

第15章 无线事务协议规范

15.1 范围

事务协议提供了交互式“浏览”(请求/响应)程序的服务需求。在一次浏览会话当中,客户端请求服务器(服务器可以是固定的或者是移动的)给予信息,服务器回答其要求的信息。这种请求/响应的往返过程在本文件中被称作“事务”。本协议的目的是可靠地发送事务,并平衡应用程序所需的可靠性和发送这种可靠性所需的开销。

WTP运行在数据报服务和可选择的安全服务之上,它被定义为一个轻量级的面向事务的协议,适合于瘦客户(移动台)的实现,可以在无线数据报网络上有效的运行。使用 WTP的好处如下:

- 提高了数据报服务的可靠性。如果使用数据报服务,重传和确认是必需的,但使用 WTP后,减轻了其上层重传和确认的负担。
- 提高了面向连接的服务效率。WTP没有明确的连接建立或连接拆除阶段。
- WTP是面向消息的,为了象“浏览”一样的事务它被设计为面向服务的。

15.2 协议概述

15.2.1 协议特征

以下部分概括了WTP的特征:

- 三个级别事务:
 - 第0级:无结果消息的不可靠消息调用。
 - 第1级:无结果消息的可靠消息调用。
 - 第2级:有可靠结果消息的可靠消息调用。
- 可靠性通过使用单一的事务标识、确认、重复拆卸以及重传来达到。
- 没有明确的连接建立或者连接拆卸阶段。明确的连接打开和 /或连接关闭给通信链路增加了过多的开销。
- 用户到用户的可靠性。WTP用户确认每一个收到的消息(这是可选的)。
- 事务的最后确认可以包含与事务有关的其他信息,例如性能的测量(这是可选的)。
- 在适当的地方可以使用串联,以便在数据报传输的一个服务数据单元中传送多个协议数据单元。
- 面向消息。互相交换的基本单元是一个完整的消息,而不是一个字节流。
- 提供了机制而使重播的事务数最少,这些事务数是复制分组的结果。
- 未完事务的放弃,包括客户端和服务器未发送数据的清零,这种放弃可由用户通过删除一个已请求的服务来触发。
- 为了消息调用的可靠性,成功和失败都要报告。如果响应者不能处理调用,就返回给发起者一个放弃消息而不是结果。

- 协议允许异步事务。当数据可以利用时，响应者回送结果。

15.2.2 事务级别

以下部分描述了WTP的事务级别。发起一个事务的WTP提供者被称作发起者,对事务作出响应的WTP提供者被称作响应者,事务级别由发起者设置,在调用消息中发送给响应者。事务级别不能协商。

1. 第0级 无结果消息的不可靠消息调用

第0级事务提供一种不可靠的数据报服务,它由不需要“可靠推操作”服务的应用程序使用。这个级别的目的是增大事务服务的能力,它适用于那些在一个现有的WTP会话的上下文中偶尔使用WTP发送数据报的应用程序。它不是传送数据报的主要手段,把数据报服务作为传送数据主要手段的应用程序应该使用WDP[WDP]。

第0级事务的基本行为如下:一个调用消息从发起者发送到响应者,响应者不确认这个调用消息,发起者也不执行重传。对于发起者来说,调用消息一被发出这个事务就终止了,对于响应者而言,调用消息一被收到这个事务就终止了。这种事务是无状态的,不能被放弃。

2. 第1级 无结果消息的可靠消息调用

第1级的事务提供了一种可靠的数据报服务。它可以由需要“可靠推操作”服务的应用程序使用。

第1级事务的基本行为如下:一个调用消息从发起者传送到响应者,响应者回送对这个消息的确认。为了处理可能发生的确认重传,响应者要在确认消息发出后的一段时间内保持状态信息,这种重传可能是由于确认丢失了和/或发起者重传了这个调用消息;在发起者端,当确认收到时事务终止。这种事务可在任何时候被放弃。

如果可利用用户的确认功能,在确认消息被送回到发起者之前,响应端的WTP用户首先要证实调用消息。

3. 第2级 有可靠结果消息的可靠消息调用

第2级事务提供了基本的调用/响应事务服务。一个WSP会话可以包括几个这种类型的事务。

第2级事务的基本行为如下:一个调用消息从发起者发送到响应者,响应者明确地应答一个结果消息,这个结果消息中隐含着对调用消息的确认。如果响应者对这个调用服务的时间比它的确认计时器中设定的时间长,响应者可能在发送结果消息前,发送一个“坚持”确认,这样,可以避免发起者不必要地重传调用消息。响应者把结果消息发送到发起者,发起者要对结果消息进行确认,在确认发出之后,发起者也将状态信息保持一段时间,这是为了在得到丢失消息和/或响应者重传的结果消息时处理可能的确认重传。在响应端,接收到确认事务终止。这个事务可以在任何时候终止。

如果可利用用户的确认功能,在确认消息被送回发起者之前,响应端的WTP用户首先要证实调用消息。

15.2.3 与其他协议的关系

本节描述了WTP与其他WAP协议之间的关系。有关WAP体系结构完整的描述,请参阅[WAP]。表15-1描述了为WTP用户提供服务的地点:

表15-1 为WTP用户提供服务的地方

	WTP USER (如 WSP)
WTP	<ul style="list-style-type: none">• 事务处理• 重传, 重复拆卸, 确认• 串联和分离
[WTLS]	<ul style="list-style-type: none">• 可选择的压缩• 可选择的加密• 可选择的鉴权
数据报传输 (例如 WDP)	<ul style="list-style-type: none">• 端口号寻址• 拆组和重组 (如果提供)• 错误检测 (如果提供)
承载网 (例如 IP、GSM SMS/USSD, IS-136 GUTS)	<ul style="list-style-type: none">• 路由选择• 设备寻址 (IP地址, MSISDN)• 拆组和重组 (如果提供)• 错误检测 (如果提供)

WTP在数据报传输服务之上运行，它的协议数据单元是数据报数据的一部分，既然数据报是不可靠的，就需要 WTP执行重传及发送确认消息以便为 WTP用户提供可靠的服务。WTP还负责将多协议数据单元串联到一个传输服务数据单元之中（如果可能）。

为WAP规定的数据报传输定义在 [WDP]中。需要数据报传输是为了给到来的数据报提供路由，以寻找到正确的 WDP用户。通常，WDP用户由一个单独的端口号来标识，目前还没有为WTP-WSP分配的数据报端口。WDP的责任是为WDP用户提供数据报服务，而不关心承载网络类型的容量。幸好，数据报服务是一种通用的传输机制，大多数的承载网络已经提供了这种服务。例如，基于 IP，使用UDP以提供数据报服务。

承载网络负责为数据报选择到目标设备的路由，它的类型（IP地址或电话号码）的不同导致了寻址方法的不同。另外，一些网络使用动态地址分配，必须联接一个服务器以找到某个设备的当前地址。在WAP栈中的网络地址可能包括承载类型及地址（例如，IP;123.456.789.123）。不同地址空间的多种承载网络的来去数据到同一个 WAP栈的复用还没有规定。WAP规定了数据报服务层之上的协议。

15.2.4 安全性考虑

WTP没有安全机制。

15.2.5 管理实体

WTP管理实体用作 WTP层和设备环境之间的接口，它为 WTP层提供可能会影响 WTP设备正确操作的环境变化信息。

WTP协议根据这样一个假定来设计，即能够在传送和接收数据的环境中操作。例如，这个假定包括了移动设备必须提供的这样一些基本特定：

- 移动台处在能够承载业务的覆盖区。
- 移动台有足够的能量并且电源是处于打开状态。
- 移动台中有足够的资源（指处理和内存）供 WTP使用。
- WTP作了正确的配置。

- 用户希望接收或发送数据。

WTP管理实体监视正在进行服务的状态和移动台适应环境的能力，并通知 WTP层是否存在一个或者多个不可用的假设服务。例如，如果移动台漫游到了承载业务的覆盖区之外，承载管理实体应该向 WTP管理实体报告承载者不能再发送或接收了。接着，WTP管理实体应该指示WTP层关闭所有在这个承载上活动的连接。其他的例子，如低电压等，由 WTP管理实体用类似的方式来处理。

除了监视移动环境的状态之外，WTP管理实体可能被用作设置不同的 WTP参数的用户接口，例如设备地址。它还可执行用户可利用的一些功能，例如“断开所有数据连接”。一般说来，WTP管理实体处理所有与初始化、配置、动态重新配置有关的问题，以及属于 WTP层的资源。

由于WTP管理实体必须在移动设备的各种组件之间实现交互，因此 WTP管理实体的设计和实现已经超出了WTP规范考虑的范围，是一个具体实现的问题。

15.2.6 静态WTP的一致性条款

静态一致性条款定义了一个可以保证其实现具有互操作性的最小的 WTP特征集合。

WTP特征专门用于WTP用户。当WSP作为WTP用户时，其特征还与 WSP协议是运行在客户端还是在服务器端有关。在表 15-2中，列出了当WSP作为用户时必需的（M）和可选的（O）WTP特征。

表15-2 当WSP作为用户时WTP静态一致性条款

功 能	类 型	WSP 客户端	WSP 服务器
事务级0	发起	M	M
	响应	M	M
事务级1	发起	M	M
	响应	M	M
事务级2	发起	M	O
	响应	O	M
用户确认		M	M
串联		O	O
分离		M	M
重传直到确认		M	M
放弃事务		M	M
版本处理		M	M
错误处理		M	M
最后一个确认中的信息		M	M
异步事务		O	O
事务标识查证	发起	O	O
	响应	M	M
传输信息项目	错误	M	M
	信息	M	M
	选择	O	O
	PSN	O	O
分割和重组，可以选择重传和分组包		O	O

如果WTP提供者被要求执行一个其不支持的程序，这个事务必须被放弃并返回一个适当的错误代码。例如，如果一个不支持级别 2 的提供者收到一个级别 2 的事务，它放弃此事务，并返回NOTIMPLEMENTEDCL2放弃码。

为加强WTP服务分割和重组（SAR）及选择性重传可能被执行。如果在 WTP中不执行 SAR，这个功能应该由栈中的其他层来提供。例如，在 IS-136中SSAR子层处理SAR，在IP网络中 IP[RFC791]处理 SAR，而对 GSM SMS/USSD来说，通过使用 SMS串联来达到 SAR[GSM0340]。执行WTP SAR的动机是实现可选择重传机制，在发送大量消息时，这可以提高协议空中接口的效率。

是否支持WTP SAR由发起者在事务被调用的时候决定,表15-3给出了WTP发起者和响应者如何在执行和不执行WTP SAR的WTP提供者之间确认互操作的能力。

注意，如果一个不支持 WTP SAR的响应者从支持WTP SAR的发起者收到一个未分组的消息，不需要放弃此事务，发起者不会知道响应者不支持 WTP SAR。

表15-3 支持和不支持WTP SAR的WIP提供者间的互操作

响 应 者	支持WTP SAR	发 起 者	不支持WTP SAR
支持WTP SAR	完全交互操作		响应者不必以一个 WTP分割消息来响应
不支持WTP SAR	响应者放弃该事务，并返回 NOTIMPLEMENTEDSAR放弃码 不使用 WTP SAR 的事务，发起者必须重新发送这个事务		完全交互操作

15.2.7 其他的WTP用户

本协议用来在不可靠的数据报服务上提供具有可靠的事务服务的 WSP，然而，这个协议可用于具有类似通信要求的其他应用程序中。

15.3 层间通信元素

15.3.1 使用的符号

1. 服务原语及参数的定义

层间通信及同一层的实体间的通信由服务原语来完成。服务原语以抽象的方法提供了在事务层及相邻层之间的信息和控制的逻辑交换,它们没有定义或限制各种实现。

服务原语包括命令及这些命令各自与其他层要求的服务相关的响应。一个原语的通用语法是：

X - Generic name . Type (Parameters) [X - 一般的名字.类型 (参数)]

在这里，X指提供服务的层。在本规范中，X为“TR”，即传输层。

一个WTP层服务原语的实例是TR调用的请求。

服务原语不同于一个应用程序接口（API），它并不意味着任何特定的API实现方法。服务

原语是一个抽象的方法，它描述由协议层提供给上一层的服务。如何把这些概念映射到一个实际的API及其相关的语法是一个实现的问题，这个问题不属于本规范的范畴。

2. 原语类型
本规范定义的原语类型在表 15-4中列出：

表15-4 原语类型

类 型	缩 写	描 述
请求	req	当一个较高层向其下层请求服务时使用
指示	ind	提供服务的层使用此原语类型通知其上一层与对等层（例如请求原语的调用）或是服务提供者（例如一个由协议产生的事件）相联的操作
响应	res	某一层使用响应原语类型以确认从下一层收到了指示原语类型
确定	cnf	提供请求服务的层使用确定原语类型来报告操作已成功完成

3. 服务参数表
服务原语用表来定义，表中指明了哪些参数是可用的，以及它们怎样与不同的原语类型一起使用。例如，一个简单的确定原语可能被表 15-5定义：

表15-5 简单的确定原语的定义

参 数	传输原语 TR-PRIMITIVE			
	REQ	IND	RES	CNF
参数1	M	M(=)	—	—
参数2	—	—	O	C(=)

在表15-5中，在TR-primitive.request中参数1总是存在。参数2可以在TR-primitive.response中定义，在这种情况下它必须存在且在相应的 TR-primitive.confirm中有同样的值，否则，它不能存在。

