

目 录

一、彩信接收分析.....	2
1.1 彩信（1.0）接收信令流程分析.....	2
1.1.1 信令过程异常分析.....	2
1.1.2 成功率分析.....	3
1.1.3 彩信接收失败原因分析.....	3
1.1.3.1 timeout 失败原因分析.....	3
1.1.3.2 abort 失败原因分析.....	4
1.1.3.3 disconnect 失败原因分析.....	5
1.1.3.4 PDP context deactivation 失败原因分析.....	5
1.1.3.5 status_code 异常值失败原因分析.....	6
1.1.4 彩信（1.0）接收用户原因区分.....	6
1.2 彩信（2.0）接收信令流程分析.....	6
1.2.1 信令过程异常分析.....	6
1.2.2 成功率分析.....	7
1.2.3 彩信接收失败原因分析.....	7
1.2.3.1 timeout 失败原因分析.....	7
1.2.3.2 reset 失败原因分析.....	8
1.2.3.3 PDP deactivation 失败原因分析.....	8
1.2.3.4 status_code 异常值失败原因分析.....	9
1.2.4 彩信（2.0）接收用户原因区分.....	9
二、彩信发送分析.....	9
2.1 彩信（1.0）发送结果分析.....	10
2.1.1 成功率分析.....	10
2.1.2 彩信发送失败原因分析.....	10
2.1.2.1 timeout 失败原因分析.....	10
2.1.2.2 abort 失败原因分析.....	11
2.1.2.3 disconnect 失败原因分析.....	11
2.1.2.4 PDP context deactivation 失败原因分析.....	12
2.1.2.5 status_code 异常值失败分析.....	12
2.1.3 彩信（1.0）发送用户原因区分.....	13
2.2 彩信（2.0）发送结果分析.....	13
2.2.1 成功率分析.....	13
2.2.2 彩信发送失败原因分析.....	13
2.2.3 彩信（2.0）发送用户原因区分.....	13

一、彩信接收分析

数据业务的信令流程相对来说较为复杂，且具有很大的不确定性，尤其涉及到具体业务应用的协议层部分，由于手机终端的相关协议开发定义可能并不规范，这给业务过程以及结果的分析判断带来了很大困难。对于 MMS 分析来说，这种情况尤为突出。

