

GGSN 协议的一个流程问题分析及改进^①

夏亚梅^② 陈俊亮

(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

摘 要 在网关 GPRS(通用无线分组业务)支持节点(GGSN)项目与服务 GPRS 支持节点(SGSN)项的集成测试中发现了 GGSN 相关协议的一处流程交互的定义的缺陷,该缺陷导致在异常的流程处理中大量垃圾上下文的生成,并使得 GGSN 与 SGSN 两侧信息产生较大的不一致,这些不一致的信息将在较长时间影响系统的后续运行并产生运行错误。本文分析了此流程缺陷产生的原因并对相应流程提出一种改进的方法。实验环境下的测试结果表明,对协议做这样的改进可以完全解决这一问题并保证工程实现的高效性。

关键词 服务 GPRS 支持节点(SGSN),网关 GPRS 支持节点(GGSN),协议缺陷,PDP 上下文,一致性

0 引 言

通用无线分组业务(general packet radio service, GPRS)是一种基于全球移动通信系统(global system for mobile communications, GSM)的无线分组交换技术,提供端到端的、广域的无线 IP(internet protocol)连接^[1]。GPRS 网络是在 GSM 网络基础上新增两个节点——网关 GPRS 支持节点(gateway GPRS support node, GGSN)和服务 GPRS 支持节点(serving GPRS support node, SGSN)而形成的移动分组数据网络,此两节点之间通过 GPRS 隧道协议(GPRS tunneling protocol, GTP)相连^[2,3]。GGSN 与 SGSN 之间通过信令面的通信来传递流程控制信息,以保证 GGSN 与 SGSN 两侧信息的一致性和同步性。通信的相关流程如下:GPRS 分组从基站发送到 SGSN,SGSN 继续与 GGSN 进行通信;GGSN 对分组数据进行相应的处理,再发送到目的网络,如因特网或 X.25 网络。GGSN 通过配置一个包数据协议(packet data protocol, PDP)地址被分组数据网接入,它存储属于这个节点的 GPRS 业务用户的路由信息,并根据该信息将协议数据单元(protocol data unit, PDU)利用隧道技术发送到移动台(mobile station, MS)的当前的业务接入点,即 SGSN^[4-6]。

SGSN 通过发送激活请求、更新请求、去活请求、ECHO 等信令与 GGSN 进行信令面通信,同时 GGSN

有相应的请求应答给 SGSN。信令面通信完成后即可进行数据面的数据转发。GGSN 与 SGSN 之间信令面的通信主要来保证 GGSN 与 SGSN 两侧信息的一致性和同步性。

在 GGSN 项目与 SGSN 项目的集成测试中发现,系统运行一段时间后,GGSN 侧与 SGSN 侧出现了大量的信息不一致,这种不一致的信息进而转换为系统的垃圾上下文,并进一步影响着 GGSN 与 SGSN 两侧的流程控制、计费 and 审计等功能,对系统的正常运行造成了较大的负面影响。为了找到该问题产生的原因,我们进行了大量的测试及调试工作,发现工程实施严格按照协议中的规定,不会导致产生该问题。之后,再进一步对相关协议进行详细分析时发现问题真正原因是 GGSN 相关协议 29060(3G)及 0960(2.5G)中,对于重启的异常流程处理存在着一处漏洞,该漏洞导致 GGSN 与 SGSN 两端信息产生不一致。本文详细的分析了问题产生的原因并提出了一种可行的协议改进方案。

1 相关背景

相关的协议规定如下:

(1)SGSN 或 GGSN 运行时须保留最近一次本地的重启次数和与它有连接的 GGSN 或 SGSN 的重启次数,如在 GGSN 侧,与它有连接的 SGSN 的重启次数保留如下表 1 所示。

^① 973 计划(2007CB307100)和国家自然科学基金(60432010)资助项目。

^② 女,1976 年生,博士生;研究方向:通信软件,服务组合;联系人,E-mail: xiayamei@gmail.com
(收稿日期:2007-11-13)

表 1 GGSN 侧保留的 SGSN 的重启次数

SGSN	SGSN1	SGSN2	SGSN3	...
重启次数	3	12	18	...