如果某个原语类型是不可用的，那一列会被省略。原语类型栏中的符号含义在表 15-6中定义。

表15-6 参数使用符号的说明

M	参数的存在是强制性的，即它必须存在
C	参数的存在与否依环境而定，依赖于其他参数的值
O	参数的存在由用户选择；它可被省略
P	参数的存在与否由服务提供者选择；某一个实现可能不提供它
ñ	参数是默认的
*	参数的存在由底层协议决定
(=)	参数值等于上一个服务原语的对应参数值

15.3.2 对下层的需求

WTP被规定为在数据报服务之上运行。数据报服务必须具有下述功能：

- 有端口号以便为到来的数据报选择到 WTP层的路由。
- 向上传送到WTP层的SDU的长度信息。

数据报服务可能完成下述功能：

• 错误检测。例如，使用校验码。

另外，希望下层提供分割和重组（SAR）。然而，这通常由数据报层的下一层执行。例如，在IP网中，IP协议处理SAR。

15.3.3 为上一层提供的服务

1. 传输调用（TR-Invoke）

这个原语用来发起一个新的事务见表 15-7。

表15-7 传输原语TR-INVOKE

参 数	REQ	IND	RES	CNF
源地址	M	M(=)		
源端口	M	M(=)		
目的地址	M	M(=)		
目的端口	M	M(=)		
确认类型	M	M(=)		
用户数据	O	C(=)		
级别类型	M	M(=)		
返回信息			O	C(=)
处理	M	M	M	M

- (1) 源地址
源地址就是向 WTP层发出请求的那个设备的唯一地址。源地址可能是一个 MSISDN号、IP地址、X.25地址或其他标识符。
- (2) 源端口
与源地址有关的源端口号。
- (3) 目的地址
用户数据的目的地址，这些数据被提交到 WTP层。目的地址可能是一个 MSISDN号、IP地址、X.25地址或其他标识符。
- (4) 目的端口
与目的地址有关的目的端口号，用于被请求的或现存的事务。
- (5) 确认类型
此参数用来打开或关闭用户确认功能。
- (6) 用户数据
WTP协议携带的用户数据。递交到 WTP层或从WTP层收到的数据单元也被称为服务数据单元（Service Data Unit），这是高层提交给WTP层传输的完整的数据单元（消息）。WTP层将传输服务数据单元，并且在不对其内容进行任何处理的情况下将这些数据送到目的地。
- (7) 级别类型
指出WTP事务级别。
- (8) 返回信息
附加的用户数据被传送到发起者，表示事务完成。
- (9) 处理
事务处理是返回给上一层的索引，以使上层可以标识事务并将接收到的数据与一个活动

的事务进行关联。TR-Handle单一地标识一个事务，它是事务源地址、源端口、目的地址及目的端口的别名。

TR-Handle仅有本地意义。

2. 传输结果（TR-Result）

此原语用来将上次发起的事务的结果送回见表 15-8。

表15-8 传输原语TR-RESULT

参 数	REQ	IND	RES	CNF
用户数据	O	C(=)		
退出信息			O	C(=)
处理	M	M	M	M

3. 传输放弃（TR-Abort）

此原语用来放弃一个存在的事务（见表 15-9）。

表15-9 传输原语TR-ABORT

参 数	REQ	IND
放弃码	O	C(=)
处理	M	M

放弃码
放弃码指出了事务被放弃的原因，它可包括WTP协议所产生的放弃码及用户定义的本地放弃码。

15.4 操作的级别

15.4.1 第0级事务

1. 动机
第0级是一种不可靠的数据报的服务，它可被 WSP[WSP]使用。例如，在使用同一个套接字联合的会话中执行不可靠的“推”操作。

这个级别的目的是增大服务的能力，以便应用程序在使用 WTP的现存的会话上下文中偶尔发送数据报，这些应用不是以发送数据报为主要手段。需要数据报服务的应用程序应该使用[WDP]中定义的WDP。

2. 协议数据单元（PDU）

使用了以下的PDU：

- 调用PDU（Invoke PDU）

3. 过程

第0级事务由WTP用户通过将事务级别参数设置为0的TR-Invoke请求原语来发起。WTP提供者发出调用消息，成为事务的发起者；远程的WTP提供者接收调用消息，成为事务的响应者。发起者不等待或者期望响应。如果响应者收到调用消息，则立即接受它，不执行重复的拆卸或证实程序。然而，客户端必须在每一个事务间增加TID计数器，而服务器不能更新其缓存的TID。

WTP提供者必须支持这个事务级别,它必须既能充当发起者又能充当响应者。

15.4.2 第1级事务

1. 动机
- 第1级事务是一种无结果消息的可靠调用消息 ,此类事务可被 WSP[WSP]用来实现可靠的“推”服务。
2. 服务原语序列
- 表15-10描述了合法的服务原语序列 ,列在第一栏中的服务原语后面仅能跟随着行中标记了“X”的原语。

表15-10 用于第1级事务的原语序列

	TR-INVOKE			TR-ABORT	
	REQ	IND	RES	CNF	IND
TR-Invoke.req					
TR-Invoke.ind					
TR-Invoke.res		X			
TR-Invoke.cnf	X				
TR-Abort.req	X	X	X		
TR-Abort.ind	X	X	X		

3. 协议数据单元 (PDU)
- 使用了以下的PDU：
- 调用PDU。
 - 确认PDU (Ack PDU)。
4. 过程
- 第1级事务由WTP用户通过将事务级别参数设置为 1的TR-Invoke请求原语来发起。WTP提供者发出调用消息，成为事务的发起者；远程的 WTP提供者接收调用消息，成为事务的响应者。响应者检查事务标识并决定是否发起验证，如果不，它将把消息送给用户并向发起者返回最后的确认。响应者必须保持状态信息以便在知道丢失时重新传送最后的确认。
- WTP提供者必须支持这个事务级别，它必须既能充当发起者又能充当响应者。

15.4.3 第2级事务

1. 动机
- 第2级事务是基本的请求/响应事务服务，它是用得最广泛的事务服务。例如， WSP[WSP]用它来进行方法调用。
2. 服务原语序列
- 表15-11描述了合法的服务原语序列。
- 列在第一栏中的服务原语后面仅能跟随着行中标记了“X”的原语。
- WTP用户通过发布 TR-Result请求原语来发送结果消息。当发起者收到结果消息，返回给响应者最后的确认。发起者必须保持状态信息以便在知道丢失时重新传送最后的确认。
- 如果响应者不支持这个事务级别，则返回带有放弃原因 NOTIMPLEMENTEDCL2的放弃

PDU作为对调用消息的响应。

表15-11 用于第2级事务的原语序列

	TR-INVOKE				TR-RESULT				TR-ABORT	
	REQ	IND	RES	CNF	REQ	IND	RES	CNF	REQ	IND
TR-Invoke.req										
TR-Invoke.ind										
TR-Invoke.res		X								
TR-Invoke.cnf	X									
TR-Result.req		X*	X							
TR-Result.ind	X*			X						
TR-Result.res						X				
TR-Result.cnf					X					
TR-Abort.req	X	X	X	X	X	X	X			
TR-Abort.ind	X	X	X	X	X	X	X			

* 代表如果使用用户确认则无效。

15.5 协议特征

15.5.1 消息传送

1. 描述

WTP包括两种类型的消息：数据消息和控制消息。数据消息携带用户数据。控制消息用来进行确认、错误报告等等，不携带用户数据。这一节给出了 WTP怎样实现事务的全面介绍，强调了保证可靠的消息传送的程序，像串联和分离、重传直到确认、事务放弃、用户确认及其他一些特殊的功能在单独的节中有进一步的详细描述。

重要的是注意到并非所有的消息和功能被所有的事务级别使用。表 15-12表示了不同的事务级别使用的消息。

注释1：仅仅当用户服务于调用消息的时间长于响应者确认时间的间隔时才被发出。

注释2：第0级事务是不可靠的，不期望任何来自响应者的响应，不执行验证。

表15-12 WTP消息传输的总结

消息/功能	第2级	第1级	第0级
调用消息	X	X	X (注释2)
验证	X	X	
坚持确认	X (注释1)		
结果消息	X		
最后的确认	X	X	

2. 服务原语

名义上的WTP事务使用下列服务原语,它们的使用依赖于事务级别：

- 传输 - 调用 (TR-Invoke)。
- 传输 - 结果 (TR-Result)。

3. 传输协议数据单元 (PDU)

在WTP事务中使用下述PDU，重要的是注意并非所有的PDU都用在每一个事务级别中。

- 调用PDU。
- 结果PDU (Result PDU)。
- 确认PDU。

4. 定时器间隔和计数器

WTP事务使用下述的定时器间隔和计数器，它们的使用依赖于事务的级别。

- 重传间隔。
- 重传计数器。
- 确认间隔。
- 等待超时间隔。

间隔与计数器值与它们之间的关系可能依赖于所使用的事务级别。

5. 过程

事务在两个WTP提供者之间发生。一个WTP用户通过发布TR-Invoke请求原语发起一个事务，原语的TCL参数指示了事务的级别：0、1或是2。在WTP中，发起者是发起事务的WTP提供者，响应者是响应这个已经发起的事务的WTP提供者。

(1) 调用消息

调用消息总是事务的第一条消息，它通过调用 PDU来发送。每发起一个事务，发起者就将事务标识符 (TID) 加1，以达到管理事务的目的，TID在事务的每一个PDU中传送。当调用PDU被发出，发起者开始重传计时并等待响应，当响应者接收到一个带有有效 TID的调用PDU时，它产生TR-Invoke指示原语，并向用户发送消息。

(2) 验证

当响应者收到并接受了调用消息时，它应该缓存其 TID，这是为了过滤掉较小的或相等的TID值 (参考15.5.8节) 的重复性消息及旧消息。如果响应者经判定得知调用 PDU中的TID是无效的，则它可验证这个调用消息是新的还是被延迟的消息，这通过发送一个 Ack PDU来完成，这个过程向发起者发起了一个三段握手过程。在这种情况下，响应者直到三段握手过程成功完成后才向用户发送数据，如果三段握手失败了，则发起者放弃该事务。

(3) 坚持确认

当确认消息被发送到WTP用户后，确认计时器开始计时。如果WTP用户为调用消息提供的服务时间比确认计时器设定的时间间隔长，则响应者可能或者应该或者必须发送一个“坚持”确认，这是为了避免发起者重传调用 PDU。当发起者收到确认 PDU时，它停止重传调用PDU，并产生TR-Invoke确定原语。

(4) 结果消息

在配置数据上，WTP用户通过发起TR-Result请求原语来发送一个结果消息，结果消息用结果PDU来重传。当结果PDU被发出，响应者开始重传计时并等待响应，在发起者收到结果PDU后，如果一个TR-Invoke确定原语已经被发布并且转寄 TR-Result指示原语，则产生 TR-Invoke确定原语。

(5) 最后的确认

当事务的最后一个消息被收到时，发出最后的确认 PDU，确认的发出者必须保持所需的状况信息以处理上一个消息的重传，这可以通过使用一个等待计时器来实现，或者是通过保

持指示上一个事务结果的事务历史记录来实现。

15.5.2 重传直到确认

1. 动机

重传直到确认的程序用来保证在分组丢失的情况下，WTP提供者之间的可靠的数据传输。为了使空中传送的分组数最小，WTP在任何可能的地方使用明确的确认。一个例子就是结果消息的使用，以便清楚地确认调用消息。

2. 传输协议数据单元（PDU）

使用了下述的PDU：

- 调用PDU。
- 结果PDU。
- 确认PDU。

3. 计时器间隔和计数器

使用了下列的计时器间隔和计数器：

- 重传间隔。
- 重传计数器。

计时器和计数器的值和它们之间的关系依赖于所用的事务级别。计时器和计数器的详细描述在15.7.4节。

4. 过程

当一个分组被发出，重传计时器开始计时，且重传计数器被置为0。如果当重传计时器到时而还没有收到响应，则重传计数器加1，分组被重传，并且重传计时器重起，WTP提供者坚持重传直到重传的次数超过了最大重传值。如果当重传计数器和计时器都到时之时没有收到任何确认，则事务被终止，通知本地WTP用户。

PDU第一次传输时，报头中的重传指示器（RID）字段是空的。在所有的重传中，RID字段被设置。除RID字段之外，WTP提供者不必改变PDU报头中的任何字段。

设置重传指示器的目的是方便接收者检测网络中重复的消息。接收到两个相同的RID设置为0消息的WTP提供者可以安全地忽略掉第二个消息，因为它一定是网络复制的消息，任何RID标志为1的后到的重传不能被接收者忽略。网络复制的重传消息必须被WTP提供者以有效消息对待，此时接收者不再能分辨是WTP提供者重传的消息还是网络复制的消息。在这种情况下，如果消息是个调用PDU，则事务有被重放的危险。为避免此错误的发生，WTP提供者应该执行TID有效性验证。

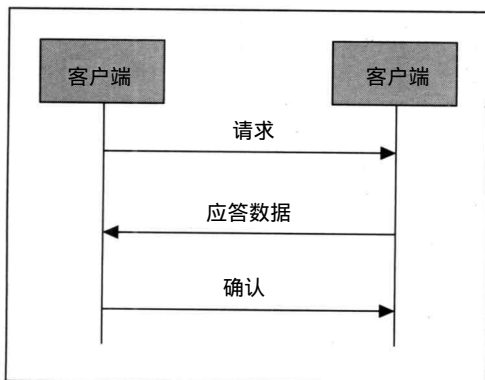
15.5.3 用户确认

1. 动机

用户确认功能考虑到了WTP用户需要确定每一个从WTP提供者接收到的消息情况。

当此功能可以使用时，在WTP用户证实了指示服务原语（通过发布响应原语）之后，WTP提供者才响应一个接收到的消息。如果WTP用户在一个特定的时间后不确定指示原语，则提供者放弃此事务。注意，这是一个比传统的定义[ISO8509]强得多的确定服务形式。确定服务的传统形式是来自服务提供者的确定，然而，它与来自对等的服务用户的响应无关。在

