245678				-	2024-01-30 17:49:18		客户付费, 月度授权	后付费	查看详情
0411-sn				-	2024-04-11 17:57:44		客户付费, 一次性授权	后付费	查看详情
test-11111				-	2024-05-06 20:07:20		厂商付费, 月度授权	后付费	查看详情

详情页面展示设备信息和设备相关的网络情况，展示聚合服务器节点和上下行速率统计，同时展示流量包列表。

进入设备管理 > 网络信息，单击查看实时监控，将跳转至该设备的实时流量监控页面。



## 4. 修改厂商设备信息

在基本信息卡片中，单击**修改**，即可修改设备名称和备注字段。



## 5. 相关 SDK 接口

### Linux SDK

**说明：**

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

调用如下接口配置加速的各项参数（仅配置参数不启动加速）：

```
curl -X 'POST' 'http://127.0.0.1:9801/api/v2/client/mp-speeder' -H
'accept: */*' -H 'Content-Type: application/json' -d '{
```

```
"vendor": "get vendor from tencent and replace it here",
"sn": "get sn from tencent and replace it here",
"interfaces": ["usb0", "usb1", "eth0"],
"scheduleMode": "bonding"
}'
```

本接口传入 Client 实体，定义如下：

名称 ( name )	类型 ( type )	说明
vendor	string	必选，设备厂商型号。
sn	string	必选，设备厂商提供的设备唯一序列号。
interfaces	[string]	必选。参与多网聚合的接口列表。支持指定优先级，范围0~255，数值越小，优先级越高，优先级与网卡以冒号间隔。如果不传入优先级，则默认为64。注：redundant 模式下不支持传入优先级。["eth1:100", "eth2"]。
scheduleMode	string	必选。默认加速模式。"bonding", "redundant", "rtc"。

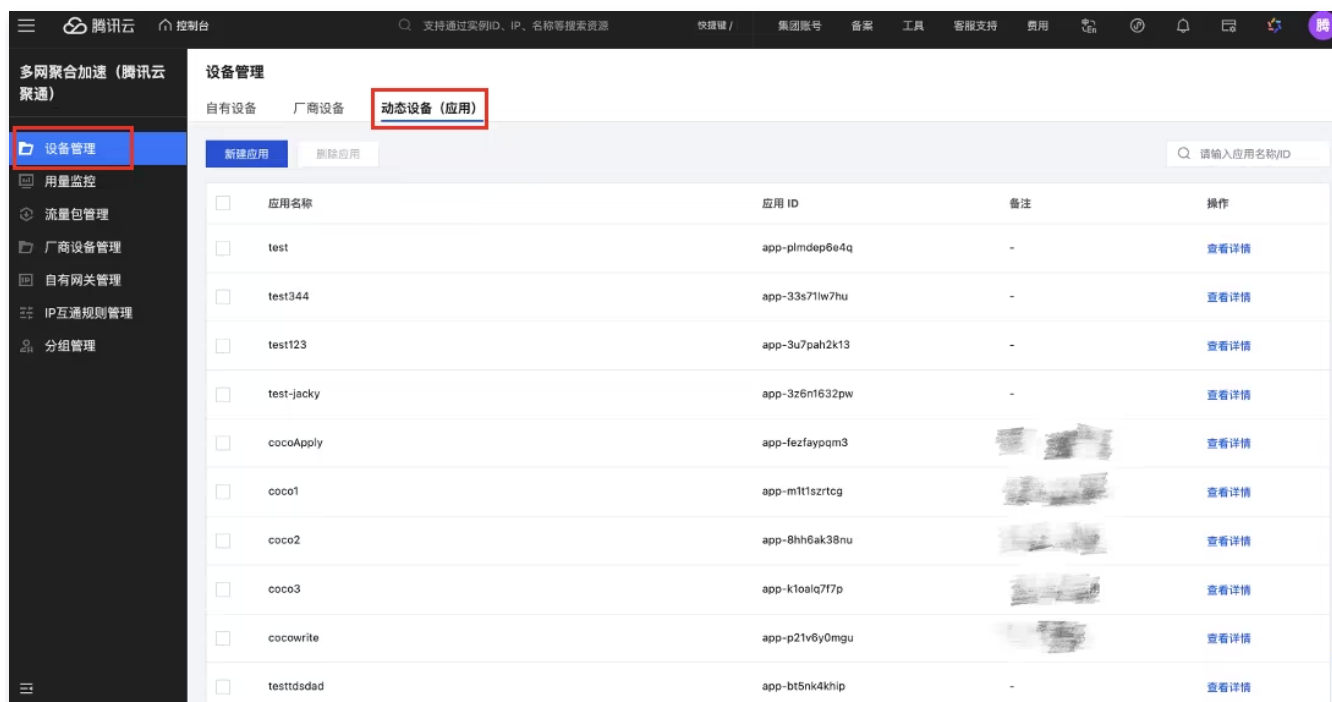
# 设备管理（动态设备/应用）

最近更新时间：2026-06-17 10:00:00

**说明：**  
动态设备仅支持手机端（安卓、iOS、HarmonyOS）。

## 1. 新建应用

在左侧菜单栏中，单击**设备管理**，选择**动态设备（应用）**列表，进入动态应用管理页面。



单击**新建应用**，进入新建应用页面。在页面上输入应用名称，依次填充和设置信息，单击**保存**即可成功新建应用。

### 新建应用 ✕

i 同一个产品下，应用名称需要保证唯一性

应用名称 \*

支持英文，数字，@，.，:，下划线，-的组合，最多不超过 48 个字符

备注

支持中文，英文，数字，下划线，-的组合，最多不超过 16 个字符

密钥  使用提供的密钥  使用自定义密钥

接入环境 \*  公有云网关  自有网关  公有云网关和自有网关

保存
取消

! **说明：**

- 密钥：建议优先选择**使用提供的密钥**。
- 接入环境：客户可以选择拟接入的网关类型。
- 创建设备/应用成功后页面会返回该设备/应用的密钥 dataKey，请妥善保存记录，后续若丢失遗忘请[联系我们](#) 查询。

通过 **SDK 集成接入** 的客户需要进行**签名操作**，分为 Java、Objective-C 和 Go 语言版本，请参照下方介绍。记录控制台展示出的应用 ID 与应用密钥，以助后续按应用创建的设备 Sign 签名的生成与鉴权。

### 添加结果 ✕

✓ 已成功添加新应用，请查看应用密钥！

应用 ID    app-████████sdg 🔗

应用密钥    X10 █████████f97O4KgD █████████zA2fzz █████████c= 🔗

注意：请妥善保管您的应用密钥，避免泄露风险。

开始管理应用
返回应用列表页

## 生成按应用创建的设备 Sign 签名

按应用创建的设备 Sign 签名的生成需要两个参数：客户自定义业务控制设备唯一标识即 deviceName；上图中创建成功后返回的应用密钥。利用两个参数启动脚本，完成按应用创建的设备 Sign 签名生成。

- Java 语言

```
package com.android.tencentvpn.util;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import java.security.Key;
import java.util.Date;
import javax.crypto.spec.SecretKeySpec;

public class SignGenerate {
    private static final long EXPIRE_TIME = 72 * 60 * 60 * 1000; // 过期
    时间，单位毫秒
    /**
     * @param deviceName 业务自己控制设备唯一标识
     * @param SecretKey 云api返回的 应用密钥
     * @return 多网sdk 需要传入的签名字符串 最终通过setSign 调用
     */
    public static String generateSign(String deviceName, String
    SecretKey) {
        try {
            Date now = new Date();
            Date expireDate = new Date(now.getTime() + EXPIRE_TIME);
            Key key =
                new SecretKeySpec(SecretKey.getBytes(),
                SignatureAlgorithm.HS256.getJcaName());
            return Jwts.builder()
                .claim("deviceName", deviceName)
                .expiration(expireDate)
                .signWith(key)
                .compact();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "";
    }
}
```

```
}
```

- Objective-C 语言

```
//podfile中添加pod 'JWT', 集成JWT库
// .h文件:
@interface SignGenerator : NSObject
/**
 * @deviceName 业务自己控制设备唯一标识
 * @param SecretKey 云api返回的 应用密钥
 * @return 多网sdk 需要传入的签名字符串 最终通过setSign 调用
 */
+ (NSString *)generateSignWithDeviceName:(NSString *)deviceName
secretKey:(NSString *)secretKey;
@end
// .m文件
#import "SignGenerator.h"
#import <JWT/JWT.h>
@implementation SignGenerator
+ (NSString *)generateSignWithDeviceName:(NSString *)deviceName
secretKey:(NSString *)secretKey {
    // 1. 设置过期时间（72小时后）
    NSDate *currentDate = [NSDate date];
    NSDate *expireDate = [currentDate dateByAddingTimeInterval:72 *
60 * 60];
    // 2. 构建 Claims（负载）
    NSDictionary *claims = @{
        @"deviceName": deviceName,
        @"exp": @((long)[expireDate timeIntervalSince1970]) //
Expiration
    };
    NSData *keyData = [secretKey
dataUsingEncoding:NSUTF8StringEncoding];
    // 3. 使用 HS256 算法和密钥签名
    JWTBuilder *builder = [JWTBuilder encodePayload:claims]
.secretData(keyData)
.algorithmName(@"HS256"); //算法名
    NSString *jwt = builder.encode;
    // 4. 错误处理
```

```
    if (builder.jwtError) {
        NSLog(@"JWT 生成失败: %@", builder.jwtError);
        return @"";
    }
    return jwt;
}
@end
```

## • Go 语言

```
package main
import (
    "fmt"
    "github.com/dgrijalva/jwt-go/v4"
    "time"
)
func main() {
    sign, expireTime, err := GenerateSign("tencent-test")
    if err != nil {
        panic(err)
    }
    println(sign, expireTime)
}
// SignClaims JwtCustomClaims
type SignClaims struct {
    jwt.StandardClaims
    DeviceName string `json:"deviceName"`
}
var (
    ExpireTime = 72
    // 过期时间, 单位小时
    SecretKey = []byte("shCrVLKq/Hp6UdtVpw81Bqu6ZDcLwOABf2Az67e/pjI=")
    // APP密钥
)
// GenerateSign 生成签名
func GenerateSign(deviceName string) (string, int64, error) {
    expireTime := time.Now().Add(time.Duration(ExpireTime) * time.Hour)
    claims := &SignClaims{
        StandardClaims: jwt.StandardClaims{
```

```
        NotBefore: jwt.Now(),
        ExpiresAt: jwt.At(expireTime),
    },
    DeviceName: deviceName,
}
println(fmt.Sprintf("%#v", claims))

token := jwt.NewWithClaims(jwt.SigningMethodHS256, claims)
sign, err := token.SignedString(SecretKey)
if err != nil {
    return "", 0, err
}
return sign, expireTime.Unix(), nil
}
```

## • ArkTS 语言

```
// 依赖 @ohos/jsonwebtoken 库用于生成 JWT 签名
// "dependencies": {
//     "@ohos/jsonwebtoken": "1.0.2"
// },

import jwt from '@ohos/jsonwebtoken';
import { JwtPayload } from '@ohos/jsonwebtoken';

const SIGN_EXPIRE_TIME = 72 * 60 * 60 * 1000; // 过期时间, 72小时, 单位毫秒

function generateSign(deviceName: string, secretKey: string): string {
    try {
        let exp = Math.floor((Date.now() + SIGN_EXPIRE_TIME) / 1000);

        let payload: JwtPayload = {
            deviceName: deviceName,
            exp: exp
        };

        let token: string = jwt.sign(payload, secretKey);
```

```

hilog.info(DOMAIN, TAG, "generateSign success, jwt=%{public}s",
token);
return token;
} catch (e) {
hilog.error(DOMAIN, TAG, "generateSign error: %{public}s",
JSON.stringify(e));
return "";
}
}

