

IMS 网络中最优负载均衡的服务触发算法研究

张 庆¹,冯 晨¹,余江涛¹,罗 威²,任 祥³

(1. 国网上海市电力公司,上海 200122;

2. 南京南瑞信息通信科技有限公司,江苏 南京 211000;

3. 南京邮电大学 通信与信息工程学院,江苏 南京 210003)

摘 要:随着电力系统中 IMS 系统承载业务种类的不断增多,网络负载的不断扩大,这就对 IMS 网络的负载均衡能力提出了很高的要求。文中提出了一种应用于 IMS 系统网络中的异构多应用服务器系统的负载均衡方案。在 IMS 系统网络中,服务呼叫会话控制功能利用由归属订户服务器下载的初始过滤标准并通过会话发起协议消息生成服务触发器。为了在不同结构的多个应用服务器之间进行负载均衡,实现资源优化,应设置服务触发器的权重值。在以前的系统中,权重值往往由操作员手动设置,而该方法自动计算负载均衡的最佳权重。此外,还提出了简化算法来减少特定情况下的计算量。仿真结果表明,所提方案在负载均衡中优于以前的方案,并以快速收敛的方式自动调整系统的变化。

关键词:IMS;异构系统;多应用服务器;负载均衡;权重值;服务触发算法

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2019)10-0191-05

doi:10.3969/j.issn.1673-629X.2019.10.037

Research on Service Triggering Algorithm for Optimal Load Balancing in IMS Network

ZHANG Qing¹,FENG Chen¹,YU Jiang-tao¹,LUO Wei²,REN Xiang³

(1. State Grid Shanghai Municipal Electric Power Company, Shanghai 200122, China;

2. Nanjing Nanrui Information and Communication Technology Co., Ltd., Nanjing 211000, China;

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

Abstract:With the continuous increase of IMS system load types and network load in the power system, the load balancing of IMS network is highly required. We propose a load balancing scheme for heterogeneous multi-application server system in IMS network. In the IMS network, serving call session control function uses the initial filter criterion downloaded by home subscriber server and generates service triggers through session initiation protocol. In order to balance the load among multiple AS with different structures and realize resource optimization, the weight value of service trigger should be set. In previous systems, the weights are often set manually by operators, and the proposed method is to automatically calculate the optimal weights of load balancing. In addition, a simplified algorithm is proposed to reduce the computational complexity in specific cases. The simulation shows that the proposed scheme is superior to the previous ones in load balancing and adjusts the change of the system automatically in a fast convergence way.

Key words:IMS;heterogeneous system;multi-application server;load balancing;weight value;service triggering algorithm

0 引言

IMS^[1-2](IP multimedia subsystem, IP 多媒体子系统)是支持 SoIP(service on Internet protocol,因特网协议上的服务)的常用系统,是多媒体服务和 VoIP(voice over Internet protocol,因特网协议语音)的融合技术,被 3GPP 定义为多媒体通信的核心网域^[3-5]。在

IMS 网络中,AS(application server,应用服务器)通过处理服务触发分組和执行服务逻辑在执行各种多媒体服务中起着至关重要的作用。随着服务和用户数量的增加,现有 AS 负载量越来越大,这是导致系统错误的主要原因。因此,许多通信服务公司在现有系统中增加了新的 AS,更多、更稳定地为更多用户提供服务^[6]。

收稿日期:2018-12-04

修回日期:2019-04-08

网络出版时间:2019-06-26

基金项目:国家自然科学基金面上项目(61771252,61471203)

作者简介:张 庆(1966-),男,硕士,高级工程师,研究方向为电力系统通信技术运维。

网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20190626.0833.050.html

在这种情况下,AS 供应商可以与其他供应商不同,并且可以向每个 AS 注册不同的服务。在使用这些异构 AS 时,AS 之间的负载平衡始终是一个问题,因为集中流量到特定 AS 是产生系统错误的重要原因。要解决这个问题,必须通过有效的服务触发算法进行负载均衡。