此功能是 WTP 中的可选项，然而，WSP 的确使用用户确认特征，因此任何以 WSP 为其高层的 WTP 的实现必须执行此功能。WSP 需要一种特征，即在请求-响应事务端，服务器得到一个关于客户已经处理了响应的确定的指示。此过程在图 15-1 中描述。



在这个模型中，确认用来传送客户端应用程序已经接收并处理了的响应，重要的是注意到在图中客户端和服务端指的是客户端和服务器的应用程序，而不仅仅是协议栈。

```

sequenceDiagram
    participant Client as 客户端
    participant WSP as WSP
    participant WTP as WTP
    participant Server as 服务器

    Client->>WSP: S-Method.req
    WSP->>WTP: TR-Invoke.req
    WTP->>Server: TR-Invoke.ind
    Server->>WTP: S-Method.ind
    WTP->>WSP: TR-Invoker.res
    WTP->>Server: S-Method.res
    Server->>WTP: S-Reply.req
    WTP->>WSP: TR-Result.ind
    WTP->>Server: TR-Result.req
    Server->>WTP: S-Reply.cnf
    WTP->>WSP: TR-Result.cnf
    WTP->>Server: S-Reply.cnf
    WSP->>Client: S-Method.cnf
    WSP->>Client: S-Reply.ind
    WSP->>Client: S-Reply.res
  
```

The diagram illustrates the interaction between a Client (客户端) and a Server (服务器) through a Web Service Proxy (WSP) and a Web Service Target (WTP). The sequence of messages is as follows:

- Client to WSP:** S-Method.req
- WSP to WTP:** TR-Invoke.req
- WTP to Server:** TR-Invoke.ind
- Server to WTP:** S-Method.ind
- WTP to WSP:** TR-Invoker.res
- WTP to Server:** S-Method.res
- Server to WTP:** S-Reply.req
- WTP to WSP:** TR-Result.ind
- WTP to Server:** TR-Result.req
- Server to WTP:** S-Reply.cnf
- WTP to WSP:** TR-Result.cnf
- WSP to Client:** S-Method.cnf
- WSP to Client:** S-Reply.ind
- WSP to Client:** S-Reply.res

在图15-1中，原语序列由S-Reply.res开始，TR-Result.res原语实现完成及确定原语。如果由于某种原因，应用程序和 /或WSP不发布这些原语，WTP以NORESPONSE原因放弃此事

务,WSP服务器使用放弃来指示结果没有被客户端正确地接收或处理。

原语序列以S-Method.res开始,TR-Invoke.res原语可由客户端WSP用来指示应用程序(及人)服务器WSP没有收到调用消息。

当使用这个功能时,WTP可能独立于WTP用户确认接收到的消息。在图15-2中这意味着响应原语可能被WTP提供者忽略,换句话说,WTP提供者接收一个消息,返回一个确认,并指示用户消息被收到。如果发生错误,WTP提供者将放弃事务。如果WTP用户是活动的但不能处理消息,它可能以一个适当的放弃原因放弃事务。

此功能是可选的。它应用于第1级和第2级事务。

注意,即使WTP用户已经发布了一个响应原语,仍然不能保证它已经解释了数据并开始处理。WTP用户可能仅仅把数据从一个缓冲区拷贝到另一个缓冲区,或者是发布了没有进行任何操作的响应原语。一个WTP用户如果发现接收到的数据是损坏的或因某种原因不能处理(参见15.5.7节),它总能放弃事务。

2. 协议数据单元(PDU)

使用了下述的PDU:

- 调用PDU。
- 放弃PDU。

3. 过程

发起者在调用PDU中设置U/P-flag以指示需要请求用户确认,不支持此功能的响应者以放弃原因NOTIMPLEMENTEDUACK放弃该事物,则发起者可能决定重新发起没有用户确认功能的事务。

当响应者接收到带有U/P-flag设置,它产生TR-Invoke指示并开始确认计时,为了给WTP用户时间来读取指示原语中的参数及发布TR-Invoke响应原语,计时器的值可能比提供者的确认计时器的值要高。在WTP用户发布TR-Invoke响应原语之前响应者不必返回一个响应。如果发起者根据没有确认来重传调用PDU,响应者必须悄悄地放弃PDU并重起确认计时器。当WTP用户发布TR-Invoke响应原语时,响应者被设为能够发送确认PDU。如果在一个特定的时间后TR-Invoke响应原语还没有发布,则提供者以原因NORESPONSE放弃事务。如果WTP用户发布TR-Result请求原语,则发送结果而不是确认。收到确认PDU的发起者产生确定原语,此原语指示远程的WTP用户已经发布了相应的响应原语。

对第2级事务来说,如果发起者已经指示要使用用户确认功能,那么这对整个事务都有效。这意味着当发起者收到结果并产生TR-Result指示原语时,在最后的确认发出之前必须等待一个来自于WTP用户的TR-Result响应原语。如果在一个特定的时间之后TR-Result响应原语还没有发布,则提供者以NORESPONSE放弃事务,当响应者接收到NORESPONSE放弃时,它产生TR-Abort指示原语,指示WTP用户事务失败。

15.5.4 最后确认中的信息

1. 动机

WTP用户仅仅被允许在事务的最后确认中附加上信息,此功能的是为了传输与事务相关的少量信息。例如,此信息可能是收集到的性能测量数据以评估用户感觉到的服务质量。

在第2级事务中,此功能可被发起者用来把某些信息传回响应者;在第1级事务中,此功能

可被响应者用来向发起者传送一些信息。

2. 服务原语

使用了下述的服务原语及参数：

- TR-Result.res(Class 2)
- TR-Invoke.res(Class 1)

3. 协议数据单元 (PDU)

使用了下述的协议数据单元：

- 确认PDU。

4. 过程

在第2级事务中，通过发布带 ExitInfo参数的TR-Result响应原语将信息附加在最后的确认中。

在第1级事务中，通过发布带 ExitInfo参数的TR-Invoke响应原语将信息附加在最后的确认中。

在确认PDU头的可变部分中返回信息作为传输信息项目 (TPI)来传送。

在第2级事务中，ExitInfo参数不必包括在 TR-Invoke响应原语中，并且 Info TPI不必包括在对确认调用PDU的确认的PDU中。

15.5.5 串联和分离

1. 动机

串联是一个过程，用来在一个承载网的服务数据单元 (SDU) 中传送多个 WTP 协议数据单元 (PDU)。当用到串联时，使用了一种从 WTP PDU到SDU的特殊映射。

分离是一个过程，用来从一个 SDU中抽取多个PDU。当PDU被分离时，它们被分派到各自的事务。

因为只需要更少的空中传输，串联和分离用来提供高的空中效率。

2. 过程

串联仅仅能够对有相同地址信息 (源端口和目的端口、源地址和目的设备地址) 的消息使用。

不同事务PDU的串联可发生在任何时候。例如，一个事务的最后确认可与下一个事务的调用消息串联，串联和分离在 WTP状态机之外执行。

串联的实现没有具体规定，仅仅规定了当多个分组串联时使用的结构，分组怎样缓存及串联是一个实现问题。

15.5.6 异步事务

• 动机

WTP提供者的实现应该能够在其接收到第一个事务的响应之间发起多重的事务，多事务应该被异步处理。例如，事务号为 1、2和3的事务的响应可能以 3、1和2的顺序到达发起者，响应者应该在自己一准备好后就送回结果，而与其他的事务无关。

在任何时刻未完成事务的最大数受事务标识符的最大数字限制。事务标识是 16比特的，但最高位用于指示消息的方向，因此未完成事务的最大数为 2^{15} ，实现环境也会限制其可以

同时处理的未完成事务的数目。

15.5.7 事务放弃

1. 动机

WTP用户可以通过发布 TR-Abort请求原语来放弃一个未完成的事务，用户放弃可被应用程序（例如，从用户输入）触发或者是得到一个否定的结果（例如，WTP用户由于一个错误不能产生结果）。

WTP提供者还可以根据协议错误（例如，拒绝收到的数据）或者是一个请求的功能未被正确执行来放弃一个未完成的事务。

此功能必须小心地使用。如果调用消息已经发出，响应消息可能正在通往客户端的路上，这时放弃仅仅是增加了网络的负担。

2. 传输协议数据单元（PDU）

使用了下述的数据单元：

- 放弃PDU（Abort PDU）

3. 服务原语

使用了下述的服务原语：

- TR-Abort

4. 过程

放弃过程有三种情况：

- 负责发送的WTP提供者还没有发出消息，提供者必须放弃其内存中的消息。
- 负责发送的WTP提供者已经向对等端发出了消息或者是正在处理消息的发送，提供者必须发送放弃PDU给远程的对等端以放弃所有与该事务相关的数据。
- 负责接收的提供者接收到了放弃PDU，它产生TR-Abort指示原语并放弃所有的事务数据。

当一个放弃PDU被发出，在放弃原因字段中指示了放弃的原因。有两种主要的放弃类型：用户放弃（USER）和提供者放弃（PROVIDER）。当WTP用户发布TR-Abort请求原语时发生用户放弃，当WTP提供者中有错误时发生提供者放弃。

15.5.8 事务标识符

1. 动机

一个事务被套接对（源地址、源端口、目的地址和目的端口）及事务标识符（TID）唯一地标识。每发起一个事务，发起者将TID加1，这意味着TID1、TID2和TID3可能传到服务器A；TID4、TID5和TID6可能传到服务器B；而TID7、TID8和TID9又可能传到服务器A。

TID的主要用处是标识属于同一个事务的消息。当一个消息被重传时，TID在重传的消息中重新使用。在接受了调用消息并进行了TID验证后响应者可能选择记住TID以避免重放事务。并且，对每一个事务，发起者将TID加1，响应者可使用这个信息从旧的及重复的调用消息中过滤出新的调用消息，一个新的调用总会有一个更高的TID值。

既然多个事务可在同一个套接连接中从任一个方向同时发起，则TID的高位用来表示事务的方向。在调用PDU中发起者设置高位为0，再往后，高位比特总是在其被加到响应分组之前

在收到的TID中转换。通过设置TID字段的高位比特，在发起者中为0而在响应者中为1，可保证发起者端被分配的TID不会与远程的实体发生冲突。

TID为16比特，但高位比特用于指示方向，这意味着TID的最大数是 2^{15} 。TID是一个无符号整数。

2. 响应者一端的过程
- (1) 变量
- 如果响应者对每一个不同的发起者缓存旧的TID值，则旧的TID值被叫作LastTID。在收到的调用消息中的TID被叫作RcvTID。
- (2) 当收到一个新的调用消息时的决策
- 当响应者收到一个调用消息时，它采取表15-13（取决于响应者是否缓存旧的TID值）中所列行动之一，拥有下层传输的特征，并且有TID测试的结果（在下一章中描述）。
- 注释1：这是一个事例，例如，若在WTP下有安全层并且可以去除重复。
- 注释2：这样做的效率不是很高，应该避免。

表15-13 当接收到新的调用消息时的决策

事 件	环 境	操 作
TID测试失败	下层传输服务可以保证无重复（注释1）	开始事务
TID测试成功	下层传输服务不能保证无重复（注释1）	调用TID验证 LastTID=RcvTID 开始事务
TIDnew标志设置		LastTID=0 调用TID验证
无缓存	响应者缓存每一个发起者的TID到网络的最大 的分组生命周期，且在此周期中没有重起及丢 失信息。如果调用不是新的，则响应者的缓存 中会有最后的TID 响应者不缓存TID（注释2）	为此发起者创建新 的记录 LastTID=RcvTID 开始事务 调用TID验证

- (3) TID测试
- 使TID有效的一种方法是使用窗口机制。响应者可能缓存每一个不同的发起者的最后的有效TID（LastTID），当响应者接收到一个新的调用消息时，它将调用消息中的这个TID（RcvTID）与缓存中的TID相比较。设W为窗口的尺寸，如果 $W=2^{14}$ ，则W表示当两个TID值的差别是 2^{14} 这个边界值，也就是说，TID最大值的一半。
- 表15-14和表15-15给出了TID测试的不同结果。如果测试成功，则保证了收到的调用消息是新的而不是旧消息的延迟。这些结果成立的前提是假定所有消息都有最大的分组生命周期，并且在MPL秒后保证了在网络中没有重复的消息（参见注释）。此外，还假定TID的增加不会快于 $2 \times \text{MPL}$ 中的 2^{14} 步伐。

表15-14 TID Test; RcvTID LastTID

RCVTID-LASTTID	RCVTID LASTTID
	TID TEST
0	失败
W	成功
>W	失败

表15-15 TID Test; RcvTID<LastTID

RCVTID-LASTTID		RCVTID<LASTTID
		TID TEST
<W		失败
W		成功

注意，对一些网络类型，平均的最大分组生命周期可能有很大的变化。例如，在一个像 GSM SMS 这样的存储-转发网络中，一个短消息在它发送到目的地之前可能在 SMS-C 中驻留一个非常长的时间，在某些情况下这可能破坏 TID 确认的正确性。