```

### 设置应用签名鉴权

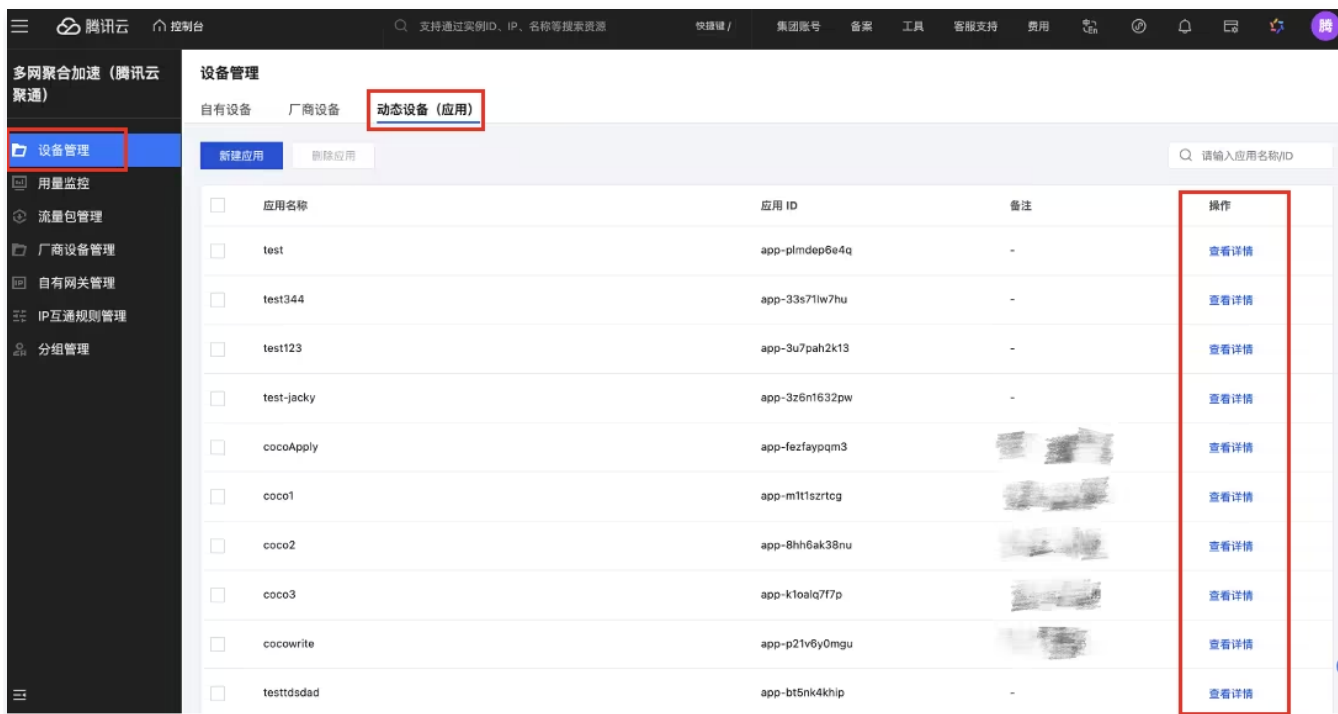
调用接口 setSign，传入创建应用时记录的 appId 即应用 ID 和 sign 即签名字符串，完成按应用创建的设备，设置应用签名鉴权。

#### ⚠ 注意：

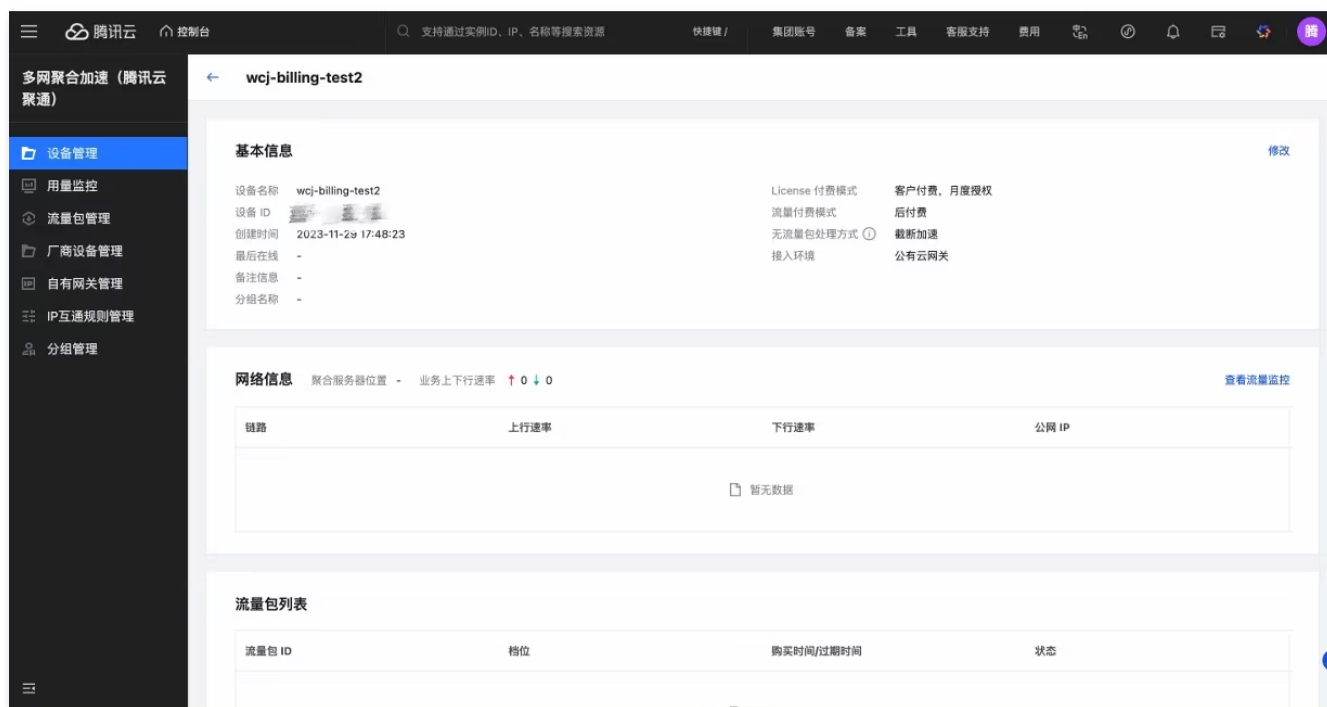
1. 签名建议在服务端生成下发到终端，不建议把应用 ID，应用密钥等在设备端暴露和存储，避免安全风险。
2. 设备名称 deviceName 请严格保持每个物理设备和 deviceName 一一对应。按应用鉴权方式下，设备用量监控/流量统计需要以 deviceName 作为索引获取。多设备混用 deviceName 或者一个设备使用多个 deviceName 会导致查询异常。

## 2. 查看动态设备/应用详情

单击查看详情，进入该设备/应用的详情页面。

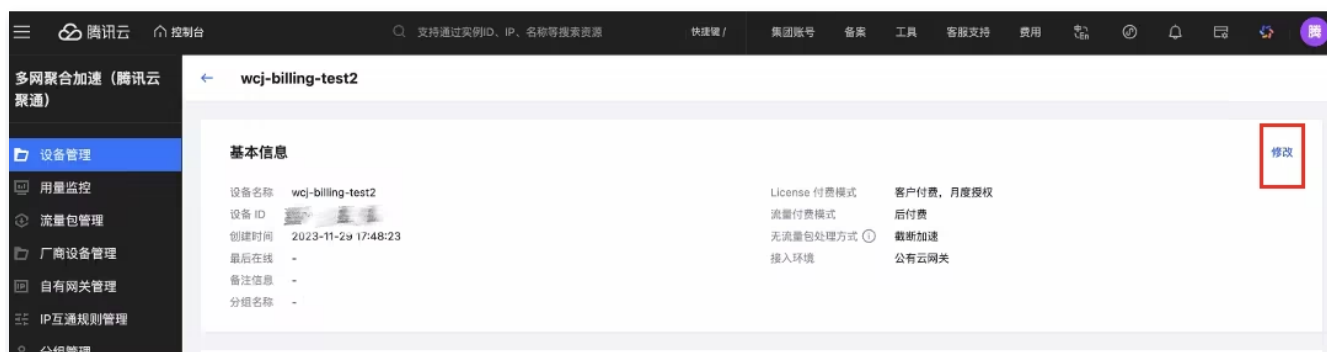


详情页面展示设备信息和设备相关的网络情况，展示聚合服务器节点和上下行速率统计，同时展示流量包列表。进入**设备管理 > 网络信息**，单击**查看实时监控**，将跳转至该设备的实时流量监控页面。



### 3. 修改动态设备/应用信息

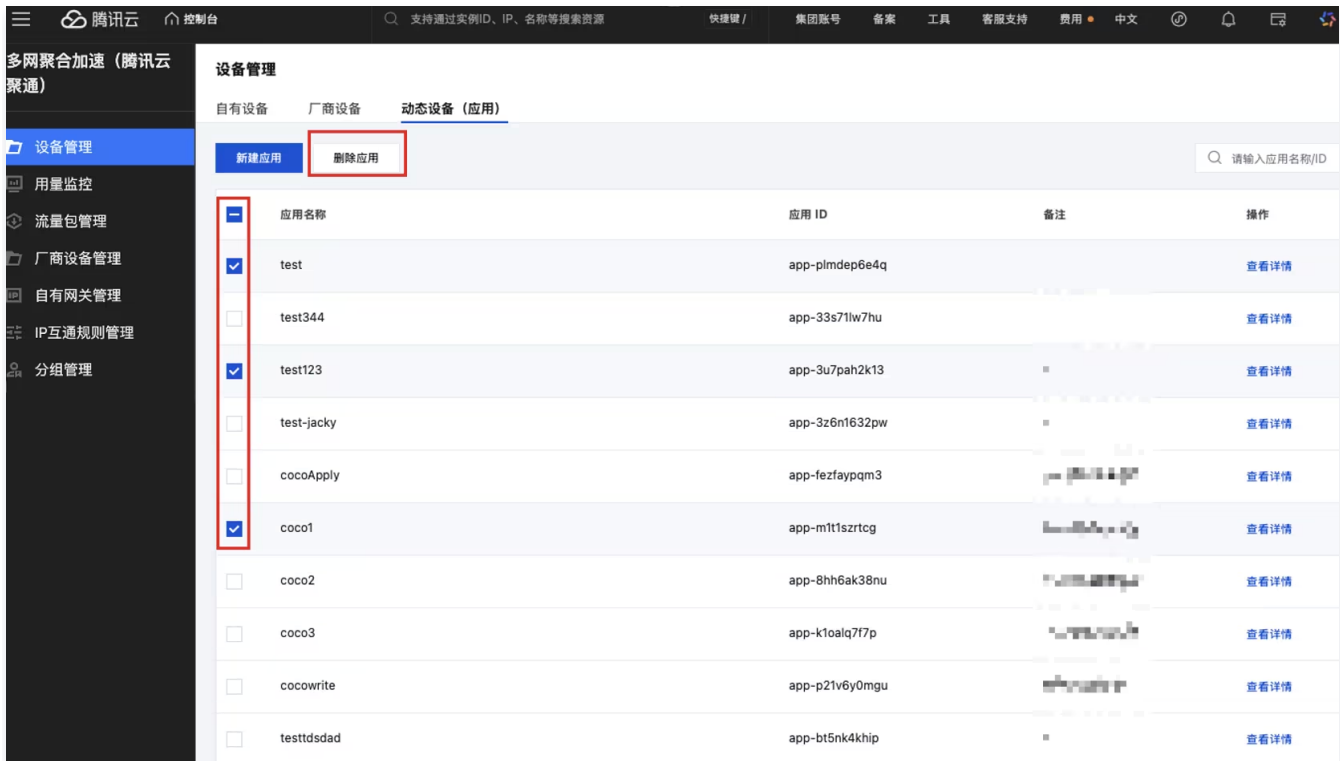
在应用详情页面基本信息卡片中，单击**修改**，即可修改设备名称和备注字段。



### 4. 删除应用

您可以在管理设备页面删除您的设备/应用，删除操作支持单选设备和多选设备。

选中需要删除的设备前方的勾选框，在列表上方，单击**删除设备**，弹出二次确认弹窗，单击**确认**后删除设备。



**注意:**

设备/应用删除后，该设备的密钥将会失效，已经产生的 License 授权费用和流量费用不会减免，将按照正常的出账周期进行结算。

## 5.相关 SDK 接口

### 5.1 Android SDK

**说明:**

SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

#### setSign

设置应用签名鉴权，用于按应用创建的设备。

```
public static void setSign(String appId, String sign)
```