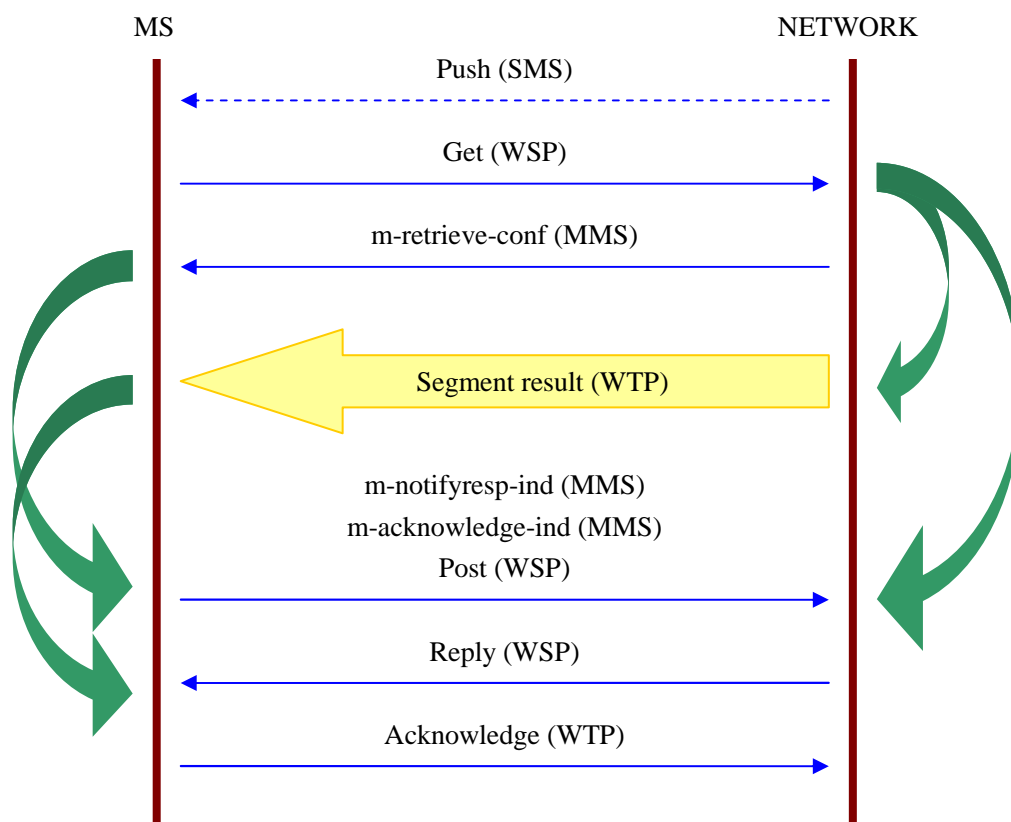
1.1 彩信（1.0）接收信令流程分析

在信令分析过程中因传输协议的不同而将彩信区分为彩信（1.0）与彩信（2.0）。其中彩信（1.0）采用 WSP/WTP 协议，彩信（2.0）采用 HTTP 协议。

1.1.1 信令过程异常分析

彩信（1.0）的信令流程中，缺失某一条信令或者多条信令都比较常见。缺失信令的原因主要有两种：一是信令采集丢包造成；二是手机终端相关协议开发定义不规范造成；三是因为网络性能原因引起数据传输障碍。从实际的分析结果来看，后两者更为普遍。

彩信（1.0）具体信令流程如下图：



分析彩信（1.0）的信令流程，主要包括三条信令交互，分布是：

- m-retrieve-conf (MMS)
- m-notifyresp-ind (MMS)、m-acknowledge-ind (MMS)、Post (WSP)中任意一条
- Reply (WSP)

在一般的分析结果中，信令的缺失情况大致如下：

信令缺失类别	比例
信令齐全	80%
缺失 m-notifyresp-ind/m-acknowledge-ind/post & reply	9%
缺失 m-retrieve-conf & m-notifyresp-ind/m-acknowledge-ind/post & reply	6%
仅缺失 m-notifyresp-ind/m-acknowledge-ind/post	3%
缺失 m-retrieve-conf & reply	1%
仅缺失 m-retrieve-conf	0.5%
仅缺失 reply	0.4%
缺失 m-retrieve-conf & m-notifyresp-ind/m-acknowledge-ind/post	0.1%
彩信总量	100.0%

信令齐全的比例大致是 80%，而信令缺失并不一定意味着彩信接收的失败。

1.1.2 成功率分析

从用户角度来看，只要其接收到所有的彩信数据包，无论有无回复确认消息，实际上都已经完成了彩信的接收。因此，判断彩信接收的成功与否主要考虑手机终端是否接收到了所有的彩信数据和 WSP status 是否正常。