表 1 中 SGSN1, SGSN2, SGSN3 表示与 GGSN 有连接的 SGSN。重启次数通过 SGSN 与 GGSN 信令中的一个可选信元 Recovery 来传递。GGSN 与 SGSN 在建立首次连接时必须保存对端的重启次数,在之后的运行过程中,当发现对端信令中携带此信元且其值与原来保留的对端的重启次数值不同时,则认为对端重启了。当 GGSN 或 SGSN 在认为与它连接的 SGSN 或 GGSN 重启之后,应采取相应的处理流程,如在 GGSN 侧,当 GGSN 检测到 SGSN 的信令报文中携带了 Recovery 信元且其值与之前保留的此 SGSN 的重启次数不同时,则认为此 SGSN 重启了。流程图如图 1,设 GGSN 与 SGSN2 有连接且 GGSN 保存的 SGSN2 之前的重启次数为 12。

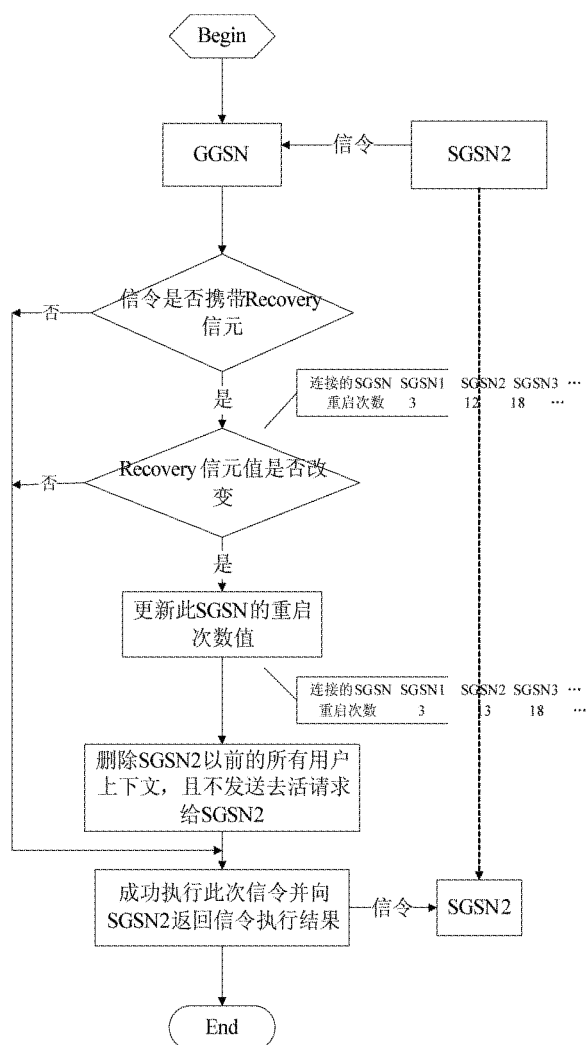


图 1 GGSN 侧信令处理流程

以上流程图中,当 GGSN 收到 SGSN2 携带 Recovery 值且其值为 13 的信令报文时,GGSN 的处理为:更新此 SGSN2 的重启次数;删除 SGSN2 以前激活的所有用户上下文且不发送去活请求给 SGSN2,之后再成功执行这一次的信令。详细规定参见协议 3GPP TS 23.007。

(2) 协议规定: Recovery 信元为可选信元,当 SGSN 与 GGSN 进行第一次连接时各自发送给对方的第一个信令报文中必须携带此信元。如果 GGSN 或 SGSN 重启,重启的一方在发送给对方的第一个信令报文中也必须携带此信元,其他情况下为可选信元。

(3) 另有相关协议规定:在 GGSN 侧,GGSN 的重启次数存储在非易失存储介质中,而与 GGSN 相连的 SGSN 的重启次数存储在易失存储介质中^[1,4,7]。

以上协议规定表面上看是合理的,但隐藏了问题出现的可能性。由于 SGSN 侧与 GGSN 侧处理流程相同,以下主要从 GGSN 侧讨论。

2 问题出现过程分析

GGSN 重启将丢失存储在易失存储介质中的所有 SGSN 的重启次数值,如表 2。SGSN 的所有当前连接用户的上下文信息也将同时丢失,且 GGSN 与 SGSN 之间的信令路径也不再存在。这里的用户上下文是指当前连接用户的用户标识,其中存储着用户信息,它的存在表示用户的状态及连接正常,它被删除则表示用户已断开连接或网络出现异常;信令路径指 GGSN 与 SGSN 为了进行信令面通信而建立的连接,当第一个用户激活时信令路径建立,至少有一个正常用户存在时信令路径存在,在信令路径上 GGSN 与 SGSN 可以进行信令交互。

表 2 SGSN 的重启次数 (“-”表示“不存在”)

SGSN	SGSN1	SGSN2	SGSN3	...
重启次数	-	-	-	...