(4) 非顺序调用消息的接收

消息可能非顺序到达，这意味着即使发起者对每一个事务将 TID 增加 1，但具有低 TID 值的事务可能比高 TID 值的事务后到，这可能导致 TID 测试失败而开始 TID 验证。这不会违犯协议，然而，这会导致性能的降低，一个避免的方法是保持以前事务的 TID 值的队列。如果接收到的 TID 不在队列中，则它可直接被接受而不需通过 TID 验证，这个解决方案提高了性能但是需要响应者保持更多的信息。

3. 发起者一端的过程
- (1) TID 的管理
- 对于每一个事务，发起者负责把 TID 加 1，这不一定比 2*MPL 中的 2**14 步执行得快。
- (2) 破坏 TID 的单调特性
- 在几种情况下发起者可能产生非单调的 TID 值，也就是，下一个 TID 可能比上一个要小：
- 1) 发起者崩溃、重起并随机地找到一个比以前的 TID 值小的 TID 数。
 - 2) 囊括了有限空间中所有的 TID 值。例如，如果发起者发送一个事务给响应者 A，又发送 2**14 事务给响应者 B，最后返回给响应者 A，此时就会发生这种情况。为此在响应者 A 中发送者缓存的 TID 值将比现在的 TID 值要小。

这两种情况都不会违犯协议，然而，TID 验证将被调用并导致效率降低。

在 1) 中，如果响应者在 MPL 秒后丢弃所缓存的 TID 值，并且重起的时间比它长，则响应者将接受新的 TID 值而不经 TID 验证。我们假设了增加 TID 2**14 步将比 2*MPL 花费更多的时间，然而，如果响应者缓存 TID 值长于 MPL 秒，它将发起一个 TID 验证。

在 2) 中囊括状态仅仅在发起者缓存了上一个发送到每一个响应者的 TID 时才会被检查到。

在 1) 和 2)，都应该通过在调用 PDU 中设置 TIDnew 标志来避免 TID 验证机制的额外使用，这将使响应者对发起者的 TID 缓存无效。当发起者使用 TIDnew 标志时，它在 TID 验证完成前不必发起任何后来的事务，原因在于 TIDnew 可能被网络延迟。如果在此周期中具有更高的 TID 的事务被发起，当响应者已经在 TIDnew 分组中以低的 TID 更新其缓存时，重复的事务将不被接受。

15.5.9 事务标识符验证

1. 动机
- 事务标识符验证程序是一个三段的握手过程，在发起者（I）和响应者（R）之间的三段握手采用以下三个步骤：

- I R 这是TID。(调用PDU)
- I R 你携带这个TID的未完事务吗？(确认PDU)
- I R 有/没有。(确认PDU/放弃PDU)

TID验证程序需要保证：根据旧的重复分组，同一个调用消息不被接受且发送给 WTP用户多于一次。

在TID验证程序成功完成之前，调用消息不必发送给用户。

2. 协议数据单元 (PDU)

使用了下述的PDU：

- 调用PDU
- 确认PDU
- 放弃PDU

3. 过程

当响应者收到一个来自发起者的调用 PDU并决定用事务标识程序的规则来验证它时，使用下面的处理。

响应者发送一个确认PDU，此PDU带有Tve flag设置，指示它已经接收到一个带有此 TID的调用消息。当发起者接收到来自响应者的确认 PDU时，它检查是否有一个带有此 TID的未完成事务。在这种情况下，发起者发送回一个确认 PDU，此PDU带有TIDok标志设置，指示此TID有效。这就完成了三段握手过程。如果发起者没有相应的未完成事务，它必须通过发送一个带有放弃原因INVALIDTID的放弃PDU来放弃事务。

根据TID验证的结果，WTP应该采取不同的操作。这在表 15-16中列出。

TIDnew标志在调用消息中设置，并且被发起者用来校验响应者的缓存。

表15-16 根据TID验证结果采取的操作

TID验证的结果	条 件	操 作
有效的TID	TIDnew=正确	开始事务 LastTID=RevTID
	TIDnew=错误	开始事务 LastTID=LastTID
无效的TID		放弃事务

15.5.10 传输信息项

1. 动机

WTP PDU报头的可变部分可包含传输信息的选项 (TPI)，如果没有，则报头的可变部分必须是空的。TPI的使用考虑到了未来的协议扩展。

2. 过程

所有的TPI都有一个通用的结构：TPI实体、TPI长度以及TPI数据；长度可为0。表15-17列出了现在定义的TPI及它们在哪一部分被解释。

表15-17 WTP传输信息项目 (TPI)

传输信息项目	在部分中的描述	传输信息项目	在部分中的描述
错误	“ 传输信息选项 (TPI) ”；参见 15.5.10节	选项	“ 参数的传输 ”；参见 15.5.11节
信息	“ 在最后确认中的信息 ”	分组序列号	“ 分割和重组 ”；参见 15.5.14节

除了错误 TPI 以外所有的 TPI 都是可选的。错误 TPI 用来通知发送者是否收到了一个不支持的或是错误的 TPI，因此，WTP 提供者必须能够识别通用的 TPI 结构，并忽略不支持的 TPI 并报告发送者。

当 WTP 提供者接收到一个它不支持的 TPI 时，WTP 提供者返回带有指示“不知道的 TPI”的错误码的错误 TPI 以及不支持的 TPI 的标识；当 WTP 提供者接收到一个它支持的 TPI 但是不懂 TPI 的内容时，WTP 提供者返回带有指示“知道 TPI，不知道内容”的错误码的错误 TPI 以及 TPI 的标识，并且以内容的第一个八位组（octet）为变元。

15.5.11 参数传输

1. 动机

协议参数可以在 PDU 报头的可变部分中通过使用选项 TPI 在两个 WTP 提供者之间传输。

没有定义强制性参数。

分割及重组功能所用的可选参数在 15.5.14 节中列出。

2. 过程

WTP 提供者可能仅仅支持全部参数的一个子集，参数使用选项 TPI 在 PDU 头的可变部分中传输。选项 TPI 的第一个八位组标识参数，后面的八位组包含参数的值。不支持参数的 WTP 提供者忽略它并返回错误 TPI。

15.5.12 错误处理

1. 动机

在事务进行期间，检测到未恢复的错误时，此事务必须被放弃。目前还没有定义恢复机制。

2. 协议数据单元（PDU）

使用了下述的协议数据单元：

- 放弃 PDU。

3. 过程

在事务进行期间，WTP 提供者中发生了错误，此事务必须用一个适当的放弃原因来放弃且通知本地的 WTP 用户。

15.5.13 版本处理

1. 动机

如果 WTP 提供者接收到一个带有比它支持的版本号高的调用消息，它必须放弃此事务。

2. 协议数据单元（PDU）

使用了下述的 PDU 及参数：

- 调用 PDU。
- 放弃 PDU。

3. 过程

发起者在调用 PDU 的版本字段中指示其版本。

如果响应者不支持此版本，它必须返回一个放弃原因设置为 WTPVERSIONONE 的放弃

PDU。这指示 WTP 提供者支持 WTP 协议的版本 1。

15.5.14 分割和重组 (可选的)

1. 动机

如果消息的长度超过了现有承载者的 MTU, 则此消息可能被 WTP 分割成几个分组来发送。当一个消息以小分组的最大数目发送时, 这些分组可能以一群来发送及确认, 发送者可以依靠网络的特性通过改变分组群的大小来实行流量控制。

此功能是可选的。如果 WTP 中不实现 SAR, 此功能就由栈中的另一层提供。例如, 在 IS-136 中, SSAR 层处理 SAR; 在 IP 网络中, IP[RFC791] 处理 SAR; 在 GSM SMS/USSD 中, SAR 通过使用 SMS 串联来达到 [GSM0340]。实现 WTP SAR 的动机是选择性的重传程序, 如果发出大的消息, 可以提高协议的空中效率。

2. 分割的过程

为了简单起见, 在此仅仅描述了分割一个调用消息的过程 (除了 PDU 的名字外, 结果消息的分割是相同的)。

一个超过网络 MTU 的调用消息被分割成一个有顺序的序列, 此序列由一个调用 PDU 及随后的一个或多个分割的调用 PDU 组成。初始的调用 PDU 有绝对的分组序列号 0, 随后的被分割的调用 PDU 有分组序列号 1, 再后面的被分割的调用 PDU 的分组序列号总是比其前一个大 1 (n , $n+1$, $n+2$ 等等)。既然分组序列号不包括在报头的一个字段中, 则调用 PDU 的分组序列号置为绝对的。客户端在调用 PDU 中通过 TTR 标志是否为 0 指示调用消息是否被分割, 如果调用消息被分割, 则服务器从分组号 0 开始对调用 PDU 计数并且等待后面的被分割的调用 PDU。分组序列号不能相同, 分组序列号字段为 8 比特, 因而分组的最大数目为 256。

3. 分组群的过程

分组 (被分割的调用 PDU 和/或被分割的结果 PDU) 以群来发送及确认。在上一个分组群被确认之前, 发送者不能发送任何属于同一个事务的新的分组, 也就是说, 分组群根据停止 - 等待协议来发送。发送者决定每一个分组群的分组数目, 决定分组群的大小时应该考虑到网络 and 设备的特性。没有定义任何决定分组群大小的过程。

一个分组群中的分组在一批中发送, 群中的上一个分组有 GTR 标志设置, 完整消息的最后分组群的最后分组有 TTR 标志设置。既然第一个群在不知道接收者状态时发送, 则分组的数目不应该太大。当接收者接收到一个不是 GTR 或 TTR 的分组时, 它必须存储此分组并等待一个新的分组。

当接收者收到一个有 GTR 标志设置的分组时, 它必须检查是否收到了属于这个分组群的所有分组。如果已经收到了所有的分组群, 则接收者返回一个带有 PSN TPI 的确认 PDU, 此 PSN TPI 包含 GTR 分组序列号。如果丢失了一个或多个分组, 则接收者返回一个包含行动的分组序列号的 Nack PDU。丢失的分组以原来的分组序列号重传, 但重传指示标志被设置。当接收者已经收到的完全的分组群 (包含重传的) 时, 它确认 GTR 分组。

当接收者已经收到了完全的分组群且最后的分组有 TTR 标志设置时, 它应该能够重组消息。

如果发送者在重传计时器满时还没有收到确认, 则仅仅 GTR/TTR 分组被重传, 而不是整个的分组群。

4. 可选择重传的过程

当已经收到一个 GTR或TTR分组并且丢失了一个或多个分组群中的分组时， WTP提供者返回带丢失分组序列号的 Nack PDU。例如，如果接收者已经收到了分组号 2、3和7，并且分组号7有GTR标志设置，则返回带分组号 4和6的Nack PDU以指示丢失的分组。丢失分组的序列号包含在Nack PDU的头中。

如果Nack PDU被接收时丢失分组字段被设置为 0，这意味着要重传全部的分组群。

丢失的分组以原来的分组序列号被重传。当发送者已经重传所需的分组时，它又回去等待原来的确认（对GTR或者TTR分组）。

当接收者已经收到所有的分组时，应根据通常的过程使用确认 PDU来确认 GTR或者TTR 分组。

当一个或者多个分组被请求重传时，一个不支持此功能的 WTP提供者必须重传整个的消息。

15.6 协议数据单元的结构和编码

15.6.1 概述

一个PDU包含一个整数数目的八位组，并有：

- 报头 包括：固定部分，可变部分。
- 数据 如果存在。

报头的固定部分包含经常使用的参数及 PDU码，固定部分的长度和结构由 PDU码定义。在表15-18中列出了当前定义的PDU类型。

表15-18 WTP PDU类型

PDU 类 型	PDU 编 码
* 不允许 *	0x00 (注释1)
调用	0x01
结果	0x02
确认	0x03
放弃	0x04
分割的调用	0x05 (注释2)
分割的结果	0x06 (注释2)
否定的确认	0x07 (注释2)

可变部分用来定义不经常使用的参数。可变参数在传输信息选目（ TPI ）中携带。

固定头的第一个比特指示 PDU是否有可变头,固定头的长度由PDU类型给出，可变头包括TPI。每一个TPI有一个针对自身长度的长度字段,它的第一个比特指示它是否是最后的 TPI。

PDU的网络字节顺序是低字节高置的存取次序，换句话说，最重要的字节在网络上首先传输，随后传输次重要的字节。

注释1 如果数据报的第一个字节为 0X00，将解释为数据报包含多个串联的 PDU。参见 15.6.5节。

注释2 此PDU仅在执行了可选的分割和重组功能时有效。

15.6.2 通用的报头字段

1. 连续标志位(CON)
作为报头固定部分的第一个比特，连续标志位指出在可变部分中是否存在 TPI。如果这个标志被设置，则在报头的可变部分中有一个或多个 TPI；如果这个标志是空的，则报头的可变部分是空的。
此标志还被当作一个TPI的第一个比特，表示这个TPI是否为可变头的最后一个TPI。如果标志被设置，其他的TPI跟着这个TPI；如果这个标志为空，则这个TPI之后的八位组是用户数据的第一个字节。
2. 群跟踪（GTR）和传输跟踪（TTR）标志
当进行分割和重组时，TTR标志用来表示被分割消息的最后一个分组，而GTR标志用来表示分组群的最后一个分组（参见表15-19）。
默认设置应该是GTR=1且TTR=1，也就是说，不支持WTP分割和重组。