以往大多数关于服务触发算法的研究都是基于单一或同类 AS 环境,主要集中在提高系统性能或减少会话建立延迟上。然而,现有关于异构 AS 之间的负载均衡^[7]的研究还不够。文献[8-13]中提及的负载均衡方案可以分类如下:循环;给每个 AS 加权(加权触发);触发每个服务到专用 AS(专用 AS 触发)。

在商业化系统中,权重值由操作员手动设定。使用手动设置的权重值,很难执行精确的负载平衡,也很难适应系统的变化,如系统错误,新 AS 的安装或新服务的注册等。因此,文中提出以自动方式计算每个 AS 服务的最佳权重值,并执行最佳负载平衡的方案。

1 异构多应用服务器系统模型

1.1 主要假设和系统结构

考虑系统拥有 N 个 AS 和 M 个基于 iFC (initial filter criterion, 初始过滤标准) 的服务。注册到 AS 的每个服务都是基于 iFC 的,并且 S-CSCF (serving call session control function, 服务会话控制功能) 在 S-CSCF 生成服务触发分组之前从 HSS (home subscriber server, 归属订户服务器) 下载 iFC 信息。假设 S-CSCF 知道每个 AS 上的每个注册服务以及每个服务是否通过 iFC 信息注册到 AS。

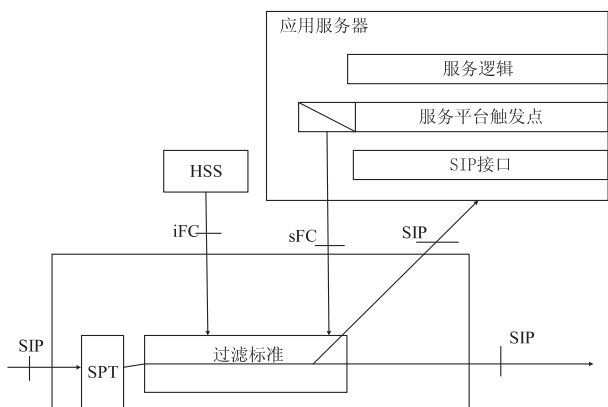


图1 服务触发的 IMS 构架

图1为系统的基本考虑模型^[14-15]。S-CSCF 在用户认证期间从 HSS 获得 iFC 信息,并基于该信息确定 AS 的候选集合。在 AS 候选集确定之后,根据在 DNS (domain name server, 域名服务器) 中设置的权重值确定用于触发到每个 AS 服务的传输概率,并且 S-CSCF 应该在传输之前通过查询 DNS 来知道它。S-CSCF 基于该传输概率经由 SIP (session initiation protocol, 会话发起协议) 消息向 AS 发送服务触发分组。

在大多数基于 IMS 的商业化设备中,都会安装作为统计服务器的 EMS (element management server, 元素管理服务器) 以便于系统管理。EMS 包含监视系统的信息,例如每个服务的呼叫数,用户数或错误历史。通过利用来自 EMS 的信息,可以获得服务触发分组的概率分布,即用户使用每个服务的频率。如果知道服务触发器的正确概率分布,则可以执行负载平衡。文中假设 EMS 使用 SOAP^[8] (simple object access protocol, 简单对象访问协议) 定期更新 DNS 中服务数据包的概率分布,并且 S-CSCF 通过查询 DNS 来学习概率的准确分布。

1.2 元素矩阵的定义

本节定义了3个矩阵来表达系统模型。首先是权重矩阵。权重矩阵(WM)为 $N \times M$ 。每行代表特定的 AS 编号(从1到 N),每列表示在 iFC 中定义的服务优先级编号。换言之,系统中总共有 N 个 AS 和 M 个服务可供获得。WM 包含 $W_{i,j}$ 个元素, $W_{i,j}$ 是服务 j 触发到 AS_i 的条件概率。因此 $\sum_{i=1}^N W_{i,j} = 1$ 。换句话说,每行或列的元素的总和是1。如果服务 j 未在 AS_i 中注册, $W_{i,j}$ 的值为0。