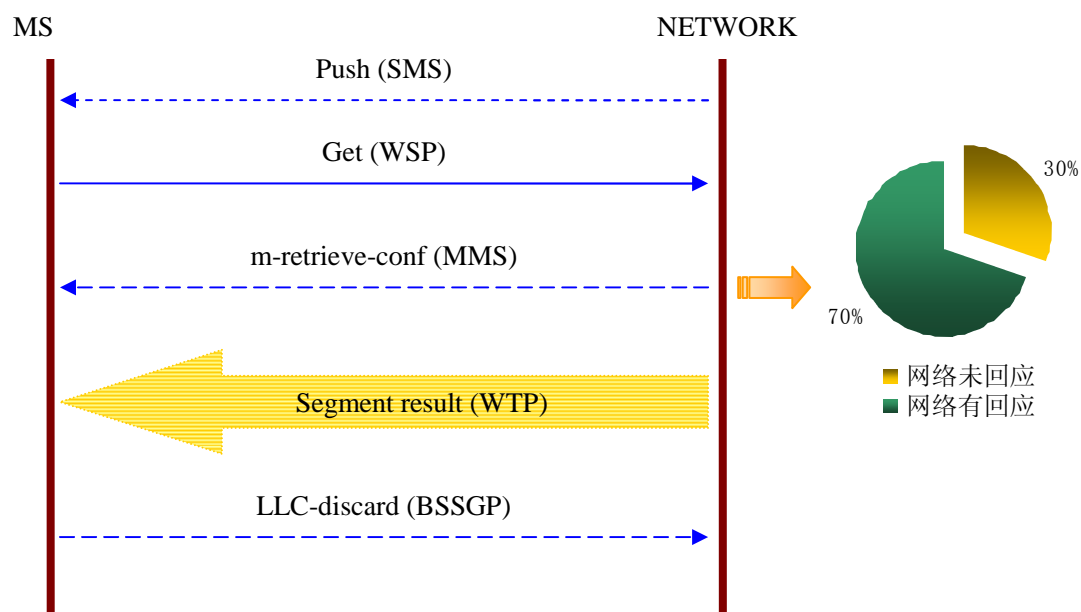
1.1.3 彩信接收失败原因分析

总体来说，彩信（1.0）接收失败的原因可分为如下五大种：

- 接收过程中断，且长时间无响应导致超时失败
- 接收过程中用户发出 abort(WTP)，且再无重新请求导致失败
- 接收过程中用户发出 disconnect(WSP)，且再无重新请求导致失败
- 接收过程中用户发出 Deactivate PDP context request(SM)信令终止
- status code 异常值导致失败

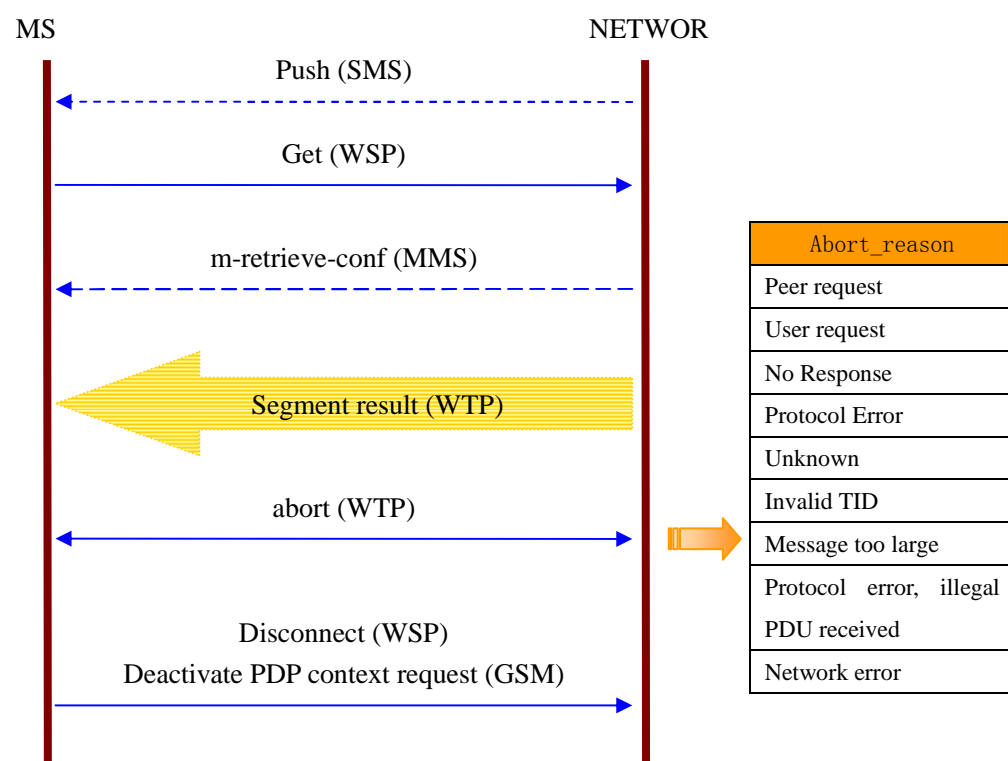
1.1.3.1 timeout 失败原因分析

在彩信（1.0）接收的信令过程中，如果手机终端侧发出 GET 请求后或者接收到 m-retrieve-conf 消息后无任何消息响应且数据包也未接收完全，则定义为彩信（1.0）接收的超时失败。其具体信令过程如下：