表15-19 GTR/TTR标志组合

GTR	TTR	描 述
0	0	不是最后一个分组
0	1	消息的最后一个分组
1	0	分组群的最后一个分组
1	1	不支持分割和重组

3. 分组序列号
它被属于分割和重组功能的PDU使用。这个号码表示这个分组在被分割消息中的位置。
4. PDU类型
PDU类型字段表示这个PDU是什么WTP PDU类型（调用、确认等等），它为接收端WTP提供者提供信息以说明PDU数据应该被怎样解释以及请求什么样的操作。
5. 恢复(RES)
除非特别规定，所有被恢复的比特被设置为值 0x00。
6. 重传指示(RID)
使接收者能够区分网络复制的分组及被发送者重传的分组。在原来的消息中，RID是空的。当消息被重传，RID被设置。
7. 事务标识(TID)
TID将一个分组与一个特定的事务相关联。

15.6.3 固定头结构

1. 调用PDU（表15-20）

表15-20 调用PDU的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=调用				GTR	TTR	RID
2				TID				
3								
4	版本		TIDnew	U/P	RES	RES		TCL

(1) 事务级别(TCL)

发起者在调用消息中指出期望的事务级别 (参见表 15-21)。

表15-21 类字段的编码

类 型	TCL	类 型	TCL
0	0x00	2	0x10
1	0x01		

(2) TIDnew 标志

当发起者已经囊括了所有的 TID 值时设置此标志,也就是说,下一个 TID 会比上一个小。当响应者接收到调用 PDU 并且 TIDnew 标志被设置时,它使发起者缓存的 TID 值无效。

(3) 版本

当前的版本为 0x00。

(4) U/P 标志

当这个标志被设置时,表明发起者请求一个来自服务器 WTP 用户的用户确认 ,这意味着 WTP 用户要确定每一条收到的消息。

当这个标志为空时,WTP 提供者可以在没有 WTP 用户确定的情况下响应一个消息。

2. 结果PDU (表15-22)

表15-22 结果PDU的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=结果					GTR	TTR RID
2				TID				
3								

3. 确认PDU (表15-23)

表15-23 确认PDU的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=确认				Tve/Tok	RES	RID
2				TID				
3								

• Tve/Tok标志

在响应者到发起者的方向上,Tve 标志 (TID 验证) 的含义是:“ 你携带有这个 TID 的未完成事务吗? ” 在相反的方向,Tok (TID OK) 标志的含义是:“ 我携带有这个 TID 的未完成事务! ”

4. 放弃PDU (表15-24)

表15-24 放弃PDU的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=放弃					放弃类型	
2				TID				
3								
4				放弃原因				

• 放弃类型及放弃原因

表15-25列出了目前规定的放弃类型。

表15-25 WTP放弃类型

放 弃 类 型	编 码	描 述
提供者 (PROVIDER)	0x00	放弃由WTP提供者自身产生。在下一部分中给出放弃原因
用户 (USER)	0x01	放弃由WTP用户产生。放弃原因由WTP用户提供给WTP提供者

1) 来自WTP提供者的放弃原因

表15-26给出了放弃原因。

表15-26 WTP提供者放弃码 .

放弃原因 (提供者)	编 码	描 述
不知道 (UNKNOWN)	0x00	通用的错误编码，表示意外的错误
协议错误 (PROTOERR)	0x01	不能解释接收到的PDU。可能有结构错误
无效的TID(INVALIDTID)	0x02	仅仅由发起者作为TID验证的否定结果使用
不执行第2级别 (NOTIMPLEMENTEDCL2)	0x03	因为响应者不支持级别2的事务，此事务不能完成
不执行SAR (NOTIMPLEMENTEDSAR)	0x04	因为响应者不支持SAR，此事务不能完成
不执行用户确认 (NOTIMPLEMENTEDUACK)	0x05	因为响应者不支持用户确认，此事务不能完成
WTP版本1 (WTPVERSIONONE)	0x06	当前的版本为1。发起者请求了一个不支持的另外的版本
暂时超过容量 (CAPTEMPEXCEEDED)	0x07	由于负荷情况，事务不能完

2) 来自WTP用户的放弃原因

来自用户的放弃原因在T-TRAbort请求原语中被提交给本地的WTP提供者给出，放弃原因因用户不同而不同。例如，如果WTP用户是WSP，在[WSP]中定义的放弃码就可使用。

5. 分割调用PDU (可选的，见表15-27)

表15-27 分割调用PDU

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=分割调用PDU				GTR	TTR	RID
2				TID				
3								
4				分组序列号				

6. 分割的结果PDU (可选的，见表15-28)

表15-28 分割的结果PDU

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=分割的结果			PDU	GTR	TTR	RID
2					TID			
3								
4				分组序列号				

7. PDU的否定确认（可选的，见表 15-29）

表15-29 PDU 的否定确认

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON	PDU类型=否定确认		PDU		保留	RID	
2				TID				
3								
4				分组序列号=N				
5				丢失分组的分组序列号				
...								
4+N								

(1) 丢失分组的数目

指出丢失分组的数目。如果是 0f00，整个分组群都被重传。

(2) 丢失分组的分组序列号

一系列的丢失分组的分组序列号。

15.6.4 传输信息项

1. 概述

PDU的可变部分可以包括一个或多个传输信息选项，即 TPI。TPI的长度可为2比特或8比特。

长的TPI（长度为8比特）结构如表 15-30所示。

表15-30 长TPI的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON		TPI标识			1	RES	RES
2				TPI长度=N				
3				TPI数据				
...								
2+N								

短的TPI（长度为2比特）结构如表 15-31所示。

表15-31 短TPI的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON		TPI标识			0	TPI Length=M	
2				TPI数据				
...								
1+M								

在上述的表中，N=0.255而M=0.3。TPI的数据字段必须包含一个整数数目的八位组，在理论上TPI的最大长度为255的八位组，然而，它还要受到承载网络的 MTU大小及同一个PDU头中其他TPI的数目和长度的限制。

表15-32列出了当前0定义的TPI。

表15-32 TPI的编码

TPI	TPI标识	注 释
错误	0x00	
信息	0x01	
选择	0x02	
分组序列号 (PSN)	0x03	注释1

注释：只有当可选的分割和重组功能被实现时，这个 TPI才能应用。

2. 错误TPI
如果使用了错误的或不支持的 TPI，发送者会接受到错误 TPI。表15-33列出了当前定义的
错误编码。

表15-33 错误TPI的编码

错 误	编 码	变 元
不知道TPI	0x01	不知道TPI的TPI标识
知道TPI，但不知道内容	0x02	内容的第一个字节的TPI标识

根据错误代码，错误 TPI有不同的结构（参见表 15-34和表15-35 ）。
注意此TPI由一个WTP提供者授权。随后，此 WTP提供者必须还能识别TPI的一般结构。

表15-34 错误TPI (UNKNOWN) 的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON					0		TPI长度=0x01
2								TPI标识

表15-35 错误TPI (KNOWN) 的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON					0		TPI长度=0x02
2								TPI标识
3								TPI的第一个字节

3. 信息TPI
这个TPI用于在PDU报头的可变部分传输少量的数据，这些数据可能是性能测量或分析数据。
表15-36中描述了信息TPI的结构。

表15-36将信息TPI以短TPI来给出，如果发送更多信息，则可使用长 TPI。

表15-36 信息TPI的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON					0		TPI长度=N
2								信息
...								
1+N								

4. 选项TPI

选项TPI用来在两个WTP实体之间传输参数，在选项 TPI中携带的参数在事务的整个生命周期中有效。表15-37列出了当前定义的选项。

表15-37 选项TPI的编码

选 项	标 识	描 述	注 释
最大接收单元	0x01	发起者用此参数来通告在结果中能接收的以字节计算的最大的数据单元	注释1
整个消息大小	0x02	在分割了的消息的第一个分组中发送此参数以通知接收者以字节计算的全部消息	注释1
延迟传输时间	0x03	当一个分组群被确认时，在确认 PDU中发送此参数，在到达规定的时间之前接收者不能发送下一个分组。这个时间是 1/10秒	注释1

表15-38描述了选项 TPI的结构。

表15-38 选项TPI的结构

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON		TPI标识		0	TPI长度=N		
2				选项标识				
3				选项值				
...								
1+N								

5. 分组序列号 TPI（可选的）

确认PDU没有分组序列号（PSN）字段（见表15-39）。当使用分割和重组时，此TPI附加在确认PDU报头的可变部分上。确认PDU所包含的PSN是确认的分组（GTR或者TTR分组）的PSN。

表15-39 分组序列TPI

BIT/OCTET	7	6	5	4	3	2	1	0
1	CON		TPI标识=PSN TPI			0	长度=0x01	
2	分组序列号							

15.6.5 串联的PDU的结构

一个数据报服务数据单元（SDU）中可能包含一个或多个WTP协议数据单元（PDU）。如下图所示：

数据报服务数据单元

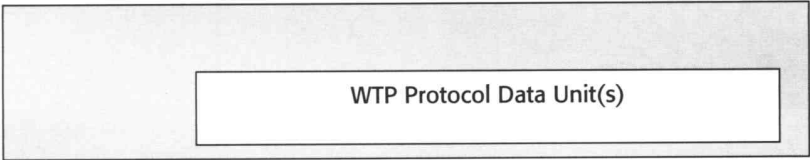


表15-40表示了有一个WTP PDU数据报的SDU，这个PDU包括报头及N个八位组的数据。

表15-40 无串联的WTP PDU

BIT/OCTET	7	6	5	4	3	2	1	0
1	WTP PDU							
...								
N								

表15-41表达了两个WTP PDU串联在承载网的同一个SDU中，第一个PDU有N八位组而第二个PDU有M八位组。

表15-41 串联的WTP PDU

BIT/OCTET	7	6	5	4	3	2	1	0
1				串联指示器=0x00				
2	0			WTP PDU 长度=N				
3				WTP PDU				
...								
N+2								
N+3	0			WTP PDU长度=M				
N+4				WTP PDU				
...								
N+M+3								

串联指示器用来指示SDU包含了多个WTP PDU。PDU的数目仅仅受到SDU最大长度的限制。

PDU的长度可以是7比特或者15比特。如果PDU长度字段中的第一个比特被设置，则长度为15比特；如果没有，则长度为7比特。这意味着根据第一个比特是否被设置，PDU的长度字段为8比特或是16比特。在表15-41中，第一个位为0，则长度为7比特。

15.7 状态表

15.7.1 概述

本章定义了没有可选分割和重组功能的核心 WTP协议的状态表。

15.7.2 事件处理

发起一个事务的 WTP提供者被称为发起者，对一个已经被发起的事务作出响应的 WTP提供者被称作响应者，一个 WTP事务的实现并不总是需要发起者和响应者都存在。例如，如果 WTP用户是 WSP客户端协议，则 WTP提供者可能仅仅支持事务的发起，也就是说，没有响应者。要使实现的标准具有一致性，请参阅 WTP一致性原因。

对上一层的接口由 WTP服务原语定义。典型的下一层是一个数据报服务，唯一需要的服务原语是单元数据的指示和请求原语，来自上一层的请求和响应服务原语以及来自下一层的指示原语在术语上被称为事件。如果多个 PDU在下一层的SDU中串联，则这些PDU必须被分割并重新分派给各自的事务。除了外部的事件外，还存在内部的事件，例如计时器到时及错误。

在被处理之前，一个事件是有效的。表 15-42列出一些测试，如果不采取其他的操作，这个事件的处理过程根据状态表完成。

表15-42 到来的事件的测试

测 试	操 作
在响应者端的 UnitData.ind：调用PDU	创建一个新事务
在发起者端的 UnitData.ind：带有TIDvc标志设置的确认PDU， 没有匹配的未完成事务	发送放弃PDU（INVALIDTID）
UnitData.ind：确认PDU，结果PDU或者放弃PDU，没有匹配 的未完成事务	忽略
不合法的PDU类型或者错误的头结构	发送放弃PDU（PROTOERR）
缓冲区溢出或者超出内存错误	发送放弃PDU（CAPTEMPEXCEED）
在响应者端的 UnitData.ind：调用PDU需要第2级事务而它不 支持第2级事务	发送放弃PDU（NOTIMPLEMENTEDCL2）
在响应者端的 UnitData.ind：调用PDU使用了SAR而它不支持 SAR	发送放弃PDU（NOTIMPLEMENTEDSAR）
响应者端的 UnitData.ind：调用PDU需要用户确认而它不支持 用户确认	发送放弃PDU（NOTIMPLEMENTEDUACK）
响应者端的 UnitData.ind：用版本！=0x00调用PDU	发送放弃PDU（WTPVERSIONONE）

15.7.3 操作

1. 计时器
- 在状态表中可使用下述的计时器操作：
- 开始计时 以一个给定的间隔值开始计时。如果计时器已经在运行，则以一个新的值重起。
 - 停止计时 停止计时，且不产生新的事件。
2. 计数器
- 在状态表中可使用下述的计数器操作：
- 重置计数器 将计数器设为0。
 - 增1计数 将计数器加1。
3. 消息
- 在状态表中可以使用下述的消息操作：
- 排队（时间T） 将一个PDU排队，使它排在队列中直到最后被发送。消息排队的时间不能长于T个时间单元。
 - 发送 发送一个PDU，使它及队列中的任何PDU立即发送。
- 排队机制可用于串联来自不同事务的消息，这可看作是在事务状态机下面操作的串联层。串联层与具体的实现有关，本规范没有定义。