**参数说明:**

参数	类型	描述
appId	String	应用 appId

sign	String	签名字符串
------	--------	-------

## 5.2 iOS SDK

**说明：**  
SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

### setSign

设置应用签名鉴权。

```
func setSign(appId: String, sign: String)
```

参数	描述
appId	应用 ID
sign	签名

## 5.3 HarmonyOS SDK

**说明：**  
SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

### setSign

用户按应用创建的设备，设置应用签名鉴权。

```
static setSign(appId: string, sign: string)
```

参数	描述
appId	应用 appId
sign	签名字符串

## 5.4 Windows SDK

**说明：**  
SDK 下载和详细文档请参考 [API 概览](#)，此处只列举本文相关功能对应的特定接口。

## 配置加速参数

在启动加速进程前，vendor 厂商需要收集必选参数，在启动 sdk 加速时需要作为参数携带。

- appId 和 appSign：app 模式注册，在 [控制台](#) 获取。
- scheduleMode：默认工作模式为 rtc。

调用如下接口配置加速的各项参数（仅配置参数不启动加速）：

```
curl -X POST "http://127.0.0.1:9801/api/v2/client/mp-speeder" ^
-H "accept: */*" ^
-H "Content-Type: application/json" ^
-d "{\"serviceMode\": 0, \"appId\": \"app-
0eoo3vpctx\", \"appSign\": \"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXZp
Y2VOYW11IjoiamFja3ktdGVzdC0yMC0xIn0.TleKYrzBL1dyp2i8WTBgs8Y8UvBJQv-
n2A2BRMT1xuQ\", \"scheduleMode\": \"rtc\"}"
```

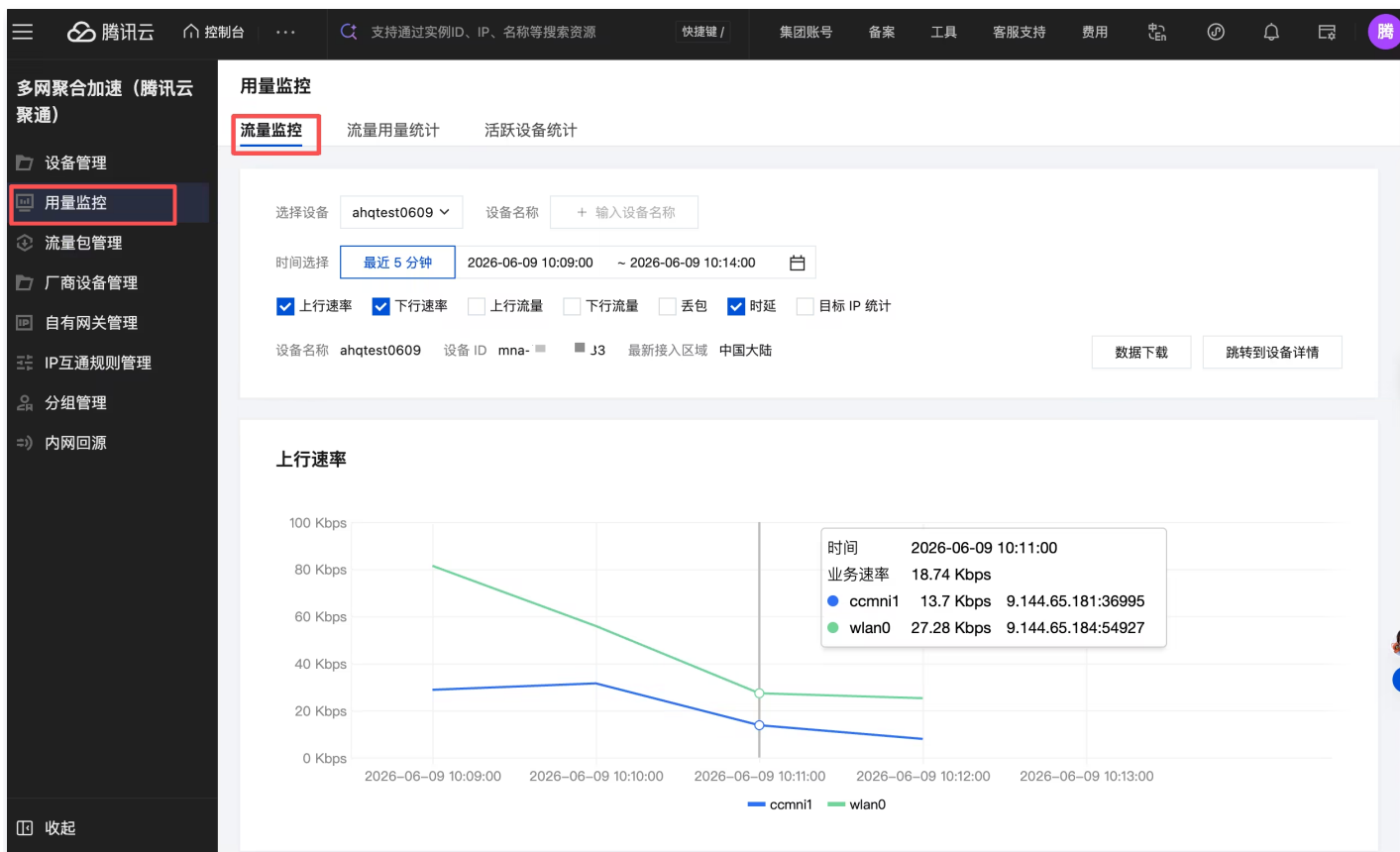
本接口传入 Client 实体，定义如下：

名称	类型	说明
appId	string	app 模式注册下使用。
appSign	string	app 模式注册下使用。
scheduleMode	string	可选：Windows 默认加速模式 "rtc"。