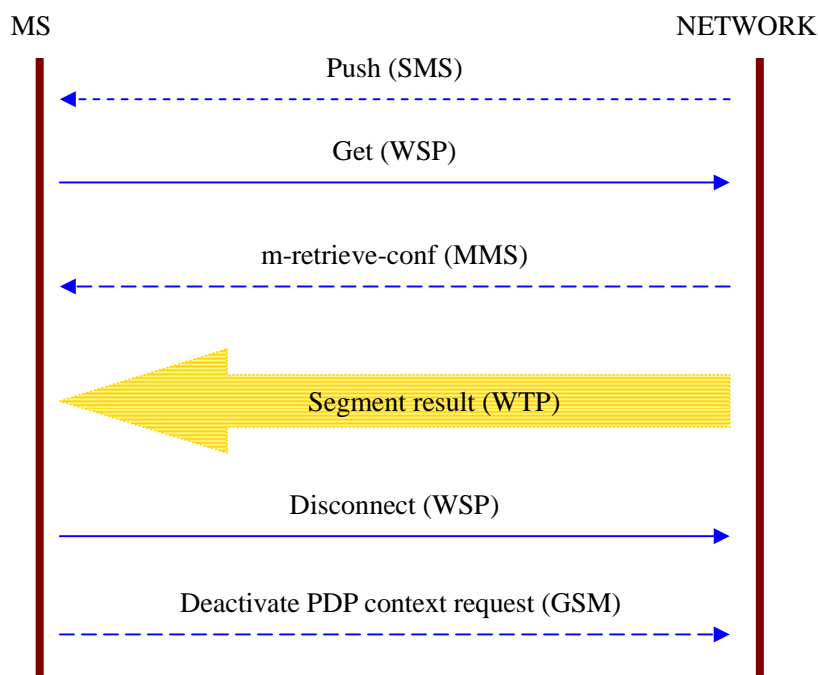
1.1.3.2 abort 失败原因分析

手机终端在发出 GET 请求后，随时都有可能出现 abort 导致进程终止，但是 abort 的发出并不意味着彩信接收的终止或失败，这是因为手机终端有可能通过 m-notifyresp-ind(mms_status:deferred) 等消息恢复彩信的接收。因此我们所定义的 abort 失败是指进程终止后并未恢复，且彩信接收也没有完成的事件。具体信令过程如下：



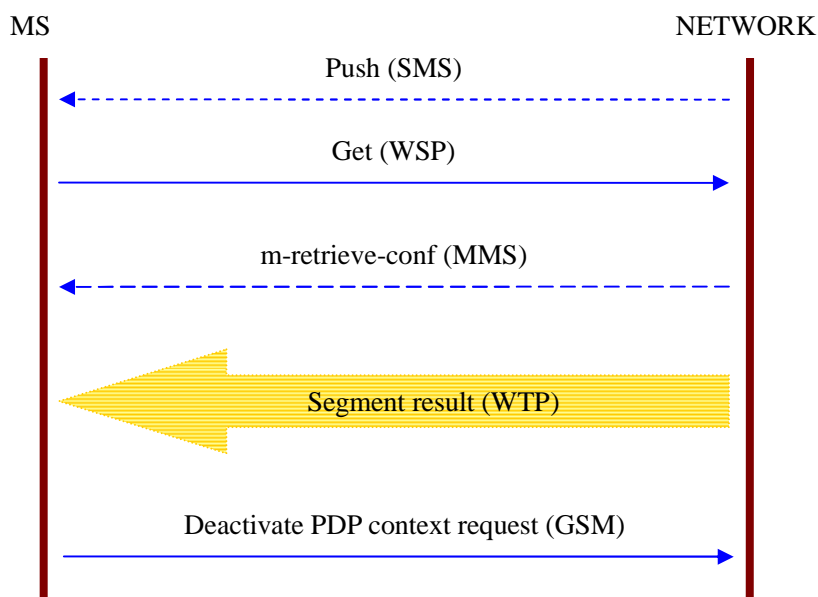
1.1.3.3 disconnect 失败原因分析

Disconnect 失败主要是手机终端在收发彩信的过程中主动终止传输的一种现象，即用户原因导致的拆线事件。准确来讲，disconnect 原因导致的彩信收发过程失败并不是真正意义上的失败。具体信令流程如下：



1.1.3.4 PDP context deactivation 失败原因分析

与 disconnect 失败原因类似，PDP context deactivation 也是用户发起的一种中断彩信收发过程的操作，只不过在信令分析过程中，并没有出现 disconnect 消息。这种现象的出现主要是由于手机终端协议开发的不规范或者用户操作问题导致。具体信令流程如下：



1.1.3.5 status_code 异常值失败原因分析

Status_code 源于 reply(WSP)消息中，正常彩信流程应返回“200 ok”的信息，表示传输一切正常。当出现异常情况的时候，协议规范要求 status code 中返回相应原因值，而信令流程基本完整。Status 异常值最多是 400 bad request，一般占总体 95% 以上。