15.7.4 计时器、计数器和变量

1. 计时器
- 表15-43中列出的是 WTP使用的计时器。

表15-43 WTP计时器

计 时 器	描 述
事务计时器	每一个事务都有一个与之相关的计时器。计时器用来给出重试间隔、确认间隔以及等待超时间隔

根据事务类型以及事务的当前状态，计时器可由不同的值开始。根据计时目的，计时器值被分组。如表 15-44所显示。

表15-44 WTP计时器间隔

计时器间隔（名字）	描 述
确认间隔（A）	它设置发送一个确认前等待时间的限度
重试间隔（R）	它设置重传一个PDU前等待时间的限度
等待超时间隔（W）	它设置在释放关于一个事务的状态信息前等待时间的限度

重试间隔可能作为一个序列来实现，而重传计数器就是索引 R[RCR]。一种指数备份算法可以通过以指数增加的值组装 R[]来实现。

计时器间隔的值依赖于下列参数：

- 承载网络的特性。
- 事务级别。
- 事务的状态（哪一个消息在重传或已经被确认了）。

不同承载网络的计时器间隔值可在本章的 15.13节中找到。

2. 计数器

WTP使用表 15-45中列出的计数器。

表15-45 WTP计数器

计数器（名字）	描 述
重传计数器（RCR）	它设置PDU重传的最大次数的范围。最大值定义为 RCR_MAX
确认终止计数器（AEC）	它设置事务计时器的最大数的范围，它被初始化为确认的间隔，并且允许终止及在事务被放弃之前重起。最大值定义为 AEC_MAX

3. 变量

WTP在发起者端和响应者端使用表 15-46中列出的变量。

Uint16类型是一种无符号的16比特整数。BOOL类型是布尔值，只能有“真”或“假”两个值。

表15-46 WTP变量

变 量	类 型	描 述	WTP 变 量
GenTID	Uint16	此TID对下一个事务使用。每当发起一个事务时加1	全局的 仅限于发起者
SendTID	Uint16	事务中此TID值在PDU中发送	每事务一个
RcvTID	Uint16	事务中此TID值在每个PDU中都应该被接收。RcvTID=SendTID XOR 0x8000	每事务一个
LastOn	Uint16	接收自某个远程主机的最后一个 TID	每个远程主机一个 仅限于响应者
HoldOn	BOOL	当接收到HoldOn确认时为真	每个2级事务一个
Uack	BOOL	在此事务请求了用户确认时为真	每事务一个

15.7.5 WTP发起者

WTP 发起者的具体描述参见表 15-17 ~ 表15-50。

表15-47 WTP发起者为空

事 件	环 境	操 作	下一个状态
TR-Invoke.req	Class = 2 1	SendTID=GenTID 发送调用 PDU 重置RCR 开始计时, R[RCR] Uack=False	结果等待
	Class = 2 1 UserAck	SendTID=GenTID 发送调用PDU 重置RCR 开始计时, R[RCR] Uack=True	
	Class = 0	SendTID=GenTID 发送调用PDU	空

表15-48 WTP发起者结果等待

事 件	环 境	操 作	下一个状态
TR-Abort.req		放弃事务 发出放弃 PDU (用户)	空
RcvAck	级别 = 2	停止计时 Generate T-TRInvoke.cnf HoldOn = True	结果等待
	级别 = 1	停止计时 产生T-TRInvoke.cnf	空
	TIDve 级别 = 2 1	发送确认(TIDok) 增加RCR 开始计时, R [RCR]	结果等待
RcvAbort		放弃事务 产生TR-Abort.ind	空
TimerTO_R	RCR < MAX_RCR	增加RCR 开始计时, R [RCR]	结果等待
	RCR = MAX_RCR	发送调用 PDU 放弃事务 产生TR-Abort.ind	空
RcvResult	级别 = 2 HoldOn = True	停止计时 产生TR-Result.ind 开始计时, A	结果响应等待
	级别 = 2 HoldOn = False	停止计时 产生TR-Invoke.cnf 产生TR-Result.ind 开始计时, A	

表15-49 WTP发起者结果响应等待

事 件	环 境	操 作	下一个状态
TR-Result.res	ExitInfo	排队(A) Ack PDU	等待超时
		开始计时, W	
		排队(A) Ack PDU	
		带信息的 TPI	
RcvAbort	AEC < AEC_MAX	开始计时, W	空
		放弃事务	
		产生 T-TRAbort.ind	
		产生 T-TRAbort.req	
RcvResult	AEC = AEC_MAX	发送放弃 PDU (用户)	结果响应等待
		忽略	
		增加 AEC	
		开始计时, A	
TimerTO_A	Uack = False	放弃事务	等待
		发送放弃 PDU	
		(NORESPONSE)	
		排队(A) Ack PDU	
		开始计时, W	等待超时

表15-50 WTP发起者等待超时

事 件	环 境	操 作	下一个状态
RcvResult		发送 Ack PDU	等待超时
RcvAbort		放弃事务	
		产生 T-TRAbort.ind	
		产生 T-TRAbort.req	
TimerTO_W		放弃事务	空
TR-Abort.req		发送放弃 PDU (用户)	

15.7.6 WTP响应者

WTP 响应者的描述参见表 15-51 ~ 表15-56。

表15-51 WTP听到的响应

事 件	环 境	操 作	下一个状态
RcvInvoke	级别 = 2 1	产生 TR-Invoke.ind	调用响应等待
	有效的 TID	开始计时, A	
	U/P 标志	Uack = True	
	级别 = 2 1	产生 TR-Invoke.ind	
	有效的 TID	开始计时, A	听到
		Uack = False	
	级别 = 0	产生 TR-Invoke.ind	
	发送 Ack(TIDve)	TIDOK 等待	
级别 = 2 1	无效的 TID		

表15-52 WTP响应TIDOK等待

事 件	环 境	操 作	下一个状态
RcvAck	级别 = 2 1 TIDok	产生TR-Invoke.ind 开始计时, A	调用响应等待
RcvAbort		放弃事务	听到

表15-53 WTP响应者调用响应等待

事 件	环 境	操 作	下一个状态
TR-Invoke.res	级别 = 1	带InfoTPI的 排队(A) Ack PDU	等待超时
	ExitInfo 级别 = 1	开始计时, W 排队(A) Ack PDU	
		开始计时, W	
	级别 = 2	开始计时, A	结果等待
TR-Abort.req		放弃事务	听到
		发送放弃 PDU (用户)	
RcvAbort		产生 TR-Abort.ind 放弃事务	听到
RcvInvoke		忽略	调用响应等待
TimerTO_A	AEC < AEC_MAX	增加 AEC 开始计时, A	调用响应等待
	AEC = AEC_MAX	放弃事务 发送放弃 PDU (NORESPONSE)	听到
	级别 = 1	排队(A) Ack PDU	等待超时
	Uack = False	开始计时, W	
	级别 = 2	发送Ack PDU	结果等待
	Uack = False	停止计时	

表15-54 WTP响应者结果等待

事 件	环 境	操 作	下一个状态
TR-Result.req		重置 RCR 开始计时, R[RCR]	结果响应等待
		发送结果 PDU	
TR-Abort.req		放弃事务	听到
		发送放弃 PDU (用户)	
RcvAbort		产生 T-TRAbort.ind 放弃事务	听到
TimerTO_A		发送 Ack PDU 停止计时	结果等待

表15-55 WTP响应者结果响应等待

事 件	环 境	操 作	下一个状态
TR-Abort.req		放弃事务	听到
RcvAbort		发送放弃 PDU (用户) 产生 T-TRAbort.ind	听到
RcvAck		放弃事务	听到
TimerTO_R	RCR < MAX_RCR	产生 TR-Result.cnf 增加 RCR	结果响应等待
		发送结果PDU	
	RCR == MAX_RCR	开始计时, R [RCR] 产生 T-TRAbort.ind	听到
		放弃事务	

表15-56 WTP响应者等待超时

事 件	环 境	操 作	下一个状态
RcvInvoke		发送Ack PDU	等待超时
RcvAbort		放弃事务	听到
		产生 T-TRAbort.ind	
TimerTO_W			
TR-Abort.req		放弃事务	
		发送放弃 PDU (用户)	

15.8 协议操作的例子

15.8.1 简介

这一节的例子试图描述并解释清楚协议是怎么进行操作的。为了简单起见，在图表中仅仅包括了与给定例子有关的报头的字段，PDU报头的标志字段中的每一个标志用一个字符来表示。表15-57给出了可出现在例子中的不同字符。

像放弃原因和错误编码等参数用清晰的文本形式写出，TPI也一样。对于事务标识符，N*是高位顺序的N设置，如果N=0x0000，则N*=0x8000。

表15-57 例子中使用的缩写

缩 写	意 思
N	TIDnew标志被设置
V	TIDvc标志被设置
O	TIDok标志被设置
U	U/P标志被设置
G	GTR标志被设置
T	TTR标志被设置
TG	GTR和TTR标志都被设置，以指示不支持 SAR
RID=X	重传指示器为 X
TID=N	事务标识为 N
c0	TCL字段指示第0级事务
c1	TCL字段指示第1级事务
c2	TCL字段指示第2级事务

15.8.2 第0级事务

基本事务（图 15-3）

发起者发起一个第0级事务（c0）。

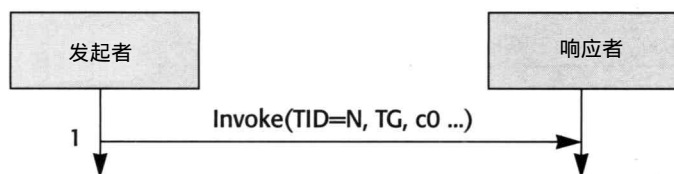


图15-3 基本的第0级事务

15.8.3 第1级事务

基本事务（图 15-4）

1) 发起者发起一个第1级事务（c1）。

2) 响应者确认所收到的调用消息。

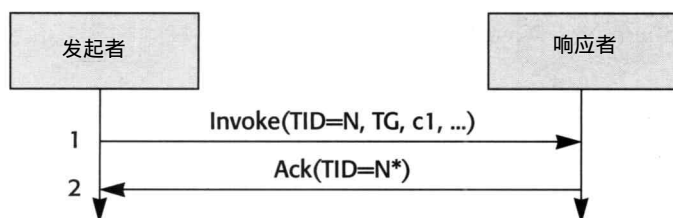


图15-4 基本的第1级事务

15.8.4 第2级事务

1. 基本事务（图 15-5）

1) 发起者发起一个第2级事务（c2）。

2) 响应者等待被处理调用消息，并以结果隐含地确认调用消息。

3) 发起者确认收到的结果消息。

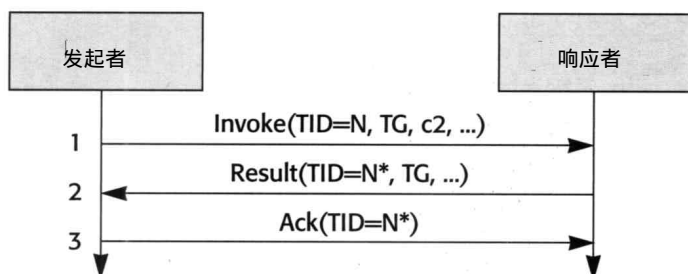


图15-5 基本的第2级事务

2. 带“坚持”确认的事务（图 15-6）

- 1) 发起者发起一个第2级事务 (c2)。
- 2) 响应者等待被处理调用消息。响应者端的确认计时器到时，发送一个“坚持”确认以避免发起者重传调用消息。
- 3) 发送结果到发起者。
- 4) 发起者确认所收到的结果消息。

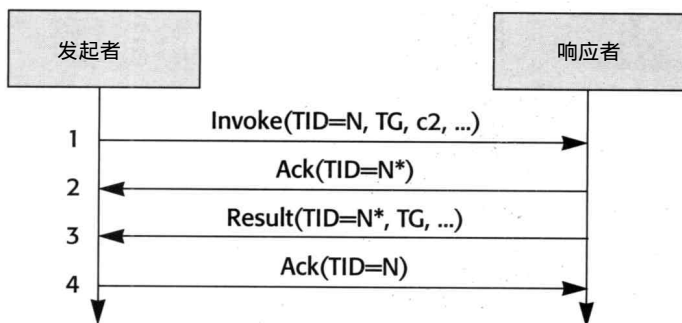


图15-6 带有“坚持”确认的第2级事务

15.8.5 事务标识符验证

1. 验证成功 (图15-7)

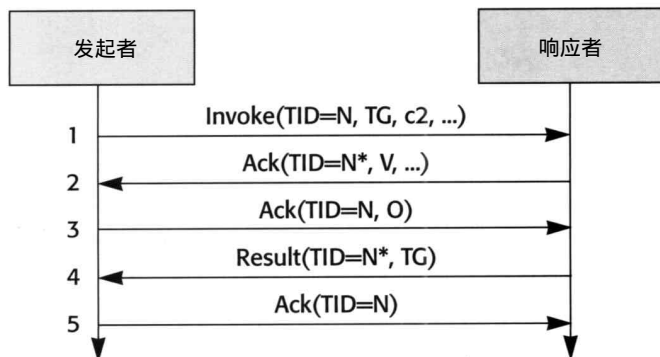


图15-7 验证成功

响应者接收到一个新的调用信息并且 TID 测试失败，这使得验证过程被调用。响应者向发起者返回一个确认信息，以确认它是否有具有这一 TID 的重要事务。在这个例子中，发起者有一个具有这一 TID 的重要事务并且确认了验证。