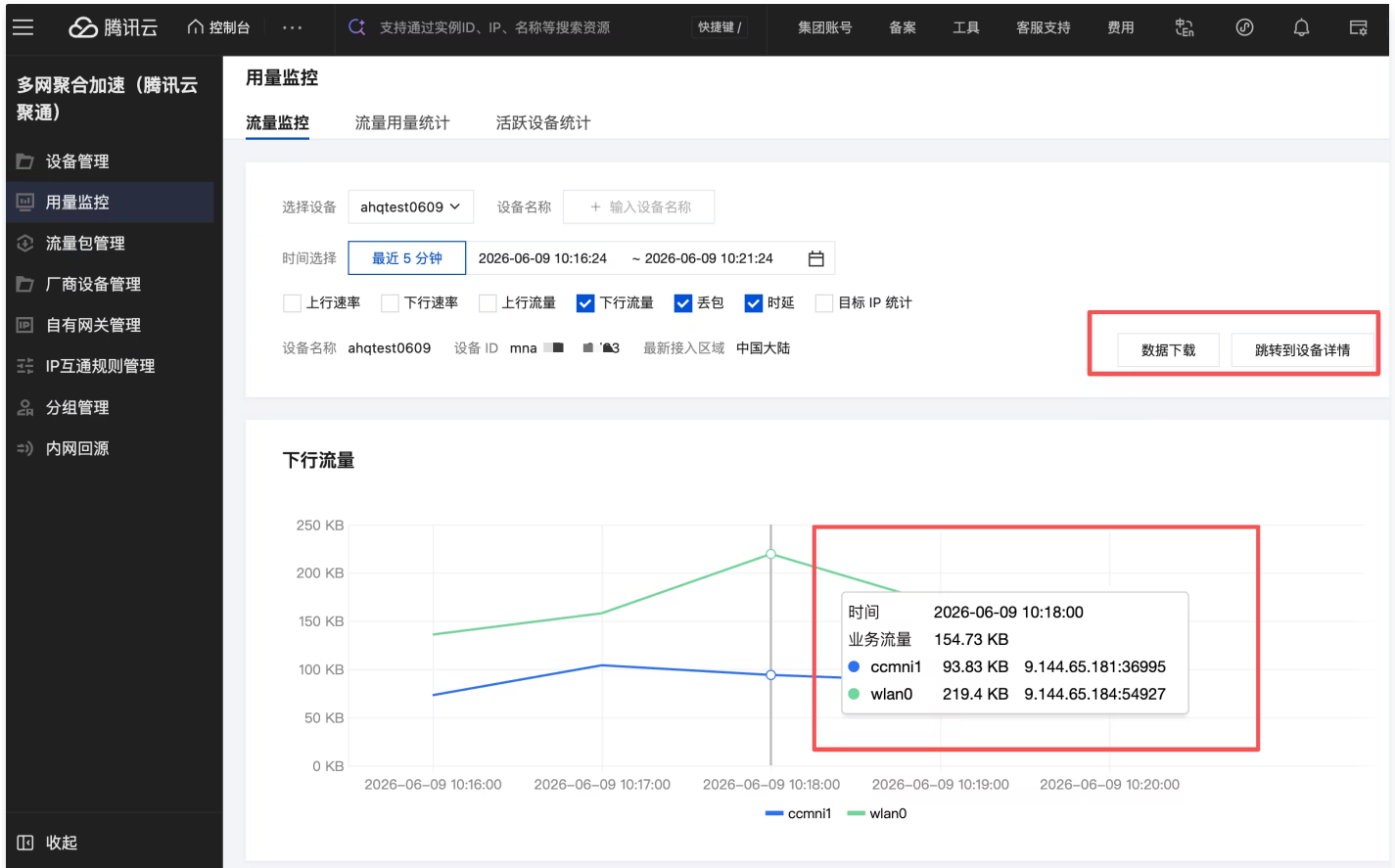
# 流量监控

最近更新时间：2026-06-09 16:10:00

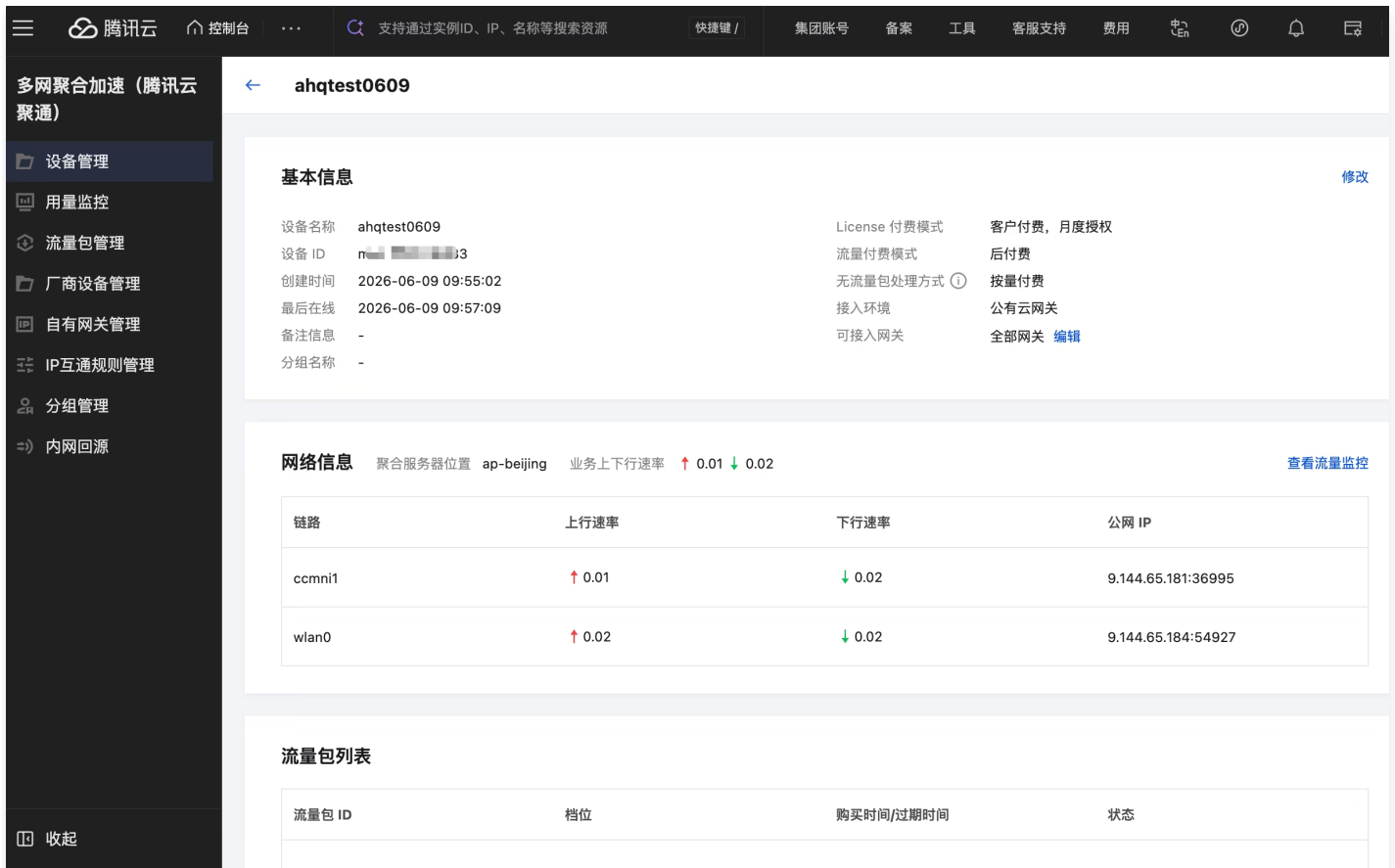
进入 [控制台](#)，在**用量监控**页面选择流量监控，可以按设备选择想要查看的设备。根据需要选择合适的时间和粒度，可以看到对应设备的流量监控。流量监控参数可勾选上行速率、下行速率、上行流量、下行流量、丢包、时延和目标 IP 统计。单击**数据下载**可以查看和下载具体数据。



拖动鼠标放在折线图具体某一点上即可显示该时间点的具体数据，如下图所示：



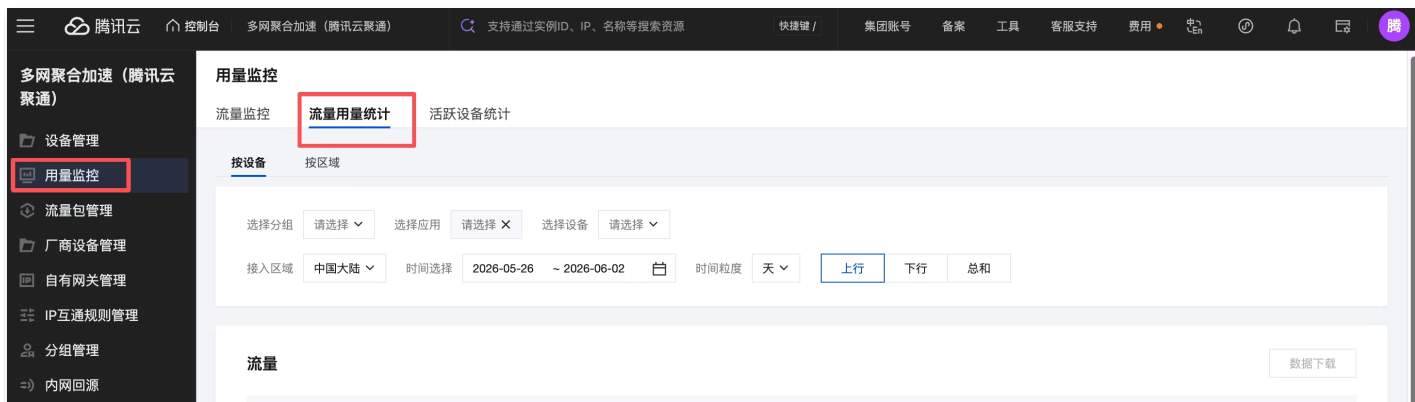
单击跳转到设备详情可以直接跳转到对应设备的设备管理页面，如下图所示：



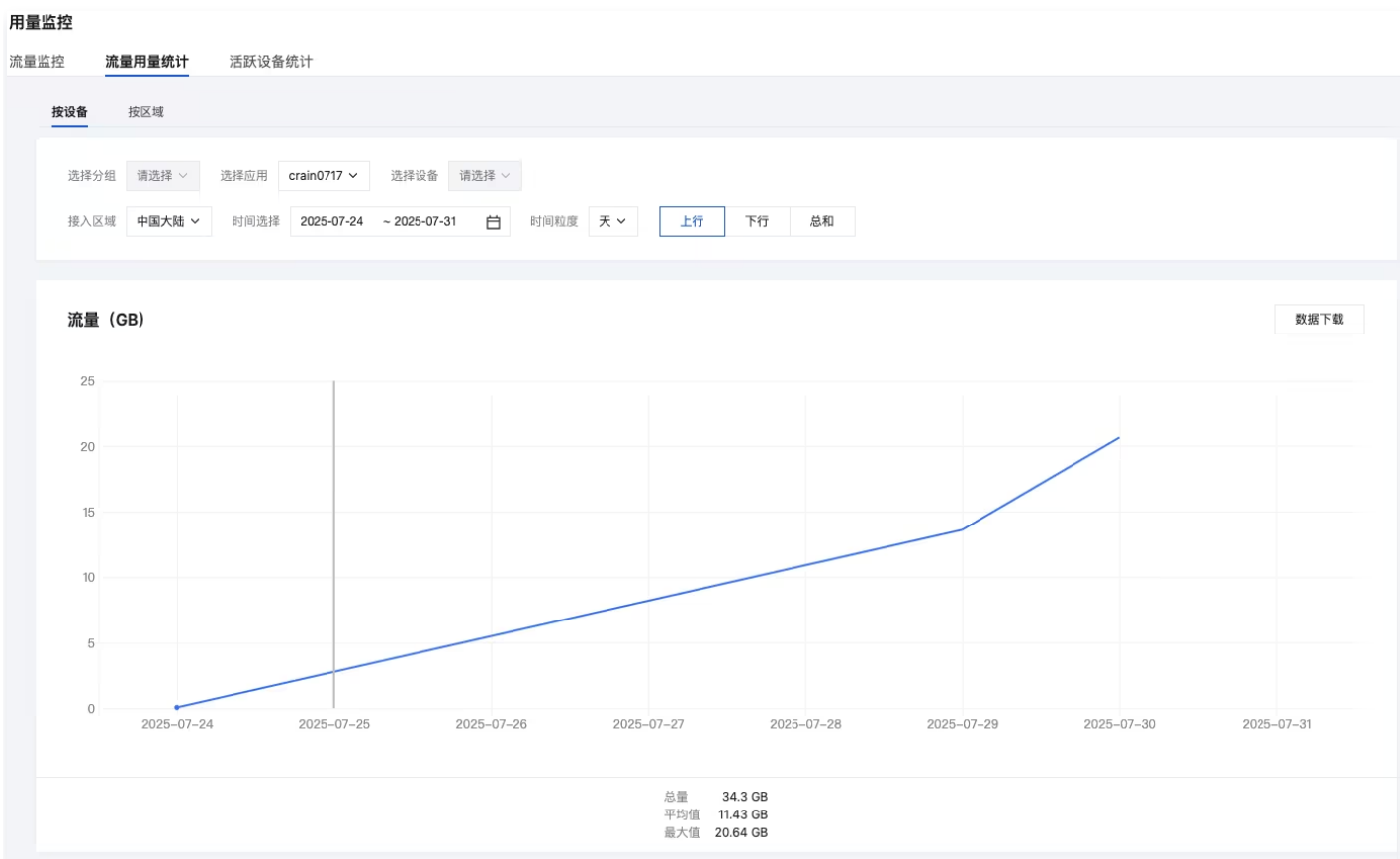
# 流量用量统计

最近更新时间：2026-06-09 16:10:00

进入 [控制台](#)，在 [用量监控](#) 页面选择 [流量用量统计](#)，可以按设备或按区域选择想要查看的设备。根据需要选择合适的时间和粒度，可以看到对应设备的流量用量统计。