1.1.4 彩信（1.0）接收用户原因区分

本次分析中可见用户原因导致彩信接收失败的数量占了较大的比重。如果简单的将用户原因和网络原因予以区分，可以了解因网络原因导致的彩信（1.0）接收失败的情况。

用户原因定义：abort(peer request/user request), disconnect, pdp deactivation, status code(bad request)

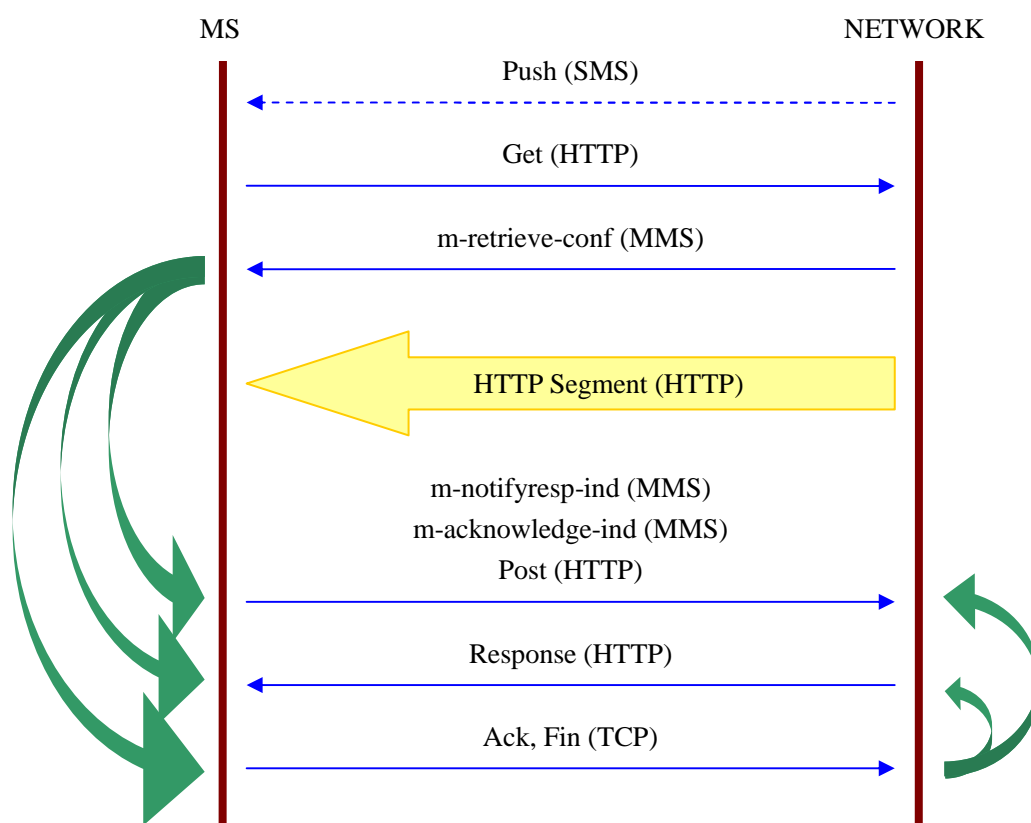
注：peer request 并不一定由手机终端主动触发，例如：长时间等不到网络侧响应也有可能触发 peer request 消息。因此单纯将 peer request 归类于用户原因并不完全准确。

1.2 彩信（2.0）接收信令流程分析

与彩信（1.0）接收不同，彩信（2.0）采用的是 HTTP 协议，其过程和结果的判断均有所不同，但是总体分析思路基本一致。

1.2.1 信令过程异常分析

彩信（2.0）接收除传输协议有所不同外，其信令流程与彩信（1.0）基本相同，同样受数据业务本身的复杂性所致，彩信（2.0）接收的信令流程也有一定的不确定性。信令流程缺失的现象时有发生，具体情况如下：



由上图，彩信（2.0）接收流程中有四条主要消息，分别是：

- m-retrieve-conf (MMS)
- m-notifyresp-ind (MMS)、m-acknowledge-ind (MMS)、Post (HTTP)中任意一条
- Reply (HTTP)
- Ack, fin (TCP)