在大多数情况下,系统操作员通过存储在 HSS 中的 iFC 知道哪些服务是在特定 AS 上注册的信息。因此,如果服务 j 未在 AS_i 中注册, $W_{i,j}$ 可以在计算最佳权重之前将其设置为0,且除非该服务新注册到 AS,否则该值无需更改。换句话说,在计算最佳权重之前,S-CSCF 可以在 WM 的整个元素之间区分非零元素和零元素。从下面开始,会用 W 代表 WM。

第二个是概率矩阵(PM) ($M \times 1$),表示每个服务的概率分布。它包含元素 P_j ,该元素表示正在进行的服务触发数据包将执行服务的概率为 j 。指标 j 表示 iFC 中定义的服务优先级。 j 的值越低,服务的优先级越高。显然, $\sum_{j=1}^M P_j = 1$,这是因为正在进行的服务触发器应该是可用服务触发集的元素。PM 的元素都是由 EMS 的统计信息给出的,可以用精确的 PM 计算最优的 WM 负载均衡。文中用 P 表示 PM。

最后一个是资源矩阵(RM),表示负载与每个 AS 的比率。RM 在计算 WM 时扮演着资源约束的角色。通过调整 RM 中元素的值,运算符可以控制每个 AS 的负载比。这个矩阵是 $N \times 1$ 。文中用 R 表示该矩阵。

利用上面描述的3个矩阵,可以用如下等式表示此系统。

$$WP = R \quad (1)$$

在该系统中,矩阵 P 的元素值通过 EMS 得到,矩阵 R 由操作员手动设置。因此,如果通过预先确定矩

阵 \mathbf{P} 与矩阵 \mathbf{R} 的值以及 S-CSCF 由矩阵 \mathbf{W} 中的元素计算出向特定 AS 触发服务的条件概率来计算矩阵 \mathbf{W} , 就可以执行最佳的负载平衡。计算 \mathbf{W} 中元素值的方法将在下一节中介绍。

2 多 AS 系统中的负载均衡方案

2.1 基本算法

为了执行文中算法, 首先将矩阵 \mathbf{W} 进行分解:

$$\mathbf{W} = \mathbf{X}\mathbf{Q} \quad (2)$$

其中, \mathbf{Q} 矩阵是 $M \times M$ 对角矩阵, 其元素可表示如下:

$$q_{i,j} = \begin{cases} \frac{1}{p_j}, i=j \\ 0, i \neq j \end{cases} \quad (3)$$

如果矩阵 \mathbf{Q} 如上述定义, $\mathbf{Q}\mathbf{P}$ 变为元素全是 1 的 $M \times 1$ 矩阵, 那么原先系统转换为以下表达式:

$$\mathbf{W}\mathbf{P} = \mathbf{X}\mathbf{Q}\mathbf{P} = \mathbf{X} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} \quad (4)$$

通过将矩阵 \mathbf{W} 分解为矩阵 \mathbf{X} 乘以矩阵 \mathbf{Q} , 每个与矩阵 \mathbf{X} 相乘的元素都变为 1。因此可以执行更简单的计算。此处将矩阵 \mathbf{X} 定义为辅助矩阵。矩阵 \mathbf{X} 的特点是每行中每个元素的总和与矩阵 \mathbf{R} 的行相同, 并且每列中每个元素的总和成为服务的特定触发概率 (p_j)。可以用以下方程表示:

$$\sum_{j=1}^M X_{i,j} = r_i \quad (5)$$

$$\sum_{i=1}^N X_{i,j} = p_j \quad (6)$$

如果获得的辅助矩阵 \mathbf{X} 有了上述限制, 则可以通过来自 EMS 信息已知的矩阵 \mathbf{Q} 得到矩阵 \mathbf{W} 。通过基于该矩阵 \mathbf{W} 执行服务触发, 可以以与矩阵 \mathbf{R} 相同的比率完成负载均衡。为获得矩阵 \mathbf{X} , 将应用以下算法:

(1) 如果服务 j 未向 AS_i 注册, 设置 $X_{i,j}$ 为 0。

(2) 在未分配列中选择具有最多零的一列。如果非零元素数量相同, 则选择包含更高优先级服务的列。

(3) 调查每一行并验证每行中未分配的非零元素 ($N_{i,j}$) 的数量。当 $X_{i,j} = 0$ 时, $N_{i,j}$ 被设置为无效。如果 $N_{i,j}$ 的每个元素都通过了验证, 则继续执行步骤 4。

(4) 分配 $X_{i,j} = \frac{p_j}{N_{i,j}} \times \frac{r_i}{\sum_{i=1}^N \frac{r_i}{N_{i,j}}}$, 并从资源矩阵的特

定行中减去分配的量。如果完成了包含该元素的整个列的分配, 则将该列视为已分配列, 并且在以下任何计算中都不会考虑该列。

(5) 如果存在任何元素满足 $X_{i,j} > r_i$, 设定该元素的 $N_{i,j} = \infty$ 。在那之后, 让 $K_j = \sum_{k=1}^N (X_{k,j} - r_k)^+$ 并且分配每个 $X_{i,j}$ 为:

$$X_{i,j} = \begin{cases} r_i, X_{i,j} > r_i \\ X_{i,j} + \frac{K_j}{N_{i,j}} \times \frac{r_i}{\sum_{i=1}^N \frac{r_i}{N_{i,j}}}, \text{otherwise} \end{cases} \quad (7)$$

(6) 如果存在更多要分配的列, 继续执行步骤 2, 否则结束算法。

通过上述算法可以获得矩阵 \mathbf{X} , 并通过已知的矩阵 \mathbf{X} 乘上矩阵 \mathbf{P} 来计算矩阵 \mathbf{W} 。如果按照矩阵 \mathbf{W} 执行服务触发, 最佳负载平衡是可能实现的。

2.2 存在同类列集时的简化算法

在上述算法中, 应该搜索每个元素并验证每个时间点 $N_{i,j}$ 和 r_i 的分配。随着服务种类或 AS 数量的增加, 进行矩阵 \mathbf{X} 的运算也会增加。因此, 为了支持具有大量 AS 的服务, 就需要一个简化算法来减少计算量。

通常, 服务的数量远远大于 AS 的数量, 并且由于成本问题, 许多公司希望以尽可能少的 AS 支持尽可能多的服务。例如, KT (Korea Telecom, 韩国电信) 用 4 个 AS 来支撑 27 个应用服务。因此, 多数会将一些服务注册到具有同类模式的相同 AS 中。换句话说, 矩阵 \mathbf{X} 中拥有相同分配结构的列也是可以存在的。此外, 如果 2^M (每个服务注册到 AS 的可能模式的数量) 小于服务的数量 (N), 这些列的集合显然符合 Pigeon-hole 原则^[8]。

使用相同的列结构, 可以很容易地注意到应该将相同的元素值分配给具有相同结构的列。因此可以得出结论, 一旦完成一列的分配, 可以容易地分配具有相同结构的列的值而无需额外计算。基于这个想法, 文中提出了一种改进的算法来减少计算量。

为了指定算法, 定义了同类列和同类列集。同类列是具有相同特定列结构的列。例如, 在表 1 中, 服务 #3 是服务 #1 的同类列, 这是因为它们具有相同的注册结构。同类列集是包含特定列的同类列的组。同类列集的元素是彼此的同类列。例如, 集合 {服务 #1, 服务 #3} 是服务 #1 和服务 #3 的同类列集。

在新算法中, 已分配列的值被复制并自动分配到其他同类列而无需进一步计算。修改后的算法如下:

(1) 如果服务 j 没有在 AS_i 中被注册, 则矩阵 \mathbf{X} 中每个 $X_{i,j}$ 的值为零。

(2) 选择未被分配的列中具有最多零的列。如果非零元素数量相同, 则选择包含更高优先级服务的列。

(3) 调查每一行并验证每行中未分配的非零元素 ($N_{i,j}$) 的数量, 当 $X_{i,j} = 0$ 时, $N_{i,j}$ 被设置为无限。如果

$N_{i,j}$ 中每个元素进行了验证,则继续执行步骤4。

(4) 设置非0元素 $X_{i,j} = \frac{P_j}{N_{i,j}} \times \frac{r_i}{\sum_{i=1}^N \frac{r_i}{N_{i,j}}}$, 从资源矩

阵的特定行中减去被分配的量。

(5) 如果存在任何满足的元素 $X_{i,j} > r_i$, 则设定该元素的 $N_{i,j} = \infty$ 。然后, 使 $K_j = \sum_{k=1}^N (X_{k,j} - r_k)^+$ 并且分配每个 $X_{i,j}$ 为:

$$X_{i,j} = \begin{cases} r_i, & X_{i,j} > r_i \\ X_{i,j} + \frac{K_j}{N_{i,j}} \times \frac{r_i}{\sum_{i=1}^N \frac{r_i}{N_{i,j}}}, & \text{otherwise} \end{cases} \quad (8)$$

重复该步骤直到每一个 $X_{i,j} \leq r_i$ 都在 j 列中。

(6) 从整个未分配的列中, 搜索在步骤5中分配的列的同类列。如果存在同类列, 则复制这个被分配的列并将其分配给其同类列集中的每个元素。如果存在任何满足 $X_{i,j} > r_i$ 的元素, 执行步骤5。在分配之后, 同类列集中的每个元素都被认为是已分配的, 并且不会参与进一步的计算。如果存在更多要分配的列, 就继续执行步骤2。否则, 结束算法。

当存在同类列集时, 应该使用改进算法, 因为当不存在同类列时, 搜索同类列的过程浪费时间。通常, 系统操作员知道同类列是否存在。因此, 系统操作员可根据预先知道的服务注册模式选择使用哪种算法。

3 仿 真

3.1 仿真环境

在具有多个 AS 的 IMS 网络中进行了实验。注册服务总数为8, AS 数量设置为4。每个 AS 的服务安排模式不同, 模式如表1所示。如果服务在 AS 上注册, 则在表中标记 O, 否则被标记为 X。为方便起见, 假设 S-CSCF 的数量为1。

表1 每个 AS 的已注册服务模式

服务	#1	#2	#3	#4	#5	#6	#7	#8
AS#1	O	O	O	O	O	X	X	O
AS#2	O	X	O	X	O	O	O	X
AS#3	X	X	X	O	X	O	O	O
AS#4	O	X	O	O	X	O	X	O

3.2 仿真结果与分析

图2显示了当每个服务具有相同的执行概率时, 现有的3种不同方案(循环, 专用 AS 触发和建议方案)中每个 AS 的接收触发数。从图中可以轻松验证两个所提方案和专用 AS 触发比循环具有更好的负载均衡性能。因为触发呼叫是以 AS 中注册服务的数量比率分配的, 所以循环触发在负载均衡中表现不佳。