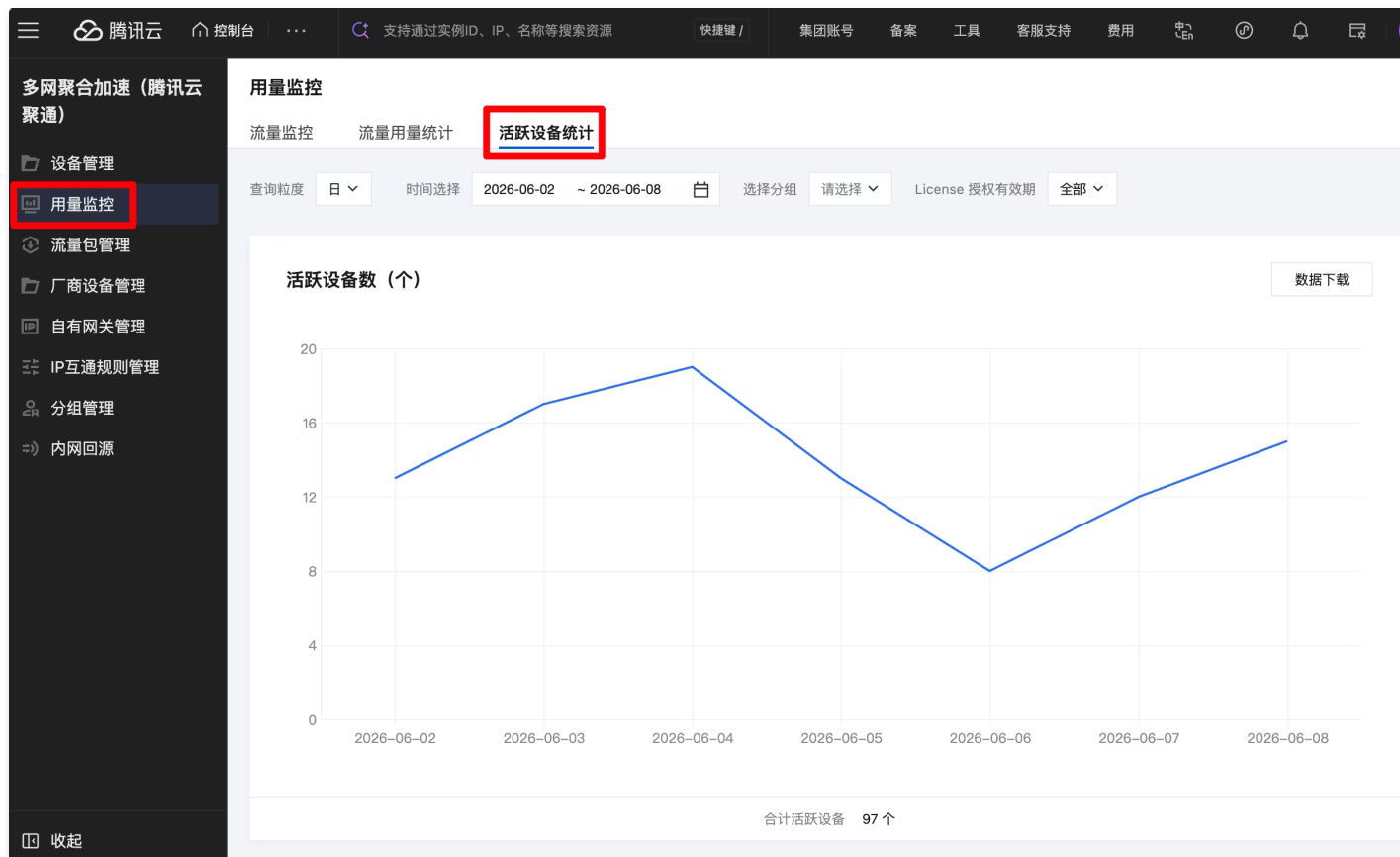
在折线图上拖动鼠标可查看任意时间点的具体数据，单击 [数据下载](#) 即可导出当前正在展示的数据内容。流量用量统计情况如下图所示：



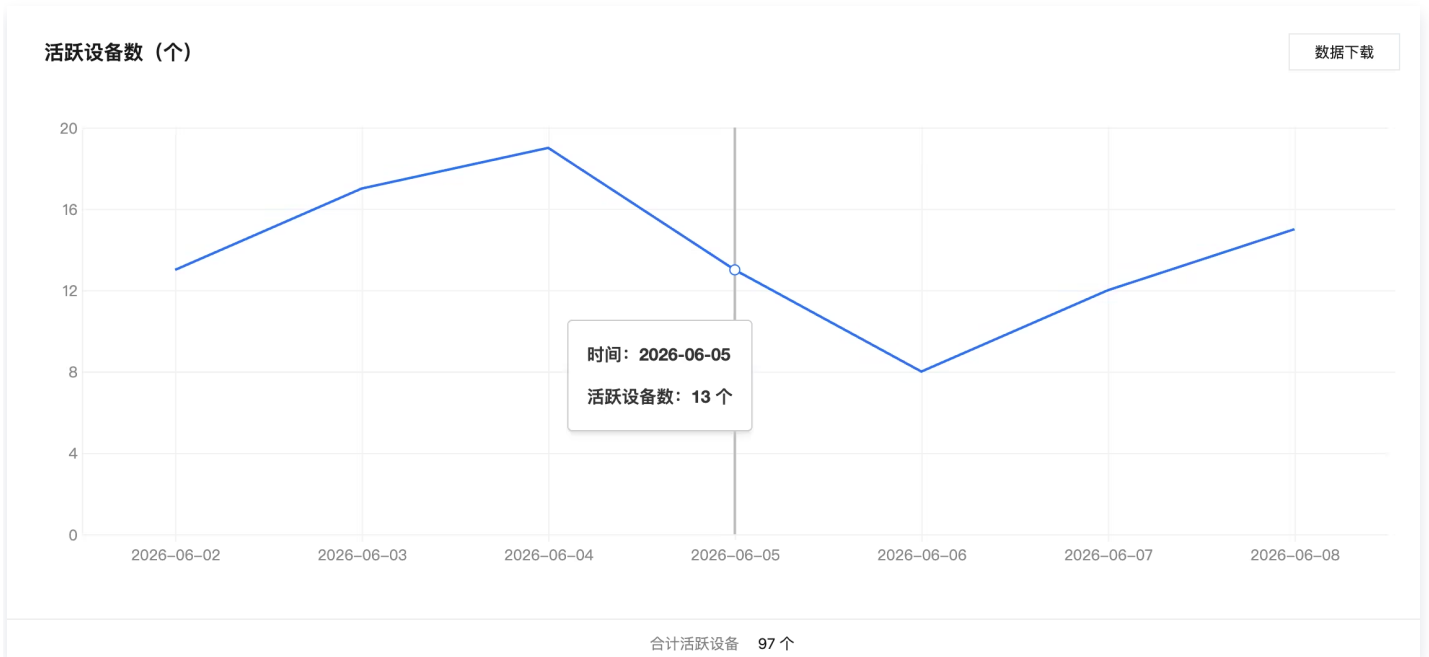
# 活跃设备统计

最近更新时间：2026-06-09 16:10:00

进入 [控制台](#)，在 [用量监控](#) 页面选择 [活跃设备统计](#)，可以按设备或按区域选择想要查看的设备。根据需要选择合适的时间和查询粒度，可以看到活跃设备数的折线图。



拖动鼠标放在折线图具体某一点上即可显示该时间点的具体活跃设备数。

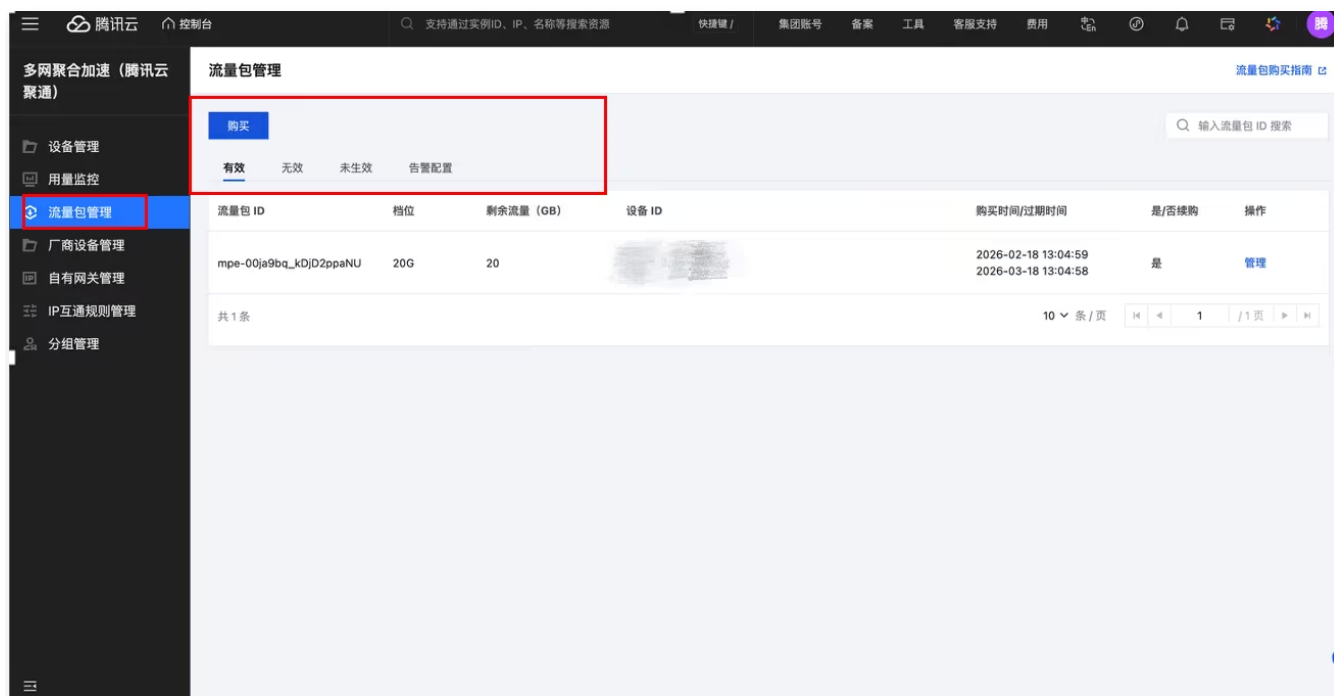


# 流量包相关

最近更新时间：2026-06-04 10:57:00

## 流量包管理

多网聚合加速（腾讯云聚通）支持通过控制台查看账号下的流量包使用情况，您可单击左侧栏**流量包管理**，在流量包列表中可查看有效流量包、无效流量包和未生效流量包及告警配置。



上图页面中各信息项的说明，如下图所示：

信息项	说明
流量包 ID	成功购买流量包后系统会自动生成唯一的流量包 ID。
档位	购买的流量包内包含的资源档位。
剩余流量（GB）	剩余流量用量，您可以直观地看到流量包的用量和余量情况。
设备 ID	购买流量包后绑定的设备，创建设备时系统会自动生成唯一的设备 ID。
购买时间/过期时间	成功购买流量包的时间/当前流量包失效的时间，一般为下个月的同一天。
是/否续购	购买流量包时是否选择自动续费（流量包到期后自动续费）。
操作	进入流量包管理配置页面，可进一步修改流量包的详细信息，包括绑定的设备、是否自动续购以及变更流量包档位。

## 告警配置

在**流量包管理 > 告警配置**页面，单击**告警配置**，可以设置告警策略，自定义配置告警阈值。



多网聚合加速（腾讯云聚通）支持设置流量包告警配置，当流量包余量消耗至告警线时，将收到告警提醒。

- 若当前流量包余量为设置的阈值时，将收到告警提示：“流量包 ID: xxxx 余量已不足 xx% 即将耗尽，如有自动续费将在耗尽后自动续购，自动续费可随时取消”。
- 若当前购买的流量包完全使用完时，将再次收到告警提示：“流量包 ID: xxxx 已经耗尽，如有自动续费将在到期后自动续购”。

## 流量包购买

单击**购买**，进入流量包购买界面。



选择**流量规格**、**设备**和**是否自动续费**，然后单击右下角**立即购买**，完成**流量包**购买。

## 无效流量包查看

单击**无效**，查看**过期**或**使用完**的流量包。

腾讯云 控制台
支持通过实例ID、IP、名称等搜索资源
快捷操作 / 集团账号 备案 工具 客服支持 费用

多网聚合加速 (腾讯云聚通)