本次分析过程中，信令缺失的情况如下：

信令缺失类别	比例
信令齐全	56%
仅缺失 Ack, fin	28%
缺失 m-notifyresp-ind/m-acknowledge-ind/post & response & Ack, fin	6%
缺失 m-notifyresp-ind/m-acknowledge-ind/post & response	4%
全部缺失	3%
仅缺失 m-notifyresp-ind/m-acknowledge-ind/post	1%
缺失 m-retrieve-conf & response & Ack, fin	0.7%
缺失 Ack, fin & m-notifyresp-ind/m-acknowledge-ind/post	0.6%
缺失 Ack, fin & response	0.4%
仅缺失 m-retrieve-conf	0.1%
仅缺失 response	0.1%
缺失 Ack, fin & m-retrieve-conf	0.1%
总计	100.0%

在信令交互的完整性上，彩信（2.0）的接收流程中信令缺失的情况更为严重，尤其是标识接收完成的 Ack,fin 消息，有近 30%的接收流程中没有该消息。另外有超过 10%的流程中没有 response 消息，即没有 status_code。。同样，接收流程的不完整并不意味着接收的失败。

1.2.2 成功率分析

在成功与否的判断中，除了考虑彩信接收流程的完整性，更主要的是判断彩信数据是否已经完全接收成功以及传输状态标识。这里把彩信数据完整接收和 HTTP status_code 不为”400+”或”500+”的流程定义为成功，而不是单纯考虑某一条或者某几条信令消息是否存在和正常。

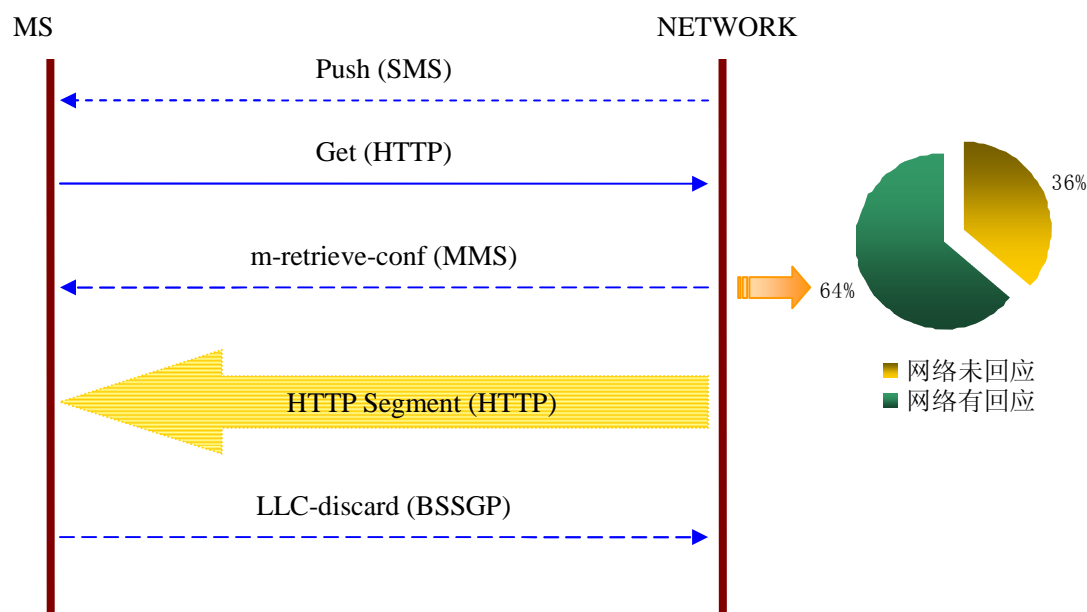
1.2.3 彩信接收失败原因分析

同彩信（1.0）接收失败原因类似，彩信（2.0）接收失败原因大致可分为 4 种：

- 接收过程中断，且长时间无响应导致超时失败
- 接收过程中用户发出 reset(TCP) ，且再无重新请求导致失败
- 接收过程中用户发出 Deactivate PDP context request(SM)信令终止
- status code 异常值导致失败