此时,GGSN 只能等待着 SGSN 向它发起连接请求以重新建立信令路径。之后的流程为当 SGSN 发送第一条信令时,GGSN 与 SGSN 之间的信令路径重新建立。根据协议规定,GGSN 重启后给 SGSN 的第一条信令报文中必须携带 Recovery 信元,且其值为当前的重启次数值;SGSN 检测到 Recovery 信元并对信元值进行比较后判断得知此 GGSN 重启了,于是

删除此 GGSN 之前的所有用户上下文,以保持 GGSN 与 SGSN 两端信息的一致性^[1,4]。但协议在此时忽略了一个问题:因为 SGSN 未重启,按照协议中的规定,SGSN 给 GGSN 的信令可以继续不携带 Recovery 信元,但对 GGSN 侧,GGSN 系统已经丢失 SGSN 的重启次数,却一直检测不到 Recovery 值的出现,因而不能得到 SGSN 的重启次数,业务运行虽然表面正常,但不存在的 SGSN 的重启次数隐藏着将要出现的问题。业务在没有其它异常的情况下将一直正常运行,直到 GGSN 重新得到此 SGSN 的 Recovery 值,此

时问题就会必然地显现出来。这种情况持续越久,期间发生的业务和激活的用户上下文数目越多,之后产生的不一致信息也就越多。

以上的流程可以用一个 SGSN 为例来说明(参见图 2)。从图 2 看到,GGSN 重启后一段时间内 SGSN 的重启次数为“不存在”,在 SGSN 重启次数不存在时得到一个 SGSN 的重启次数值,会因 SGSN 重启次数基准值的丢失而无法进行比较,SGSN 是否重启也将无法判断,因而之后的处理将无法完全正确地进行。

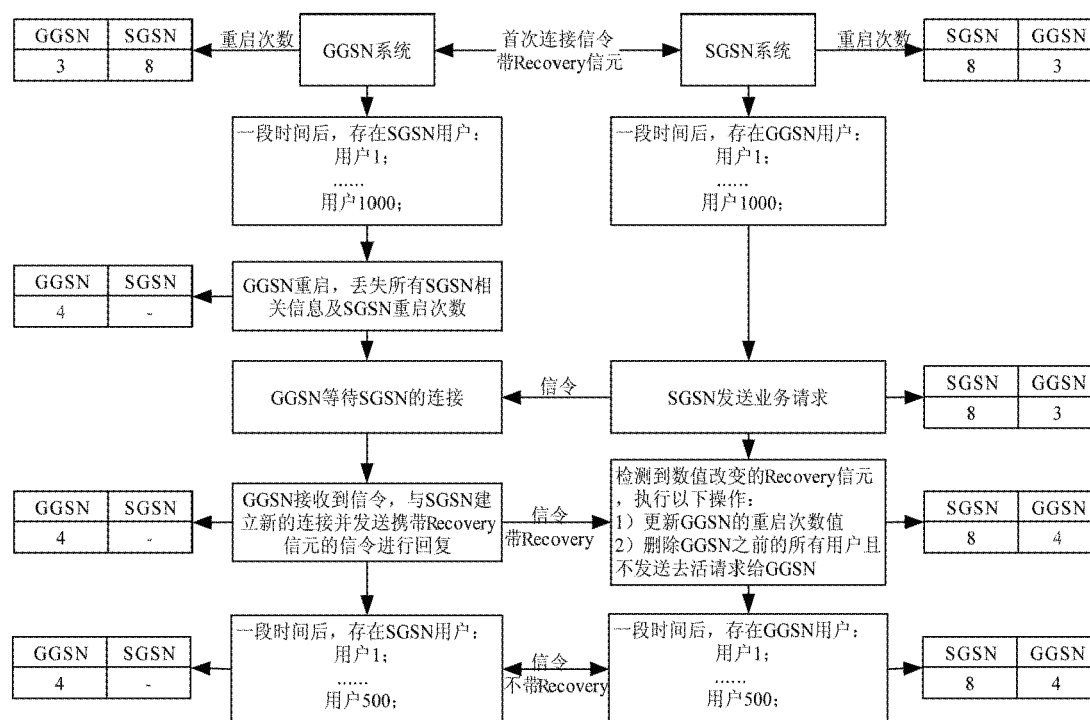


图 2 SGSN 重启次数丢失流程

GGSN 侧的处理可以分为按照 SGSN 重启和 SGSN 未重启两种情况来处理,两种处理都会因考虑到一种情况、忽略另一种情况而发生错误处理。两种情况下错误处理的流程如下:

(1)按照 SGSN 重启处理,但事实上 SGSN 并未重启的情况(接图 2)。

如果 SGSN 没有重启,只是发送了一个携带 Recovery 信元值的信令报文给 GGSN,将会产生如下结果:虽然 SGSN 没有重启,但它一段时间内的用户上下文在 GGSN 侧均被误删除,且根据协议此时 GGSN 并不通知 SGSN,因此 SGSN 无法立即获得这些正常

上下文被 GGSN 删除的信息,导致 SGSN 一端正常运行的用户上下文变为垃圾用户上下文(Waste PDP),从而造成 SGSN 与 GGSN 之间信息不一致,处理流程参见图 3。

(2)按照 SGSN 不重启处理,但事实上 SGSN 重启的情况。

如果此时 SGSN 重启了,SGSN 就丢失了所有的用户上下文,而在 GGSN 一侧却认为 SGSN 没有重启,所以 GGSN 依然保留着此 SGSN 已经丢失的所有用户上下文信息,造成了 GGSN 一侧的垃圾用户上下文,处理流程参见图 4(接图 2)。

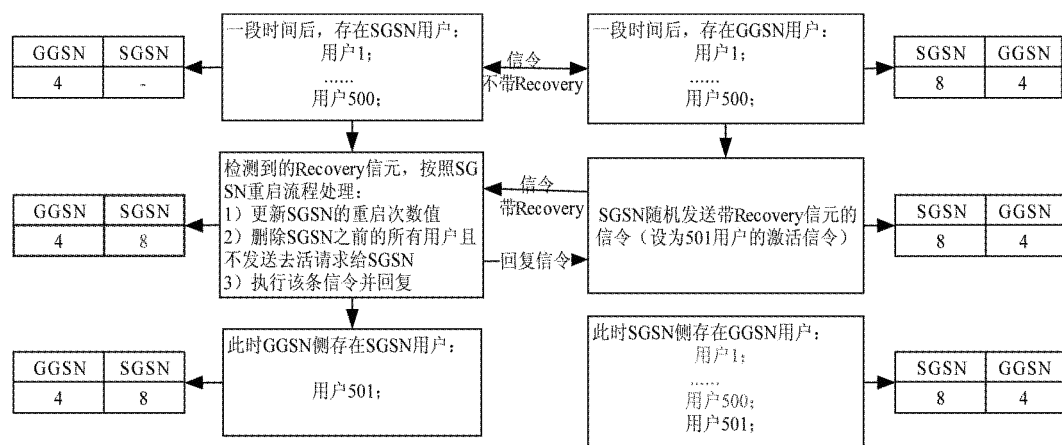


图 3 按 SGSN 重启处理产生的错误处理流程

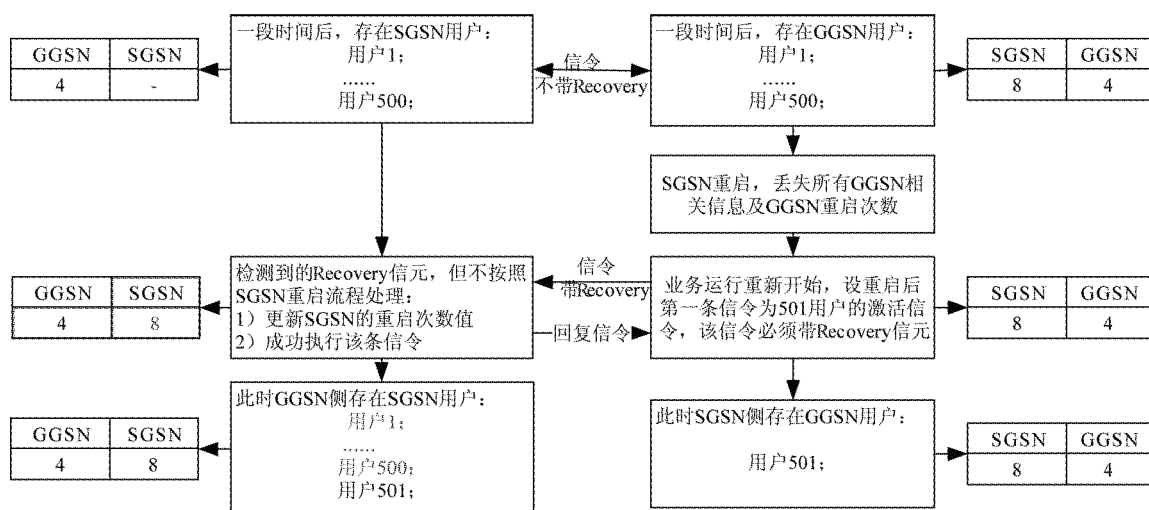


图 4 按 SGSN 不重启处理产生的错误处理流程

3 工程实现示例

上述问题是由于协议中的流程定义不严格引起的。协议在制定时对复杂的异常处理流程细节考虑不够,因而导致在工程实现时出现问题。

下面以一种工程中的实现方案为例加以说明。实现严格按照 29060(3G)及 0960(2.5G)等 GGSN 相关协议,由第 1 节中(3):“在 GGSN 侧,GGSN 的重启次数存储在非易失存储介质中,而与 GGSN 相连的 SGSN 的重启次数存储在易失存储介质中”,因而在工程实现时将 GGSN 系统中 GGSN 的“重启次数”存储在一块 EEPROM 中(非易失存储介质),每次重启时由处理器从 EEPROM 中读入内存,每重启一次,EEPROM 中的 GGSN“重启次数”值会自动增 1;而 GGSN 系统中 SGSN 重启次数的保存,是每次 GGSN 重启后在内存中动态的分配一个数组(易失存储介质),用