- 设备管理
- 用量监控
- 流量包管理
- 厂商设备管理
- 自有网关管理
- IP互通规则管理
- 分组管理

### 流量包管理

购买

有效
  **无效**
 未生效
 告警配置

输入流量包 ID 搜索

流量包 ID	档位	剩余流量 (GB)	设备 ID	购买时间/过期时间	是/否续购	操作
live-j7a000fcqnQzhlv	20G	20	mna-1fay51ul54	2023-11-29 16:21:28 2023-12-29 16:21:29	否	<a href="#">管理</a>
live-j7c000fczgjP37m	20G	20	mna-6yt9ft8bs	2023-12-01 10:17:13 2024-01-01 10:17:40	否	<a href="#">管理</a>
live-j7c000fcziXPrh2	50G	50	mna-zcdrhy300i mna-b1w0wahe6x	2023-12-01 10:21:21 2024-01-01 10:22:02	否	<a href="#">管理</a>
live-j7m000f44ugC2s	20G	20	mna-llhdd9jyqb	2023-12-11 17:13:26 2024-01-11 17:13:10	否	<a href="#">管理</a>
live-j7q000fzuj75k1	20G	20	mna-edu18inwfg	2023-12-15 18:05:25 2024-01-15 18:06:13	否	<a href="#">管理</a>
live-j8000fd_xxOhoa	20G	20	mna-test2	2023-12-21 16:20:04 2024-01-21 16:19:00	否	<a href="#">管理</a>
live-ja0000fi-Crn_72	20G	20	mna-test10	2024-02-23 16:54:01 2024-03-23 16:53:39	否	<a href="#">管理</a>
live-jab000fjn7Cinil	20G	20	mna-ptml60g391	2024-03-05 11:13:15 2024-04-05 11:13:15	是	<a href="#">管理</a>
live-jak000fk5okAUjq	20G	19.419	mna-yt5ksujqdm	2024-03-14 14:34:41 2024-04-14 14:34:41	否	<a href="#">管理</a>
live-jb3000fkg545rt	20G	20	mna-4npghxkg6j	2024-03-29 15:47:59 2024-04-29 15:47:59	否	<a href="#">管理</a>

共 27 条

10 条 / 页
 1 / 3 页

# 其他功能

最近更新时间：2026-06-04 11:08:30

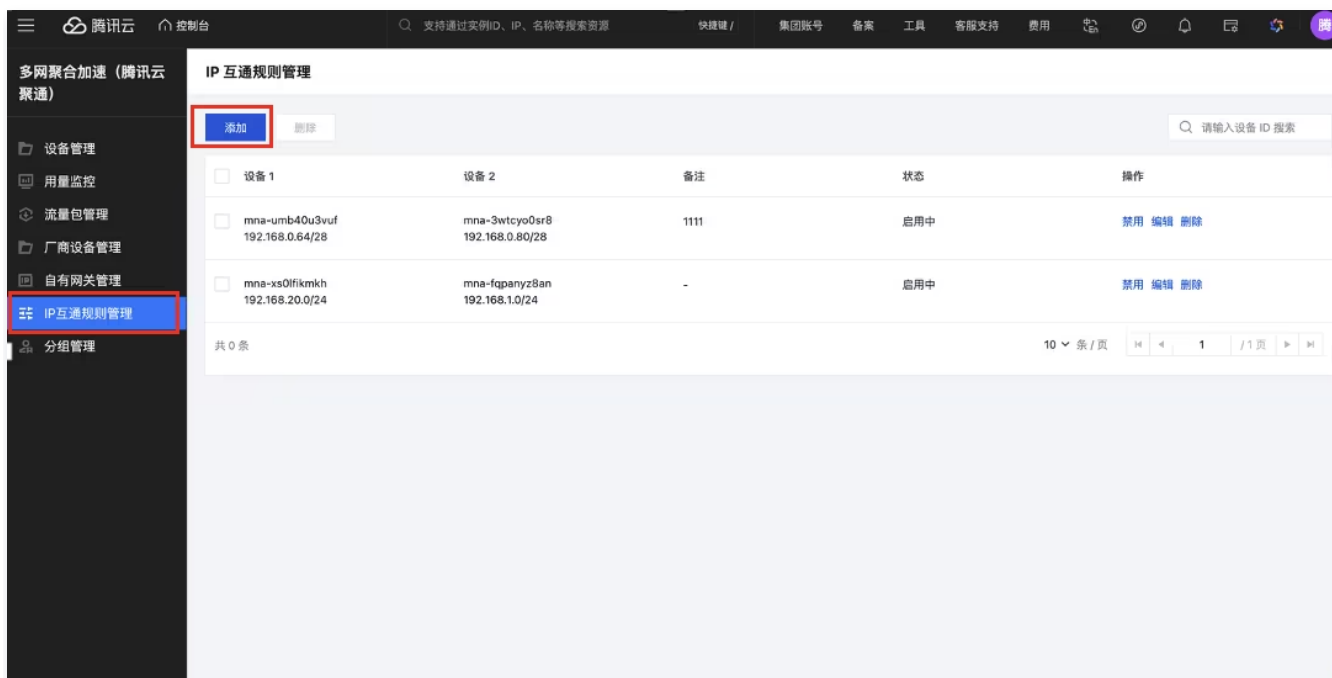
## 自有网关管理

客户自有网关被纳管后，即可在自有网关管理页面看到活跃的网关，并支持按照网关名称进行搜索。

## IP 互通规则管理

### IP 互通配置

使用 IP 互通能力需要先进行配置，单击控制台左侧栏 **IP 互通规则管理**，进入管理界面，单击**新建**进入配置界面。



选择要进行互通的设备，设置互通的网段即可。

#### 说明：

- IP 互通能力是指对两个设备的内网网段配置三层互访功能。配对后，对应的设备可以在对应的内网网段实现 IP 层双向互访。
- 需要注意一个设备上不能存在两条完全一样的对端网段。

### 添加 IP 互通规则 ✕

设备 1 \*

请选择 ▾

互通网段 (IPv4 CIDR)

10 ▾

.

0

.

0

.

0

/

12 ▾

支持 10.0.0.0 - 10.255.255.255 之间的网段

设备 2 \*

请选择 ▾

互通网段 (IPv4 CIDR) \*

10 ▾

.

0

.

0

.

0

/

12 ▾

支持 10.0.0.0 - 10.255.255.255 之间的网段

备注

i 当前仅支持两台设备之间 IP 互通

保存

取消

## 分组管理

在多网聚合加速（腾讯云聚通）分组管理页面，您可以对设备进行分组管理，查看以组为单位的加速统计数据。

分组名	设备数	描述	创建/更新时间	操作
group-0411	2	modifydescription <span style="font-size: 0.8em;">✎</span>	2024-04-11 10:54:59 2025-05-22 10:27:44	<a href="#">配置分组</a> <a href="#">删除</a>
group_robertmu	4	robertmu的设备分组 <span style="font-size: 0.8em;">✎</span>	2024-04-22 10:17:26 2025-05-22 10:27:44	<a href="#">配置分组</a> <a href="#">删除</a>
grouptest	4	0411test <span style="font-size: 0.8em;">✎</span>	2024-04-11 17:55:27 2025-10-29 15:07:51	<a href="#">配置分组</a> <a href="#">删除</a>
yonghui	1	- <span style="font-size: 0.8em;">✎</span>	2024-11-26 14:22:02 2024-11-26 14:22:24	<a href="#">配置分组</a> <a href="#">删除</a>

共 0 条 10 条 / 页 1 / 1 页

在分组列表中，单击配置分组，可查看当前分组下的所有设备，进一步查看设备详细信息。

腾讯云 控制台

支持通过实例ID、IP、名称等搜索资源

快捷操作 / 集团账号 备案 工具 客服支持 费用

多网聚合加速（腾讯云聚通）

← group-0411

**新增设备**

**分组信息**

分组名称 group-0411  
分组描述 modifydescription

**设备列表**

设备名称	设备 ID	备注	创建/最后一次在线时间	操作
xiaomi-0328	mna-kub2m2w2zn	小米测试	2024-03-28 13:46:21 2024-05-13 11:43:29	<a href="#">查看详情</a> <a href="#">移除</a>
0411-daliu	mna-ggm9fr9bvg	-	2024-04-11 17:56:15 2024-04-11 23:24:39	<a href="#">查看详情</a> <a href="#">移除</a>

共 2 条

10 条 / 页

1 / 1 页

# 物联网卡白名单配置操作指南

最近更新时间：2026-07-08 16:45:00

## 一、文档说明

对于行业端设备接入的场景（例如车联网、直播机、MFi 设备等），往往采用定向物联网卡进行网络连接。定向物联网卡需要提前向运营商注册需要访问的域名/IP 地址（即“白名单”），未在白名单中的地址将无法访问。本指南帮助您完成云聚通多网聚合加速（腾讯云聚通）服务所需的白名单配置，确保设备能正常连接加速网关。

## 方案总览

根据您的物联网卡白名单条目数量，本指南提供两种配置方案：

方案	适用场景	推荐程度
方案一：全量配置	白名单条目不受限	★★★ 首选
方案二：受限配置	白名单条目有限	★★☆ 比较推荐

## 二、配置前准备

在开始配置白名单之前，请先确认以下信息：

### 1. 确认设备运营区域

确认您的设备主要在哪些城市或区域运营（例如华东、华南等特定城市）。这将影响后续地域或网关节点的选择。

### 2. 确认物联网卡白名单条目上限

向您的物联网卡运营商了解白名单最多能添加多少条，无限制情况建议选择方案一。

## 三、白名单必加内容：控制器域名

无论您选择哪种方案，以下 4 个控制器域名都必须全部加入白名单。设备启动后需要先连接控制器完成注册和调度，缺少任何一个都可能导致无法正常接入。

域名	用途
register.multipath.tencent-cloud.com	注册中心主域名
register-backup.multipath.tencent-cloud.com	注册中心备域名
reg-rest-pre.multipath.tencent-cloud.com	预发布环境 REST 域名
reg-rest.multipath.tencent-cloud.com	现网 REST 域名

ⓘ 说明：

如果您的运营商支持泛域名，可直接用 `*.multipath.tencent-cloud.com`（示例）一条覆盖以上所有域名。明确的泛域名格式需与运营商做具体确认，不同的运营商规则可能不同。

## 四、方案一：全量配置（推荐）

### 适用场景

白名单条目数量不受限制，或条目足够多。这是最简单、效果最好的方案。所有地域白名单全部加完后，无需在控制台做任何设置，SDK 会自动就近选择最优的网关节点和运营商线路，无需手动指定。

### 步骤1 添加控制器域名

将第三节中的 4 个域名（或泛域名）全部加入白名单。

### 步骤2 添加全部接入点 IP 地址

将所有地域的 IP 地址加入白名单。

## 五、方案二：受限配置（条目有限时推荐）

### 适用场景

白名单条目有限，无法把所有地域的 IP 都加上。

### 工作原理

- 设备启动加速时，会先连接腾讯云的控制器。控制器会把可用的网关 IP 地址下发给设备，设备再去连接这些网关。默认情况下（即控制台可接入网关无配置时），控制器会下发所有地域的网关 IP。如果您的白名单没有把所有 IP 都加上，设备就可能收到一个没有加过白的 IP，导致连不通。
- 为了解决这个问题，您可以在腾讯云控制台配置“可接入网关”，这可以告诉控制器“我只加了哪些地域的白名单”。这样控制器就只会下发这些地域的网关 IP 给设备，不会下发其他地域的。设备收到的 IP 都是加过白的，就不会出现连不通的情况。

### 步骤1 添加控制器域名

将 [三、白名单必加内容：控制器域名](#) 中的 4 个域名（或泛域名）全部加入白名单。不可省略。

### 步骤2 获取限定地域的 IP 列表并加入白名单

根据您的设备运营区域，确定需要接入的地域（通常选择距离运营区域最近的一个或多个地域节点）。通过 [联系我们](#) 获取，并将获取的所有 IP 地址加入物联网卡白名单。