另一方面, 如果服务执行的概率对于每个服务是相同的, 则提出方案或专用 AS 触发可以均匀地分配负载。

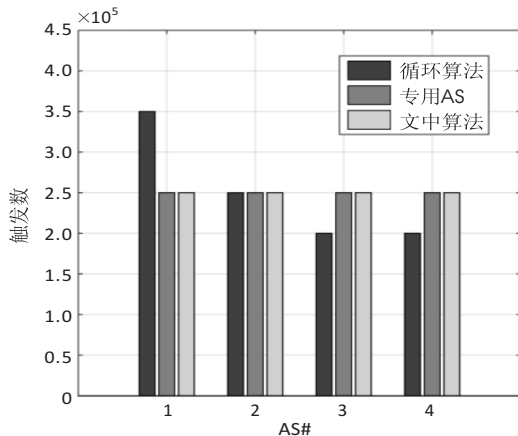


图2 每个 AS 在同类服务概率中接收的触发数

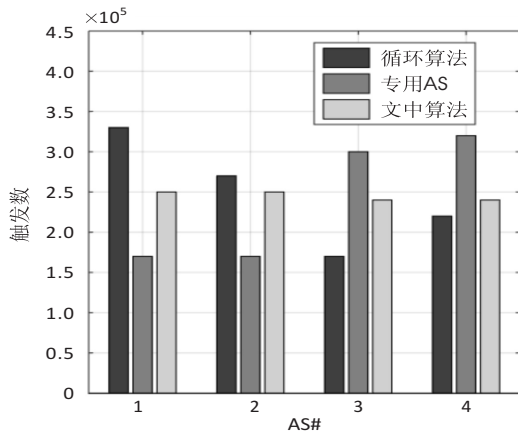


图3 每个 AS 在异构服务概率中接收的触发数

图3显示了当服务执行概率彼此不同时, 负载均衡性能。即使在相同的服务执行概率下表现出良好的性能, 专用 AS 触发也无法实现负载均衡, 这是因为服务执行的概率变得不同。然而, 即使服务概率彼此不同, 文中提出的方案也显示出了良好的负载均衡性能。

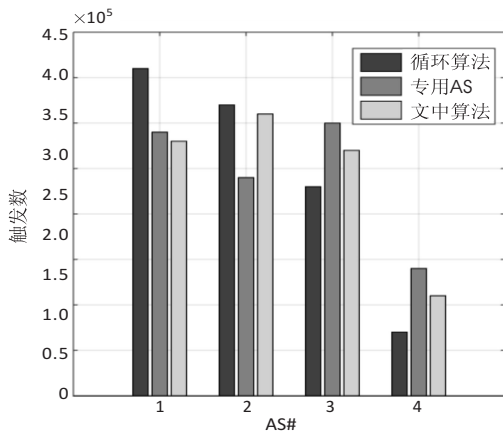


图4 存在系统错误时, 每个 AS 收到的触发数

图4显示了发生错误时每个 AS 的负载平衡性能。当触发 1 000 000 个呼叫时, AS#4 在随机时间内发生错误。在错误期间, AS#4 无法接收触发器, 所有呼叫都应分配给其他 3 个 AS。在该图中, 由于系统错

误,每个方案的 AS#4 触发器数量已减少。在图中,可以通过使用提议方案来配置 AS#1, AS#2 和 AS#3,即使 AS#4 出错,也可以接收相似数量的触发数据包。但是,在发生系统错误时,已证明循环或专用 AS 触发无法实现负载均衡,而该图可以很容易地验证,文中方案在错误环境中的负载均衡性能优于其他两种方案。

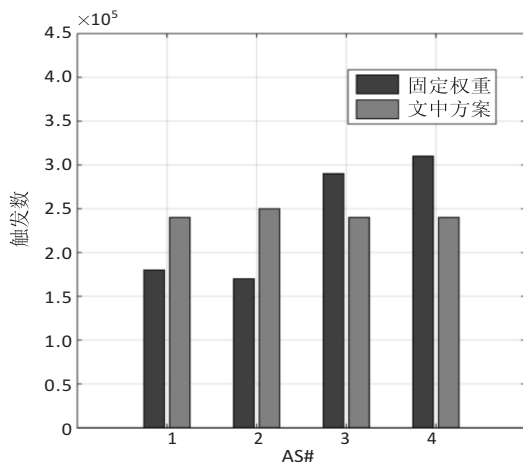


图 5 每个 AS 接收的具有不同服务执行概率的触发器数

图 5 显示了在操作期间更改服务执行概率时的负载均衡性能,比较了具有固定权重值的服务触发算法和文中提出的方案。可以发现,具有固定权重值的方案未能实现负载均衡,而文中提出的方案则成功实现。要纠正系统,操作员应根据结果手动改变权重值。文中方案以自动的方式调整权重值且具有良好性能,并且很好地适应系统环境的变化。据此可以得出结论,文中方案可以快速收敛到系统的变化。