2. 验证失败 (图15-8)

具有 TID=N 的调用消息在网络中被复制，或者已经被延迟，当它到达时，事务 N 已经被终止了，响应者向发起者询问以验证这一事务，发起者通过送出一个 Abort (退出) 来退出事务。

3. 具有颠倒次序的调用消息的事务 (图15-9)

一个调用消息在网络中被延迟，当消息最终到达响应者时，响应者已经缓冲了一个更大

的TID值，响应者发起一个验证以检查发起者是否仍然保留着 TID=N的调用消息。

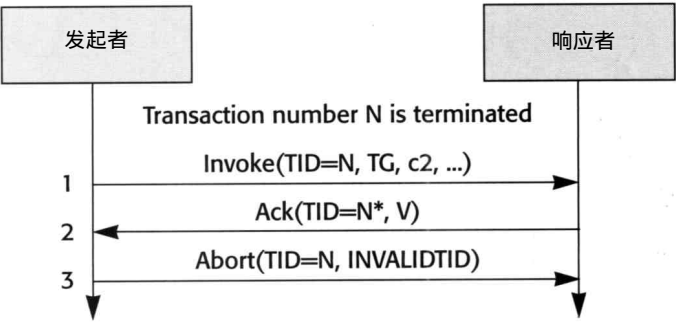


图15-8 验证失败

注意响应者不能把它已经缓存的 TID 值 (N+3) 换为更小的 TID 值。如果缓存的 TID 被向后移动，将使具有更高 TID 的旧副本将被错误地接收。

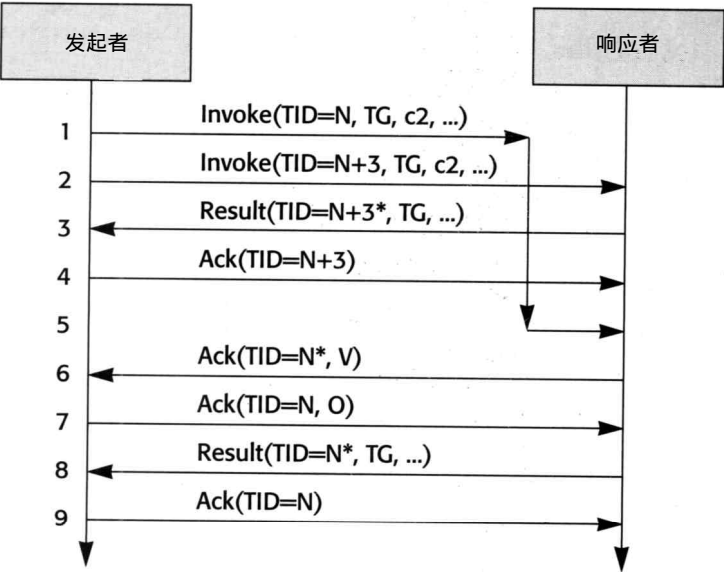


图15-9 被延迟的调用消息

15.8.6 分割和重组

图15-10给出了一个级别2事务所使用的分割。调用消息被分割，并在两个组中以五个包发送。

发起者首先在一个分支中发送最开始的 3 个分组，最后一个分组带有 GTR 标志，用来触发从响应者发来的确认，发起者一旦收到确认，就发出最后的两个包，最后的消息具有 TTR 标志集。过一段时间后，响应者把结果发回，发起者确认结果，然后这个事务结束。

注意，PSN TPI在ACK PDU中是作为包序列号 (Packet Sequence Number) 使用的。

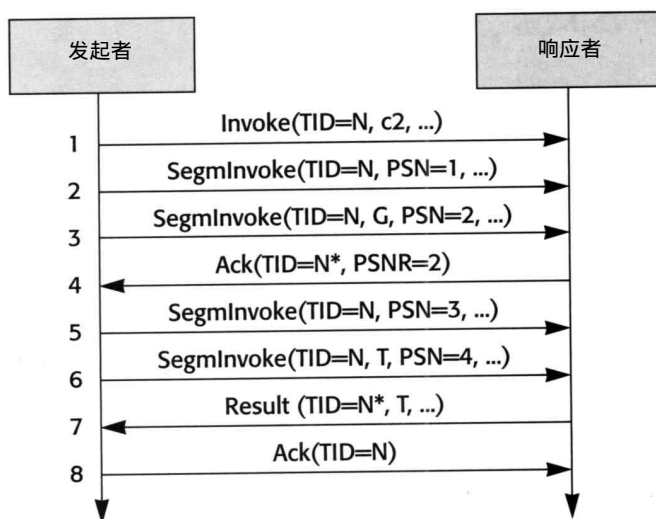


图15-10 调用消息的分割

选择性重发

图15-11给出一个使用分割第1级事务。在第一个分组群中的一个分组丢失了，响应者不得不请求重发这个分组。

发起者首先发出前三个分组，第二个分组丢失了。当响应者接收到具有 GTR的标志集时，它尝试重组这一分组群，但由于丢失了一个分组而失败。响应者返回一个 Nack要求重发丢失的分组，发起者重发丢失的分组，重发的分组具有RID标志集。一旦响应者接受到重发的分组，确认消息并结束事务。

注意PSN TPI在Ack PDU中是作为分组序列号（Packet Sequence Number）使用的。

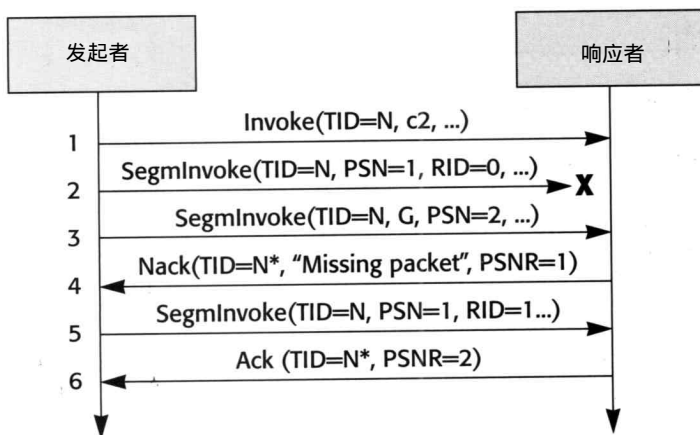


图15-11 选择性重发

15.9 术语定义

本规范采用了下列术语：

设备地址（Device Address） 分配给每一个设备的唯一的网络地址，遵循由国际标准

(如E.164的MSISDN地址, X.121的X.25地址, RFC791的IPv4地址)。一个地址唯一地标识一个发送和/或接收设备。

发起者 (Initiator) 发起一个事物的WTP提供者。

移动设备 (Mobile Device) 指通过无线链路连接到无线网络上的设备, 如: 电话、寻呼机、PDA等。尽管“移动”意味着设备是经常运动的, 但它也可以包括连接到无线网络上的固定或静止的无线设备(比如电子仪表上的无线调制解调器)。

网络类型 (Network Type) 网络类型指任何根据一定特点和标准(如空中接口)来分类的网络。网络类型的例子包括。GSM、CDMA、IS-136、iDEN、FLEX、ReFLEX和Mobitex。每一种网络都可能包括多种下层载体服务。

网络控制信息 (Protocol Control Information, PCI) 在WTP实体间交换的信息, 这些信息用于协调他们之间的操作。

协议数据单元 (Protocol Data Unit, PDU) WTP中规定的一个数据单元, 由WTP协议的控制信息以及可能的用户数据组成。

响应者 (Responder) 响应一个事务的WTP提供者被称为响应者。

服务数据单元 (Service Data Unit, SDU) 从上层协议来的信息单元, 它定义了要求下层协议提供的服务。

服务原语 (Service Primitive) WTP用户与WTP提供者之间的动作交互, 这些动作的概念是抽象的, 与具体的实现无关。

事务 (Transaction) 事务是在发起者和响应者之间动作交互的单元。一个事务从发起者发出的调用消息开始, 响应者通过接收到这一调用消息而被激活。在WTP中定义了几种事务类型, 调用消息表示了要求的事务类型, 而事务类型定义了完成这项事务所要求的动作。

用户数据 (User Data) 表示上层实体(如会话层)的两个WTP实体间传送的数据, WTP实体是为这些上层实体提供服务的。

WTP提供者 (WTP Provider) 一种被模型化的抽象的机器, 是从用户的观点出发, 看到的提供WTP服务实体的全部行为。

WTP用户 (WTP User) 一种抽象化的表示, 代表在一个单独的系统中使用了WTP服务的那些实体的总和。WTP用户的例子包括WAP会话协议WSP, 或直接在WTP上运行的应用。

15.10 缩略语

本规范采用了下列缩略语:

API	Application Programming Interface	应用编程接口
CDMA	Code Division Multiple Access	码分多址
CDPD	Cellular Digital Packet Data	蜂窝数字分组数据系统
CSD	Circuit Switched Data	电路交换数据
DBMS	Database Management System	数据库管理系统
DCS	Data Coding Scheme	数据编码方案
ETSI	European Telecommunication Standardization Institute	欧洲电信标准化协会
GPRS	General Packet Radio Service	一般无线分组业务

GSM	Global System for Mobile Communication	全球移动通信系统
GTR	Group Trailer, indicates the end of packet group	组群尾标志
GUTS	General UDP Transport Service	通用UDP传输业务
iDEN	Integrated Digital Enhanced Network	综合数字增强网
IE	Information Element	信息单元
IP	Internet Protocol	因特网协议
LSB	Least significant Bits	最低有效位
MPL	Maximum Packet Lifetime	分组最长存活期
MSISDN	Mobile Subscriber ISDN (Telephone number or address of derice)	移动用户的ISDN
MS	Mobile Station	移动台
MSB	Most significant Bits	最高有效位
PCI	Protocol Control Information	协议控制信息
PCS	Personal Communications System	个人通信系统
PLMN	Public Land Mobil Network	公共陆地移动网
R-Data	Relay Data	中继数据
RTT	Round-Trip Time	环回时间
SAR	Segmentation and Reassembly	分割与重组
SMSC	Short Message Service Center	短消息业务中心
SMS	Short Message Service	短消息业务
SPT	Server Processing Time	服务器处理时间
TIA/EIA	Telecommunications Industry Association /Electronic Industry Association	电信工业协会/电子工业协 会
PDU	Protocol Data Unit	协议数据单元
SAP	Service Access Point	业务接入点
SDU	Service Data Unit	业务数据单元
TTR	Transmission Trailer	传输尾标志
UDCP	USSD Dialogue Control Protocol	USSD对话控制协议
UDH	User-Data Heade	用户数据报头
UDHL	User-Data Header Length (see [GSM 03.40])	用户数据报头长度
UDL	User-Data Length (see [GSM 03.40])	用户数据长度
UDP	Unreliable Datagram Protocol	不可靠数据报协议
USSD	Unstructured Supplementary Service Data	无结构辅助业务数据
WAE	Wireless Application Environment	无线应用环境
WAP	Wireless Application Protocol	无线应用协议

WSP	Wireless Session Protocol	无线会话协议
WSP/B	Wireless Session Protocol - Browsing	无线会话协议/浏览
WTP	Wireless Transaction Protocol	无线事务处理协议
WDP	Wireless Datagram Protocol	无线数据报协议

15.11 参考标准

- [FLEX] FLEX Protocol Specification and FLEX Encoding and Decoding Requirements, Version G1.9, Document Number 68P81139B01, March 16, 1998, Motorola.
- [FLEXSuite] FLEX Suite of Application Protocols, Version 1.0, Document Number 6881139B10, October 29, 1997, Motorola.
- [GSM0260] ETSI European Digital Cellular Telecommunication Systems (phase 2+) : General Packet Radio Service (GPRS)—stage 1 (GSM 02.60)
- [GSM0290] ETSI European Digital Cellular Telecommunication Systems (phase 2) : Unstructured Supplementary Service Data(USSD)—stage 1 (GSM 02.90)
- [GSM0340] ETSI European Digital Cellular Telecommunication Systems (phase 2+) : Technical Realization of the Short Message Service (SMS) Point-to-Point (P) (GSM 03.40)
- [GSM0360] ETSI European Digital Cellular Telecommunication Systems (phase 2+) : General Packet Radio Service (GPRS)—stage 2 (GSM 03.60)
- [GSM0390] ETSI European Digital Cellular Telecommunication Systems (phase 2) : Unstructured Supplementary Service Data(USSD)—stage 2 (GSM 03.90)
- [GSM0490] ETSI European Digital Cellular Telecommunication Systems (phase 2) : Unstructured Supplementary Service Data(USSD)—stage 3 (GSM 04.90)
- [IS130] EIA/TIA IS-130.
- [IS135] EIA/TIA IS-135.
- [IS136] EIA/TIA IS-136.
- [IS176] EIA/TIA IS-176—CDPD 1.1 specifications.
- [IS637] TIA/EIA/IS-637: Short Message Services for Wideband Spread Spectrum Cellular Systems.
- [ISO7498] ISO 7498 OSI Reference Model.
- [ISO8509] ISO TR 8509 Service conventions.
- [ReFLEX] ReFLEX25 Protocol Specification Document, Version 2.6, Document Number 68P81139B02-A, March 16, 1998, Motorola.
- [RFC2119] S. Bradner "key Words for use in RFCs to Indicate Requirement Levels",RFC2119.
URL: <http://www.internic.net/rfc/rfc2119.txt>
- [TR45.3.6] General UDP Transport Teleservice (GUTS) ñ Stage III, TR45.3.6/ , 97.12.15.