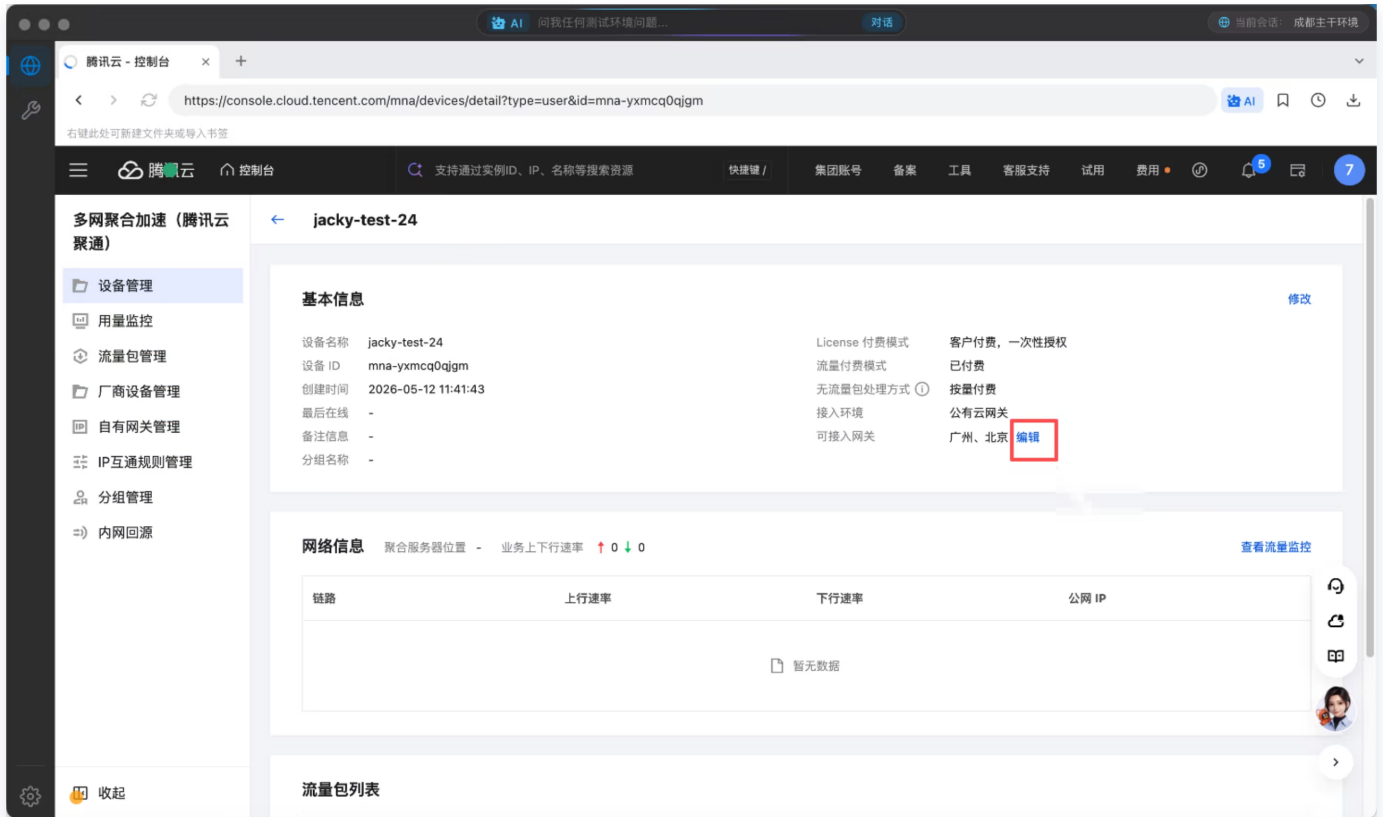
#### ⚠ 注意：

请确保将限定地域内每个节点的所有运营商线路 IP（BGP、移动、电信、联通）都加入白名单，不能只加其中某个运营商的。

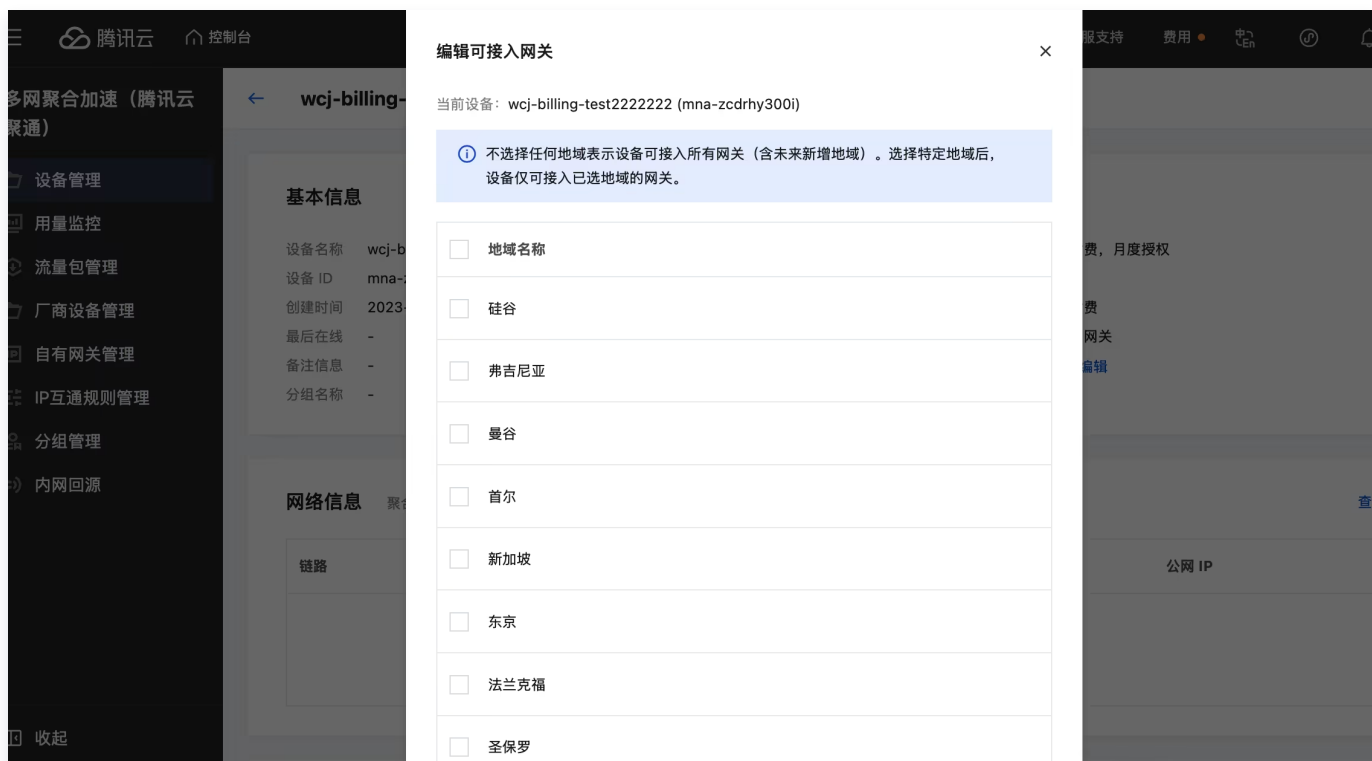
### 步骤3 配置接入地域范围

在腾讯云控制台选择“可接入网关”的地域。请确认所选地域的 IP 已在 Step 2 中加入白名单，否则设备收到未加白的 IP 会导致无法连通。

- 在 [腾讯云控制台](#) > 多网聚合加速（腾讯云聚通） > 设备管理，找到可接入网关，单击编辑。



- 在弹出的窗口中选择您要接入的地域，如下图所示：



### 说明：

配置完成后，SDK 自动调度时只会选择这些允许的地域，不会连接其他地域的网关。此外，全选仅代表当前所有 IP，后续新增地域不会自动接入。

## 示例：车联网场景，车辆在杭州运营

背景：车辆使用移动+电信双卡，主要在杭州及周边运营。

### 具体操作

- 4 个控制器域名全部加入白名单。
  - 工程师提供/自行查询杭州节点对应的 5 个 IP，将 5 个 IP（含 BGP\*1/联通\*2/电信\*1/移动\*1）全部加入白名单。
  - 腾讯云控制台配置接入地域限制为“杭州”。
- 即白名单总条数为 4 个域名 + 5 个 IP = 共 9 条。

### 注意：

如果后续设备运营区域扩展到其他城市，请在腾讯云控制台调整地域范围，并将新增地域的 IP 补充到白名单中。

## 六、常见问题

### 1. 业务服务器的 IP 需要加白名单吗？

不需要。白名单只管“设备到腾讯云加速网关”这一段。数据到了网关之后，由网关自己的网络去访问您的业务服务器，已经不走物联网卡了。

#### 🔔 说明：

如果您的设备上有些流量不经过加速网关、直接通过物联网卡访问其他服务器，那些地址需要您自行添加到白名单中。本指南只涉及加速服务相关的白名单。

## 2. 域名可以少加几个吗？

不可以。4 个控制器域名都必须添加。设备启动后需要先连接控制器完成注册和调度，任何一个域名缺失都可能导致无法正常接入。如果支持泛域名，用 `*.multipath.tencent-cloud.com` 一条搞定。

## 3. 只加限定地域/指定节点内某个运营商的线路可以吗？

不建议。请把对应地域或节点下所有运营商线路的 IP 都加上。SDK 会根据物联网卡的运营商自动选择对应线路，如果缺少某条线路的 IP，可能导致该运营商的卡无法正常加速。

## 4. 设备运营区域需要扩展怎么办？

- 方案一：无需操作。您已将全部 IP 加入白名单，新区域的网关 IP 已包含在内，SDK 会自动调度到新区域。
- 方案二：
  - 1.1 将新地域的 IP 加入白名单。
  - 1.2 在控制台勾选该地域。

## 5. 腾讯云地域节点增多了怎么办？

- 方案一：将新地域的 IP 补充到白名单即可，控制台无需配置，SDK 调度范围自动包含新地域。
- 方案二：如需使用新地域。
  - 1.1 将新地域的 IP 加入白名单。
  - 1.2 在控制台勾选该地域。

## 6. IP 地址会变化吗？

IP 地址相对稳定，但如果有节点扩容或调整，腾讯云会提前通知更新。建议定期与您的对接工程师确认是否有 IP 变更。

# 动态加速配置参考

最近更新时间：2026-06-05 17:30:30

## 一、概述

动态加速是指 SDK 持续监测网络质量，只在网络变差时自动建议开启加速，网络恢复后建议关闭。这样可以在保障体验的同时节省加速流量成本。动态加速的核心问题只有两个：什么时候开？什么时候关？本文档目标旨在帮助理解动态加速的策略配置逻辑，以及不同场景下加速参数的选择和调整。

## 二、核心概念

### 1. 三个关键维度

配置动态加速前，需要理解三个相互独立但有推荐搭配关系的维度：

- 动态加速的场景分别有游戏、实时音视频、直播推流；
- 模式可选择多发选收模式（双发）、实时模式（RTC）、聚合模式；
- 动态加速策略包括探测启动策略、快启动策略、动态回源策略和建议关闭策略。

### 2. 场景与模式的推荐搭配

以下是经过验证的推荐搭配（非强制），一般按推荐模式搭配效果最佳：

场景	推荐模式	原因	典型客户
游戏（FPS/MOBA）	多发选收（双发）	小包业务，追求极低延迟	对战类手游
实时音视频	实时模式（RTC）	音视频流对连续性要求高	<ul style="list-style-type: none"><li>• 语音通话</li><li>• 视频会议</li></ul>
直播推流	聚合模式	大带宽需求，聚合多通道	<ul style="list-style-type: none"><li>• 游戏</li><li>• 户外直播</li></ul>