实际上,在操作系统时,系统错误经常发生,服务执行概率通常是异构的,负载均衡失败会导致特定系统负载的增加,这就可能导致系统连续出错。仿真结果表明,当服务执行的概率异构或系统处于错误环境时,文中提出的方案比以前的方案表现出更好的负载均衡性能。此外,由于诸如用户数量或注册的服务模式之类的系统环境的改变,服务执行概率本身可以被改变。在这种情况下,快速调整新系统就变得尤为重要。使用手动设置的重量值,很难快速响应系统的变化,而文中方案可以自动设置权重值,并且能够高速收敛到系统变化。

4 结束语

提出了一种在 IMS 网络中使用多个异构 AS 时对服务触发数据包进行负载均衡的方案。可以从 EMS 的统计信息中获取每个服务的执行概率,以及从 HSS 下载的 iFC 是在哪个 AS 上注册哪些服务的信息。基于以上两条信息,可以通过调整每个 AS 的服务触发

概率来执行负载均衡。该负载均衡方案可以自动设置权重值,而现有方案基于操作员手动设置。因此,该方案可以更好地适应系统的变化,如系统错误,新服务的注册或新的 AS 添加到网络。此外,还提出了一种在同类型集存在时减少计算的方法。仿真结果表明,该方案比现有方案(例如循环和专用 AS 触发器)具有更好的负载均衡性能,尤其是当系统发生变化时。

参考文献:

- [1] 杨红梅,都云琪.移动通信网 IMS 系统相关标准及测试技术[J].电信网技术,2010(4):54-59.
- [2] 申 敏,魏 玄.分布式 IMS 架构与关键技术[J].数字通信,2009,36(6):36-40.
- [3] 魏永键,李文耀,王建利. Diameter 扩展协议在 IMS 中的应用分析[J].光通信研究,2008(3):54-56.
- [4] FAN Yaqin, YU Pengjun, LIU Meilin. SIP protocol in the IMS network and simulation [J]. Advanced Materials Research, 2013, 616-618:2051-2054.
- [5] 郭严赞,季新生,刘彩霞,等. IMS 网络 Diameter 协议流程漏洞挖掘[J].计算机工程,2013,39(9):6-11.
- [6] CHO Y H, JEONG M S, PARK J T. A scalable distributed architecture for multi-party conferencing using SIP [C]//IF-IP international conference on network and parallel computing. [s. l.]: Springer, 2004:374-381.
- [7] 李凌南,雷为民,林镜华,等.一种两阶段自适应的 IMS-Based IPTV 应用服务器负载均衡策略[J].计算机系统应用,2010,19(10):85-90.
- [8] XUN Zhaoyong, LIAO Jianxin, ZHU Xiaomin, et al. A group based service triggering algorithm for IMS network [J]. Journal of Electronics & Information Technology, 2008, 30(11): 1-6.
- [9] FAN Gongda, HUANG Chaochun, LIN Yibing, et al. Enhanced video phone services for NGN/IMS [J]. Wireless Communications & Mobile Computing, 2012, 12(7): 580-587.
- [10] XUN Zhaoyong, LIAO Jianxin, ZHU Xiaomin. On performance of 3GPP service triggering mechanism in IMS network [C]//34th Euromicro conference software engineering and advanced applications. Parma, Italy: IEEE, 2008:150-155.
- [11] 王青松,马瑞萍,李 爽. IMS 离线计费系统自适应均衡负载均衡算法的研究[J].计算机仿真,2015,32(12):341-344.
- [12] 刘俊轩.融合 IMS 可管可控的 P2P 流媒体系统方案设计与分析[D].北京:北京邮电大学,2010.
- [13] 黄 海,廖建新,朱晓民. IMS 实时计费系统集群的负载均衡研究[J].高技术通讯,2009,19(11):1117-1123.
- [14] 顾维青.固网 3G——平滑演进构建全 IP 的 IMS 架构[J].电信技术,2007(6):39-42.
- [15] 叶新荣.基于 IMS 架构 NGN 分析[J].信息技术,2006,30(8):21-23.