- [WAE] "Wireless Application Environment Specification",WAP Forum.
URL: <http://www.wapforum.org/>
- [WAPARCH] "Wireless Application Protocol Architecture Specification",WAP Forum.
URL: <http://www.wapforum.org/>
- [WAPUD] "WAP and GSM USSD" WAP Forum.
URL: <http://www.wapforum.org/>
- [WDP] "Wireless Datagram Protocol Specification" , WAP Forum.
URL: <http://www.wapforum.org/>
- [WSP] "Wireless Session Protocol Specification" , WAP Forum.
URL: <http://www.wapforum.org/>

15.12 参考资料

- [RFC768] J. Postel, "User Datagram Protocol" , RFC768, August 1980
URL: <http://www.internic.net/rfc/rfc768.txt>
- [RFC791] J. Postel, "IP: Internet Protocol" , RFC791
URL: <http://www.internic.net/rfc/rfc791.txt>

15.13 默认计时器和计数器值

定时器是初始的估计值，尚未验证。
定时器的单位为秒，计数器以一个事件发生的次数来表示。

15.13.1 GSM SMS

最大的环回时间是 40秒（尽管通常的时间是 10秒）。定时器建议采用表 15-58和表15-59所列的值。

表15-58 GSM SMS定时器建议采用的数值（1）

类 型	TYPE	没有用户确认	有用户确认
Acknowledgment interval (A)	ACK_T	10	10
	S_ACK_T	0	5
	L_ACK_T	20	20
Retry interval (R)	TTR_T	60	60
	S_TTR_T	35	35
	L_TTR_T	70	70
	GTR_T	45	45
Wait timeout interval (W)	WAIT_T	300	300

表15-59 GSM SMS定时器建议采用的数值（2）

计 数 器 名	栈 确 认 值	用户确认值
Max Retransmissions	4	4
Max Ack Timer Expiration	4	4

15.13.2 GSM USSD

最大环回时间是5秒，定时器建议采用表 15-60和表15-61所列值。

表15-60 GSM USSD 定时器建议采用的数值（1）

时 间 间 隔	类 型	没有用户确认	有用户确认
Acknowledgment interval (A)	ACK_T	10	10
	S_ACK_T	0	5
	L_ACK_T	10	10
Retry interval (R)	TTR_T	20	20
	S_TTR_T	14	14
	L_TTR_T	20	20
	GTR_T	10	10
Wait timeout interval (W)	WAIT_T	60	60

表15-61 GSM USSD 定时器建议采用的数值（2）

计 数 器 名	栈 确 认 值	用户确认值
Max Retransmissions	4	4
Max Ack Timer Expiration	4	4

15.13.3 CDPD

最大环回时间是3秒，定时器建议采用表 15-62和表15-63所列值。

表15-62 CDPD定时器建议采用的数值（1）

时 间 间 隔	类 型	没有用户确认	有用户确认
Acknowledgment interval (A)	ACK_T	2	2
	S_ACK_T	0	2
	L_ACK_T	2	3
Retry interval (R)	TTR_T	3	3
	S_TTR_T	3	3
	L_TTR_T	3	3
	GTR_T	3	3
Wait timeout interval (W)	WAIT_T	30	30

表15-63 CDPD 定时器建议采用的数值（2）

计 数 器 名	栈 确 认 值	用户确认值
Max Retransmissions	8	8
Max Ack Timer Expiration	6	6

15.13.4 电路交换数据

最大环回时间是3秒，定时器建议采用表 15-64和表15-65所列值。

表15-64 电路交换定时器建议采用的数值（ 1 ）

时间 间 隔	类 型	没有用户确认	有用户确认
Acknowledgment interval (A)	ACK_T	2	2
	S_ACK_T	0	1
	L_ACK_T	7	7
Retry interval (R)	TTR_T	5	5
	S_TTR_T	3	4
	L_TTR_T	10	10
	GTR_T	3	3
Wait timeout interval (W)	WAIT_T	40	40

表15-65 电路交换定时器建议采用的数值（ 2 ）

计 数 器 名	栈 确 认 值	用户确认值
Max Retransmissions	4	4
Max Ack Timer Expiration	4	4

15.13.5 计时器的使用

有相当一部分定时器有相类似的行为但值不同，这些定时器的定义是为了最优地使用可得到的带宽。表 15-66对在一个事务中不同的消息应使用什么定时器间隔作了说明。

表15-66 关于在一个事务中不同的消息应使用的定时器间隔的说明

消 息 类 型	第 2 类	第 1 类
Invoke message	TTR_T	S_TTR_T
Hold on acknowledgment	ACK_T	—
Result message	L_TTR_T	—
Last acknowledgment	L_ACK_T	S_ACK_T
Last packet of packet group	GTR_T	GTR_T

对0类，没有定时器值是可用的。

15.14 PICS 实施说明

宣称遵循本规范的协议实施者必须填写一个在本节中给出的 PICS实施说明，它包括标识提供者和实现的必要信息。

15.14.1 简介

遵循本规范的协议的实施者必须填写下面的协议实现一致性说明（ protocol Implementation Conformance Statement ）的实施说明。

一个完整的PICS 说明是在解决问题中完成的， PICS是对协议实现了哪些功能和选项的陈述，它可以有以下用途：

- 对协议实现者，作为检查列表，可以降低由于忽视而出现与协议不一致的危险。
- 对实现的协议提供者和获得者，或潜在的获得者，作为这一实现功能的详细表述。这种

描述是建立在公共的由 PICS 提供的理解基础之上。

- 具体实现的用户，或潜在的用户，作为最初考察这个实现和其他实现互连的可能性的基础（注意可能从来就不能保证互连，来自于不兼容 PICS 间的无法互连是可以预见的）。
- 对协议测试者，作为选择适当测试方法以及测试实现所宣称的兼容性的基础。

15.14.2 缩写和特殊符号

1. 状态符号

M	强制的
O	可选的
O、<n>	可选的，但是至少要求支持数字 <n> 标出的那组选择中的一项
X	禁止的
<pred>	条件选项标记，包括谓词标识符（参见 15.14.3 节中“条件状态”）
^	逻辑否定形式，应用于条件选项的谓词

2. 其他的符号

<r>	一个条款的接受方
<s>	一个条款的发送方

15.14.3 填写 PICS 实施说明的指南

1. PICS 实施说明的一般结构

PICS 实施说明的第一个部分——实现标识和协议总结，应该填写足够的信息以完整地标识出提交者和实现本身。

PICS 实施说明的主要部分是一个具有固定格式并被分成几个主要部分的问卷表格，这些部分可以分为更加细小的子部分，每一个子部分包括一组项。对问卷各项的答案要填在最右边的栏里，要么是简单的在限选答案（如“是”或“不是”）上作一个记号，或填入一个值或一个值的范围。

注意 有些项在一组可能的答案中可以有二个或两个以上的选择，在这种情况下，所有的有关值都要作记号。

每一项是由第一列中的项索引标识的，第二列是要填写的答案，第三列为在指向本规范主体中定义该项的文字的索引。剩下的列记录了这一项的状态，即对这一项的支持是强制性的、可选的、禁止的，还是有条件的，并且留有了回答问题的空间（参见 15.14.3 节中“条件状态”）。

一个提交者还可以提供更多的信息，这些信息被归为附加信息或例外信息，每一种附加的信息分别在一个标为 A<i> 或 X<i> 的子部分中提交。为了交叉索引，<i> 必须是这一项唯一的、不会混淆的标识（如一个数字），对它的格式或写法没有其他的限制和要求。

一个包括任意附加信息和例外信息的、填写完整的 PICS 实施说明，就是所讨论的协议实施的一致性说明。

注意 当一个实现可以用多于一种方法来配置时，单个 PICS 就可以描述所有的这些配置。但是，提供者也可以提交多个 PICS，每一个 PICS 是实现配置性能的一个子集，以使信息的表述更为清楚和方便。

2. 附加信息

提交者可以在补充信息的条款中提供辅助解释 PICS的信息。补充信息不宜过多,完整的 PICS并不要求一定有这类信息。例如,多种环境和配置下设置履行方式的摘要,或者基于特定应用的基本原理都可以算作补充信息,但是在多数情况下,这些应用的具体属性(当然可能是可选择的属性),都在协议履行中说明。

补充信息中条款的索引可以写在调查表中答案的旁边,其条款可以包括在“异常信息”的条款中。

3. 异常信息

有时提交者希望能够用必需或禁止回答与要求中相冲突的条款,这答案在现成表中找不到。提交者可以将缺少的答案写入支持列,同时加上 X*i*作为异常信息的标志,并且在“异常信息”的条款中提供合适的基本原理。

不要求履行的异常信息和规范一致。

注意 产生这种情况是由于发布的标准中存在不足,对无法履行的要求有必要作出更改。

4. 条件状态

(1) 条件条款

PICS 实施说明包括了许多条件条款,这些条款是说明应用是否支持某些条款或某些数值的状态,有“必需”、“任选”或“禁止”。

在很多情况下,不管条款应用是否为有条件的,条款也都具有状态。应用的同一条件下的条款组前面有关于条件的基本问题,如果选择了“Not Applicable”,就可以在调查表中略去,否则,在状态列中用一个或多个条件符号(单独行中)表明专门的条件条款。

条件条款的形式是“<pred>:<x>”,其中<pred>是15.14.3节中描述的“谓词”,<x>是状态符号M、O、O.<n>或X之一。

如果条件条款中的谓词是“真”(参见15.14.3节中的“谓词”),则条件条款可以应用,其状态由后续状态标识表明,答案列以常用方式标记。如果条件条款中的谓词是“假”,则在相应行应作不可应用标识(N/A)。多行条件条款的每一行都必须被标记,除了N/A外,一行要求一个答案。

(2) 谓词

一个谓词是下列内容之一:

- 在PICS说明中条款的一个引用。如果这个条款的被标记为“支持的”,谓词的值为真,否则为“假”。
- 一个谓词的名字,这个谓词在PICS实施说明的其他地方被定义(通常在主要功能或在包含这一有条件条款的章节的末尾)。
- 在一条款引用或谓词名字前的逻辑非前缀“^”。如果由省略“^”而形成的谓词的值为“假”,则整个谓词的值为“真”。反之亦然。

谓词名字的定义如下:

- 一个条款的引用,如上面所描述的。
- 一个包含比较运算符(=, < 或其他)的关系式,或至少一个操作数的关系式,取其中的数据作为条款索引。如果用支持列中的数值代替每个条款索引,关系式成立,那么,谓词值为“真”。

- 使用布尔运算符，运用布尔运算符“ AND ”、“ OR ”、“ NOT ”和括号，将以上两点中的简单谓词组合。如果谓词解释使得布尔表达为“ 真 ”，那么，谓词值为“ 真 ”。如果条款的索引被谓词或谓词定义引用，条款列中有星号标记。

15.14.4 标识

1. 实现标识（见表15-67）

表15-67 实现标识

提交者
对PICS有疑问时的联系方式
实现名和版本号
其他必要标识信息（如机器，操作系统的名字版本，以及系统名）

注：

前三项是所有实现都必须的，其他信息可根据标识的要求适当填写。

术语：Name（名字）和 Version（版本）应该解释为适当的词义以对应提供者的术语体系（例如 Type, Series, Model ），

2. 协议概述（见表15-68）

表15-68 协议概述

协议规范标识	WAP 无线事务处理协议
PICS proforma的修正和勘误表的标识	
支持的协议版本	
是否要求有异常信息？（见 15.14.3节中“ 异常信息 ”）	是 否
（回答“ 是 ”意味着这一实现与本规范不一致）	
日期	

15.14.5 无线事务协议

1. 适用性
- 无线事务协议适用于宣称与规范一致的所有履行者。

2. 协议功能

(1) 事务类型（见表15-69）

表15-69 事务类型

项 目	功 能	参 考	状 态	支 持
INCL0	应用是否支持类别0事务的初始化？	0	M	是 否
INCL1	应用是否支持类别1事务的初始化？	0	M	是 否
INCL2	应用是否支持类别2事务的初始化？	0	O	是 否
RECL1	应用是否支持类别1事务的响应？	0	M	是 否
RECL2	应用是否支持类别2事务的响应？	0	O	是 否

(2) 协议特点（见表15-70）

表15-70 协议特点

项 目	功 能	参 考	状 态	支 持
UACK	应用是否支持用户确认功能	0	O	是 否
CONC	应用是否支持连锁	0	O	是 否
SEPA	应用是否支持分割相互连锁的 PDU	0	M	是 否
RETR	应用是否支持重发直到得到确认	0	M	是 否
TRAB	应用是否支持退出事务	0	M	是 否
VERS	应用是否支持版本处理	0	M	是 否
ERRO	应用是否支持错误处理	0	M	是 否
VERI	应用是否支持事务验证的初始化	0	O	是 否
VERR	应用是否支持对事务验证的响应	0	M	是 否
TPIE	应用是否支持 Error TPI	0	M	是 否
TPII	应用是否支持 Info TPI	0	M	是 否
TPIO	应用是否支持 TPI选择	0	O	是 否
TPIP	应用是否支持 PSN和TPI	0	O	是 否
SAR	应用是否支持分段和重组	0	O	是 否

15.15 历史和联系信息

表15-71列出了历史和联系信息。

表15-71 历史和联系信息

文献历史		
日期	状态	备注
1998年4月29日	规范	第一版
联系信息：		
http://www.wapforum.org		
technical-comments@wapforum.org		