以记录运行时与它进行连接的 SGSN 的重启次数。为了提高运行效率,避免对 Recovery 信元及其信元值的重复检测,只在初次连接和重启之后的第一次连接时发送给对方的第一个信令报文中和运行过程中随机发送的 ECHO 信令中携带 Recovery 信元,其余情况下均不携带此信元。GGSN 在得到有效的 SGSN 的重启次数之前,SGSN 的重启次数被初始化为 0。即表 2 中不存在的 SGSN 的重启次数用 0 进行替换,如表 3 所示。

表 3 工程实现中 SGSN 重启次数不存在时的表示

SGSN	SGSN1	SGSN2	SGSN3	...
重启次数	0	0	0	...

这里虽然将不存的重启次数用数值 0 做了替换,但问题的实质没有改变。具体流程分析此处不再赘述。

4 容错机制有限性分析

为解决上述问题,首先考虑应用协议本身提供的容错机制。容错机制是考虑到网络拥塞导致数据包的丢失,从而可能导致 GGSN 端与 SGSN 端的信息少量不一致而设立的。但协议中提供的两种容错机制都不能有效地解决本文提出的问题,分析如下:

(1)一种是及时地发送 ECHO 请求以减少这种不一致的发生,但该方法无法彻底解决问题,只能减少产生的垃圾上下文的数量,原因有两点:一是协议规定 ECHO 请求的发送间隔应大于 60s,二是 GGSN 发送 ECHO 请求必须有信令路径的建立,而信令路径建立时至少已经存在一个用户(用“用户上下文”表示)^[1],在这种意义上,以上问题产生的前提已经存在了,即至少有一个用户上下文的信息会不一致。

(2)另一种容错机制 Idle 检测可清除某些长期不用的空闲上下文,但这种检测开关一般是人为控制的,且检测本身非常耗费资源,会大大降低系统的吞吐量及运行效率,一般很少使用。

另外,在工程实施中,应用以上容错机制进行了测试,没有收到好的效果。因此应用协议的容错机制解决本文提出问题的方法是不可行的,需要采用其他的更有效的解决方案。

5 解决方案

该问题是由于协议流程定义的不严格引起的,因此最根本的解决方法就是改进协议。协议的改进需要综合考虑实现的可行性和效率,如以下方案 1 和方案 2 是不可行的解决方案,方案 3 是可行的解决方案,分析如下:

方案 1: 把 Recovery 规定为必选信元。

如果把 Recovery 规定为必选信元,则 GGSN 与 SGSN 之间的每个信令报文都会携带 Recovery 值,如果有重启的情况发生,对端会在第一时间知道,当然也就避免了上述问题的发生。但这种方法造成了效率的严重降低。因为在正常运行情况下对每个信令报文都实施 Recovery 值检测并进行比较会造成大量资源浪费。所以,Recovery 应该规定为可选信元,原协议中的规定是合理的。

方案 2: 在 GGSN 侧把 SGSN 的重启次数值存放在非易失存储介质中。

协议规定在 GGSN 侧,SGSN 的重启次数存储在

非易失存储介质中而与它存在连接的 SGSN 的重启次数存储在易失存储介质中,协议这样的规定是合理的。一个 GGSN 可以和多个 SGSN 连接,每个 SGSN 的信息也会常常发生变化。大量、频繁地向一个非易失存储介质中写数据是不可行的。

本文在以上分析的基础上提出了下述可行的解决方案:

方案 3: 协议中增加规定,在收到 Recovery 信元值改变的信令后在紧接着的下一次的信令中携带 Recovery 信元回复。

从图 3 及图 4 中可以看出,不一致的信息产生于 GGSN 重启后到获得第一个携带 Recovery 信元的 SGSN 信令报文这段时间之内,也就是这段时间生成的业务信息可能会在以后造成不一致,所以缩短这个时间差就可以减少不一致信息的数量,当这个时间差之间没有业务生成时不一致的信息数量就会变为零。所以协议应补充规定:当发送方发送携带 Recovery 信元且其值改变了的信令报文后,接收方应在紧接着的下一次的信令报文中携带本地的 Recovery 信元回复,这样就可以保证这个时间差之内没有其它业务生成。协议作此简单的修改就可以避免上述不一致问题的产生了。

以上方案 3 在 3G 实验环境中进行了验证,实验结果表明该改进方案很好地解决了这一问题,且改动的工作量很小,没有增加系统的额外开销,也没有影响到其它流程的运行。

6 结 论

本文对 GGSN 协议 29060(3G)及 0960(2.5G)中的一处流程处理缺陷进行了分析,并在此基础上提出了一种协议的改进方案,该方案可以完全避免由此协议缺陷引起的信息不一致问题,并能保证工程实现中的可行性及高效率。

参考文献

- [1] 3GPP, 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface (Release 7), Tech Spec. 3GPP TS 29.060V7.2.0, 2006
- [2] Chia C Y, Chang M F. Channel allocation for priority packets in the GPRS network. *IEEE Communications Letters*, 2006, 10(8):602-604
- [3] Chen J C, Chen W M, Lin H W. Design and analysis of

- GPRS-WLAN mobility gateway (GWMG). In: Proceedings of IEEE International Conference on Communications, Seoul, Korea, 2005. 918-923
- [4] 3GPP, 3rd Generation Partnership Project; Technical Specification Group Core Network; General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp Interface, Tech Spec. 3GPP TS 09.60 V7.10.0, 2002
- [5] 3GPP, 3rd Generation Partnership Project; General Packet Radio Service (GPRS); Service description; Stage 2. Tech. Spec. 3GPP TS 23.060, 2002
- [6] Xenakis C, Merakos L. On demand network-wide VPN deployment in GPRS. *IEEE Network*, 2002, 16(6):28-37
- [7] 3GPP, 3rd Generation Partnership Project; Technical Specification Group Core Network; Restoration procedures, Tech Spec. 3GPP TS 23.007, 1999
- [8] Chakravorty R, Clark A, Pratt I. Optimizing Web delivery over wireless links: design, implementation, and experiences. *IEEE Journal on Selected Areas in Communications*, 2005, 23(2):402-416

Analysis and improvement of a flow in gateway GPRS support node protocols

Xia Yamei, Chen Junliang

(State Key laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract

A limitation was found in the definition of flow interaction in the gateway GPRS support node (GGSN) protocols in the engineering of integrating the GGSN with the serving GPRS support node (SGSN). This limitation causes many waste PDP contexts and the information inconsistency between GGSN and SGSN, which can impact the subsequent system running and even incur mistakes. This paper analyzes the cause of the flow limitation and in terms of the feasibility and efficiency of a project, puts forward a protocol improvement. The experimental results prove that the protocol improvement can completely resolve the above mentioned problems and ensure the effectiveness in engineering implementations.

Key words: serving GPRS support node (SGSN), gateway GPRS support node (GGSN), protocol limitation, PDP context, consistency