## 三、策略详解

四个策略覆盖了动态加速从开启到关闭的完整判断链路：

策略	解决什么问题	触发后行为
探测启动策略	网络持续变差，需要开加速	回调客户端后由客户决定
快启动策略	网络突然恶化，紧急需要加速	回调客户端后由客户决定
动态回源策略	加速链路比直连还差	SDK 自动回退到直连

建议关闭策略	网络恢复，加速不再必要	回调客户端后由客户决定
--------	-------------	-------------

**⚠ 注意：**

策略配置独立于模式。不同场景和模式的区别主要体现在策略参数的推荐值不同，策略逻辑本身是通用的。

## 1. 探测启动策略

SDK 持续监测网络质量，当指标在一段时间内持续超出阈值时，回调给客户端建议开启加速。属于渐进式检测，适合发现网络正在恶化的趋势。判断逻辑：在【时间窗口】内，平均时延超过阈值、或抖动超过阈值、或丢包率超过阈值（三者满足任一），即触发回调。通俗地说，过去 X 秒内，延迟太高/抖动太大/丢包太多，就建议开加速。下表为参数推荐值。

参数	含义	游戏场景	实时音视频	推流场景
时间窗口	观察时间段	8s	8s	建议全时开启（无需配置）
时延阈值	平均延迟超此值触发	60ms	80ms	-
抖动阈值	延迟波动超此值触发	15ms	20ms	-
丢包阈值	丢包率超此值触发	5%	5%	-

**ⓘ 说明：**

游戏场景阈值更低是因为游戏对延迟极度敏感（60ms 即可感知卡顿），需要更早触发。推流场景因为带宽需求大且持续，建议全时开启，无需探测触发。

## 2. 快启动策略

与探测启动互补，探测启动是8秒窗口的趋势判断，快启动是2秒窗口的紧急判断。当网络突然急剧恶化时（如进电梯、Wi-Fi 断开），快启动能更快触发。两个策略同时生效，任一触发即回调。判断逻辑：在【时间窗口】内，时延超过【时延阈值】即触发。注意快启动只看时延一个指标（不看抖动和丢包），窗口更短、阈值更高，专门用于捕捉突然变差。下表为参数推荐值。

参数	含义	游戏场景	实时音视频	推流场景
时间窗口	检测窗口时长	2s	2s	建议全时开启
时延阈值	延迟超此值触发	80ms	100ms	-

探测启动和快启动的对比如下：

对比	探测启动	快启动
----	------	-----

窗口 / 指标	8s / 时延+抖动+丢包	2s / 仅时延
触发阈值	较低（灵敏）	较高（防误触）
定位	发现持续变差的趋势	捕捉突然恶化的突变

### 3. 动态回源策略

保护机制——加速开启后，如果加速链路本身出现严重问题（比直连还差），SDK 自动将流量回退到直连主链路，避免加速反而更卡。与建议关闭策略的区别：动态回源是 SDK 自动执行的保护动作（不等客户决定），建议关闭则是通知客户由客户自行决定。触发条件如下（满足任一即自动回源）。

条件	判断规则	通俗含义
条件一完全失联	加速链路连续丢包达到【连续丢包次数】，且主链路未同时失联。	加速链路断了，但直连还通，应立即回退。
条件二延迟极高	加速链路连续【连续高延迟包数】个包的延迟 $\geq$ 【单包延迟上限】ms。	没断但延迟已高到不可接受。
条件三持续比直连差	【评估窗口】内，加速平均延迟 $>$ 主链路 $\times$ 【延迟比例阈值】%，且加速平均延迟 $>$ 【最低延迟阈值】ms。	长期观察走加速还不如直连，且延迟确实不低。

**说明：**

条件三为什么需要最低延迟阈值？——避免误判。例如主链路10ms、加速12ms 虽然超过110%，但12ms 本身完全可接受，不应回源。加上最低阈值（66ms）可过滤此类情况。

各场景参数推荐值如下。

条件	参数	游戏（多发选收）	实时音视频（RTC）
条件一	连续丢包次数	4个包	4个包
条件二	连续高延迟包数	4个包	不启用
	单包延迟上限	400ms	-
条件三	评估窗口时长	60秒	60秒
	延迟比例阈值	110%	110%
	最低延迟阈值	66ms	66ms
推流/聚合模式	仅启用条件一（连续丢包次数=4），不启用条件二、三。		

**说明：**

实时场景不启用条件二，因 RTC 协议自身有重传机制。聚合模式不启用时延类条件（二、三），因聚合天然会有较高延迟。

## 4. 建议关闭策略

当加速不再必要或物理条件不支持时，SDK 回调建议客户端关闭加速。客户端可选择接受或忽略。触发条件如下（满足任一即回调建议）。

条件	判断规则	通俗含义
条件一仅剩单网卡	设备仅剩一个网卡可用，等待【等待时长】秒后建议关闭。	多发需要至少两个网卡，只剩一个时加速无意义。
条件二完全无网	设备无可用的网卡，则立即建议关闭。	没网了加速自然要关。
条件三效果长期不佳	【质量评估窗口】内加速时延 > 主链路 × 【时延比例阈值】%，且抖动 > 主链路 × 【抖动比例阈值】%。	长期加速效果不如直连，关了省成本。
条件四副链路全挂	【测量窗口】秒内副链路全部丢包。	所有副网卡访问超时，加速无意义。

各场景参数推荐值如下。（推流/聚合模式：仅启用条件一（等待10秒）和条件二（立即触发），不启用条件三、四。聚合模式不建议用时延条件做关闭判断。）

条件	参数	游戏（多发选收）	实时音视频（RTC）
条件一	等待时长	10秒	10秒
条件二	（立即触发）	-	-
条件三	质量评估窗口	300秒	300秒
	时延比例 / 抖动比例	100% / 80%	100% / 80%
条件四	测量窗口	不启用	10秒

## 四、设置接口

### 1. 探测启动策略

测速配置对象，通过设置此参数，实现动态加速效果。

#### 1.1 Android SDK

API	描述
setRTT	设置时延阈值
setLoss	设置丢包率阈值
setJitter	设置抖动阈值
setTime	设置滑动窗口测量时间

### setRTT

设置时延阈值，当滑动窗口内终端实际网络时延大于设置的时延阈值时，会回调 onStartMpAcc 接口。

```
public MeasureConfig setRTT(int rtt)
```

参数说明:

参数	类型	描述
RTT	int	延迟阈值（毫秒）

### setLoss

设置丢包率阈值，当滑动窗口内终端实际网络丢包大于设置的丢包阈值时，会回调 onStartMpAcc 接口。

```
public MeasureConfig setLoss(int loss)
```

参数说明:

参数	类型	描述
loss	int	丢包率百分比阈值（0-100）

### setJitter

设置抖动阈值，当滑动窗口内终端实际网络抖动大于设置的抖动阈值时，会回调 onStartMpAcc 接口。

```
public MeasureConfig setJitter(int jitter)
```

参数说明:

参数	类型	描述
----	----	----

jitter	int	网络抖动阈值（毫秒）
--------	-----	------------

## setTime

设置滑动窗口测量时间。

```
public MeasureConfig setTime(int time)
```

参数说明：

参数	类型	描述
time	int	测量时长，单位：ms

## 1.2 iOS SDK

属性	描述
time	滑动窗口测量时间（ms），默认30000
rtt	时延阈值，单位：ms
loss	丢包率百分比阈值（0-100）
jitter	抖动阈值，单位：ms

## 2. 快启动策略

通过设置快启动窗口时间和快启动时延阈值实现快启动。

### 2.1 Android SDK

API	描述
setQuickTime	设置快启动窗口时间
setQuickRtt	设置快启动时延阈值

#### setQuickRtt

设置快启动时延阈值，此接口主要针对时延陡增场景，当快启动窗口内终端实际网络时延大于设置的快启动时延阈值时，会回调 onStartMpAcc 接口。

```
public MeasureConfig setQuickRtt(int quickRtt)
```

参数说明：

参数	类型	描述
quickRtt	int	快启动时延阈值，单位：ms

### setQuickTime

设置快启动窗口测量时间。

```
public MeasureConfig setQuickTime(int quickTime)
```

参数说明：

参数	类型	描述
quickTime	int	快启动窗口时间，单位：ms

## 2.2 iOS SDK

属性	描述
quickTime	快启动窗口时间，单位：ms
quickRtt	快启动时延阈值，单位：ms

## 3. 动态回源策略

逃生回源自定义对象，如果业务未设置 AdditionalOptions，SDK 会采用默认逃生逻辑（触发 onAccException），如果业务已设置，则 SDK 逃生逻辑以业务设置为主。

### 3.1 Android SDK

API	描述
PacketLossEscape	加速链路连续【 packetLossCount 】失联，触发逃生。
MaxRttEscape	加速链路连续【 maxRttCount 】 ping 值大于等于【 rttThreshold 】 ms，触发逃生。
RttExceptionEscape	加速链路【 detectWindowTime 】 ms 平均 ping 值大于主链路【 rttDiffThreshold 】%，并且 acc 平均时延大于【 accBenchmarkRtt 】 ms，触发逃生。

StopAccRemind	加速链路【detectWindowTime】内平均时延大于主链路【rttDiffThreshold】%，且抖动大于主链路【jitterDiffThreshold】%，建议关闭加速（触发onStopMpAcc）。
setAdditionalOptions	设置逃生回源自定义选项。

### setAdditionalOptions

设置逃生回源自定义选项。

```
public MeasureConfig setAdditionalOptions(AdditionalOptions additionalOptions)
```

## 3.2 iOS SDK

属性	描述
packetLossEscape	加速链路连续【packetLossCount】失联。
maxRttEscape	加速链路连续【maxRttCount】ping 值大于等于【rttThreshold】ms。
rttExceptionEscape	加速链路【detectWindowTime】平均 ping 值大于主链路【rttDiffThreshold】%，并且 acc 平均时延大于【accBenchmarkRtt】ms。
stopAccRemind	加速链路【detectWindowTime】内平均时延大于主链路【rttDiffRate】%，且抖动大于主链路【jitterRate】%，建议关闭加速。
additionalOptions	设置逃生附件选项。

## 4. 建议关闭策略

该策略一般不需要修改配置，采用默认值。如需修改在 StopAccRemind 进行修改。

### 4.1 Android SDK

API	描述
StopAccRemind	加速链路【detectWindowTime】内平均时延大于主链路【rttDiffThreshold】%，且抖动大于主链路【jitterDiffThreshold】%，建议关闭加速（触发onStopMpAcc）。

setDisableSlaveLossDetect	设置禁用从路丢包检测。
---------------------------	-------------

### setDisableSlaveLossDetect

设置禁用从路检测，针对快切模式，默认不禁用。如果在测量窗口内，从路时延出现100%丢包现象，将会回调onStopMpAcc提醒关闭加速。

```
public MeasureConfig setDisableSlaveLossDetect (boolean
disableSlaveLossDetect)
```

参数说明：

参数	类型	描述
disableSlaveLossDetect	boolean	true: 禁用 false: 不禁用

## 4.2 iOS SDK

属性	描述
stopAccRemind	加速链路【detectWindowTime】内平均时延大于主链路【rttDiffRate】%，且抖动大于主链路【jitterRate】%，建议关闭加速。
disableSlaveLossDetect	设置禁用从路丢包检测。 true: 禁用 false: 不禁用

## 五、示例代码

详情请参见 [动态加速](#)。