

# 排除AAA限制功能故障

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[工作机制](#)

[AAAMGR队列](#)

[限制](#)

[相关的思科支持社区讨论](#)

## 简介

本文档介绍AAA(RADIUS)记录的限制功能，该功能支持限制发送到RADIUS服务器的访问（身份验证和授权）和记帐记录。

此功能允许用户配置适当的限制速率，以在带宽不足时避免网络拥塞和不稳定，以适应从Cisco路由器到RADIUS服务器的突然突发记录。

## 先决条件

### 要求

本文档没有任何特定的要求。

### 使用的组件

本文档中的信息基于ASR5k平台。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

## 背景信息

当aamgr以高速率向RADIUS服务器发送RADIUS消息时（例如，当大量会话同时关闭时，会同时生成所有会话的记帐停止消息），RADIUS服务器可能无法以如此高的速率接收消息。要处理此情况，我们需要aamgr的有效速率控制机制，以便aamgr以最佳速率发送消息，使RADIUS服务器能够接收所有消息，并确保没有因RADIUS服务器的过载而丢弃消息。

## 工作机制

当aaamgr以配置的速率向RADIUS服务器发送消息时，它会在整个服务器中均匀地发送消息而不是在单个突发中发送所有消息。根据配置，每秒被划分为多个相等的时间段（每个时间段有特定的时间段）。插槽的最短时间段可能是50毫秒。

必须配置速率，并考虑这些因素

- 来电的速率，
- aaamgr实例数
- RADIUS服务器接收消息和
- 内部间隔（用于记帐配置）
- 用于服务器选择的算法

如果身份验证服务器值的配置值太低，则会出现瓶颈，堵塞，这可能导致由于会话设置超时而导致呼叫被丢弃。如果为记帐服务器配置了低值，则会观察到由于队列溢出而大量清除记帐消息。

当配置该功能时，计算第二时间段和第二时间段的时隙数并存储在半径级别。当消息准备好发送到RADIUS服务器时，系统会检查是否达到配额（此时隙的消息数）。如果达不到限制，则发送消息，如果达到，则消息在服务器级别队列中排队，以在将来的时隙中发送。每个RADIUS服务器都包含有关当前时隙中发送的消息数和时隙到期时间的详细信息。当从服务器级别队列中选择排队的消息时，它们会被放在实例级别队列的头部，确保优先于任何其他新消息。从实例级别队列中选择消息以进行维护。

## AAAMGR队列

AAAMGR中有两种类型的消息队列：

1. 实例级别队列
2. 服务器级队列

生成消息时，消息最初会在实例级别队列中排队以用于服务。

实例级队列每50毫秒处理25毫秒。从实例级别队列出列的任何消息都将尝试发送到RADIUS服务器。在某些情况下，我们可能无法发送消息（没有可用带宽或无可用ID）。在这种情况下，尝试失败的消息将在服务器级别队列中排队。每隔50毫秒，您就会选择多个ID可用且带宽可用的消息，并将其置于实例级别队列的头部（这些消息比实例级别队列中存在的任何其他消息都旧）。

当对记帐消息有速率控制，并且实例级别队列中有许多记帐消息时，则任何新的身份验证消息将进入实例级别队列的尾部。要获得处理，它必须等待所有记帐消息（在新的身份验证消息之前）发送到RADIUS服务器或移动到服务器级别队列。它是现有行为，不会修改。因此，它会导致新的身份验证消息被处理的延迟较小。

### 示例

根据最大速率（值为5），您可以在1秒内发送5条消息，每个aamgr有256条未处理的RADIUS身份验证消息（默认最大未处理配置）未被应答。如果有5条以上的消息，在1秒内，这些消息将排入队列，直到AAA服务器响应现有请求。

如果您从一个aamgr向服务器发送了256条Radius身份验证消息，则剩余请求将被置于队列中，直到AAA服务器响应现有请求。它将再次进入与最大速率相同的队列。只有当您有空闲插槽时，才会从队列中提取消息。当您收到消息响应或消息超时时，空闲插槽将进入。

# 限制

由于Cisco ASR5K是一个分布式系统，具有处理呼叫的独立会话管理器/aaamgr对，因此速率限制只能针对独立aaamgr实例实施。理论上，只需将实例总数与最大速率相乘，即可将单个实例的速率扩展到整个Cisco ASR5K机箱每个实例。

这个数字只是晴天情景中的绝对上限。您不能将Cisco ASR5K视为黑匣子，也不能假设如果系统中看到的计算值未超过上限，所有呼叫都应成功。

Radius max-rate与与系统相关的其他内部和外部参数关联。如果其中一个条件未满足，请查看预期影响。

## 条件

从demuxmgr到所有会话的呼叫的均匀分配

IMSI的均匀分布（以防万一）  
调解记帐）

不会突然出现呼叫

Radius服务器应及时响应

## 如果未满足影响

如果呼叫分布不统一，则RADIUS消息可能排队等待某些实例。因此，即使未达到理论的最大速率限制，消息排队的实例的呼叫也将被丢弃。

中介 — 记帐轮询基于基于IMSI的路由。

在这种情况下，根据IMSI分布，根据路由逻辑，某些服务器可能优先于其他服务器组，可能为那些导致呼叫丢弃的服务器组排队。

如果出现新呼叫的突发，则新生成的RADIUS消息将再次排队。到处理新RADIUS请求时。会话设置时间可能已过期，呼叫丢弃。

当RADIUS请求因服务器问题而超时时，将再次出现队列累积，因为除非从系统中删除当前期望响应的请求，否则不会发出请求。从系统中删除超时消息的速率也取决于最大未处理消息配置。

在许多情况下，我们可以看到访问请求并非由所有活动的aamgr任务处理。这意味着，在sessmgr任务中，呼叫分配不均，而且，呼叫处理中并不涉及所有aamgr实例。

呼叫分配不基于严格的轮询机制，即如果有10个来电，它们将进入单调算法中的10个会话。

呼叫分配基于这四个主要因素

- active\_session\_count
- cpu\_load
- Round\_trip\_delay(demuxmgr - sessmgr - demuxmgr)
- outstanding\_add\_request (解复用器到sessmgr)

这是当前实施。最大速率只是上限，但由于我们架构的分布性，您无法直接将其外推到机箱负载。行为取决于给定AAAmgr上的负载在特定时间。

应使用Radius max-rate队列监控系统状态。如果存在队列累积，则这意味着这4个（请参阅表）条件之一未满足，您必须确定其根本原因。

\*\*可以实施并持续监控最大速率队列阈值。