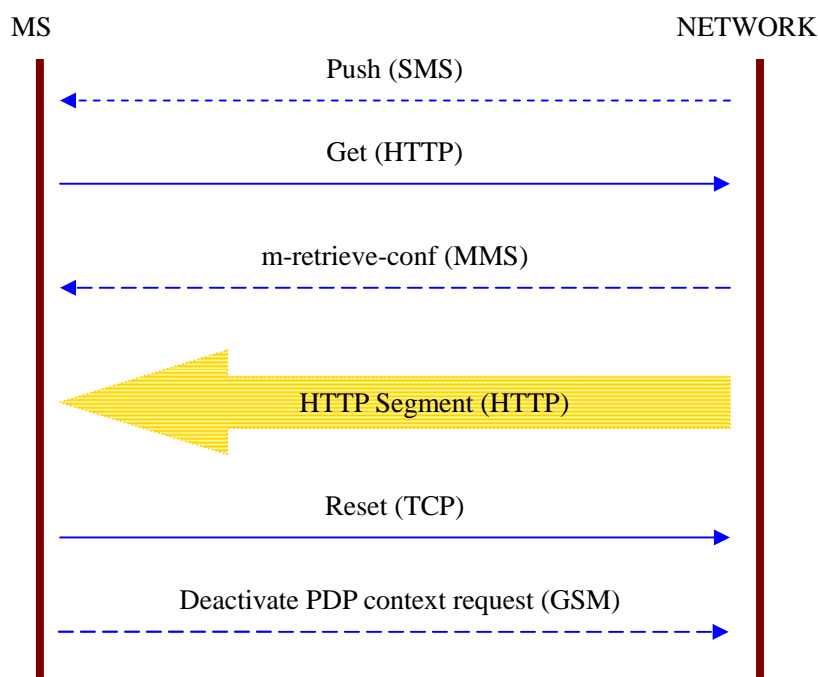
1.2.3.1 timeout 失败原因分析

同彩信(2.0)接收的信令过程相同，如果手机终端侧发出 GET 请求后或者接收到 m-retrieve-conf 消息后无任何消息响应且数据包也未接收完全，则定义为彩信（2.0）接收的超时失败。其具体信令过程如下：



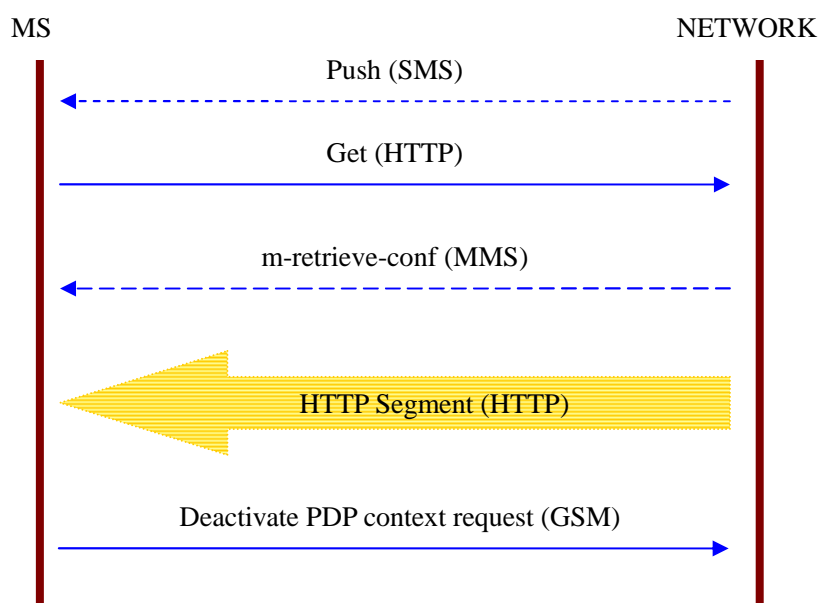
1.2.3.2 reset 失败原因分析

在彩信（2.0）的接收过程中，有可能出现 reset 消息，该消息表示传输的重置，这里的 reset 失败是指传输重置没有成功，直接导致彩信接收中止的一种现象。



1.2.3.3 PDP deactivation 失败原因分析

PDP context deactivation 失败是指手机终端在完成彩信接收之前提前中断数据传输的一种操作，其特点是在 Deactivated PDP context request 信令之前没有任何表示中断的消息而直接拆线。在这里我们归纳为 PDP deactivation 失败，是用户原因的一种。具体信令流程如下：



1.2.3.4 status_code 异常值失败原因分析

在彩信（2.0）的接收流程中，status_code 存在于 HTTP 协议中的 response 消息，用来标识协议传输状态。彩信（2.0）接收过程返回 status code 异常的比例占 30%。具体如下：

Status_code	比例
400 Bad Request	81.0%
500 Internal Server Error	10.9%
405 Method not allowed	2.0%
404 Not Found	0.9%
412 Precondition failed	0.6%
502 Bad Gateway	0.2%
501 Not Implemented	0.2%
403 Forbidden	0.2%
503 Service Unavailable	0.1%

比例最大的原因值是 400 Bad Request，即用户错误请求原因引起。

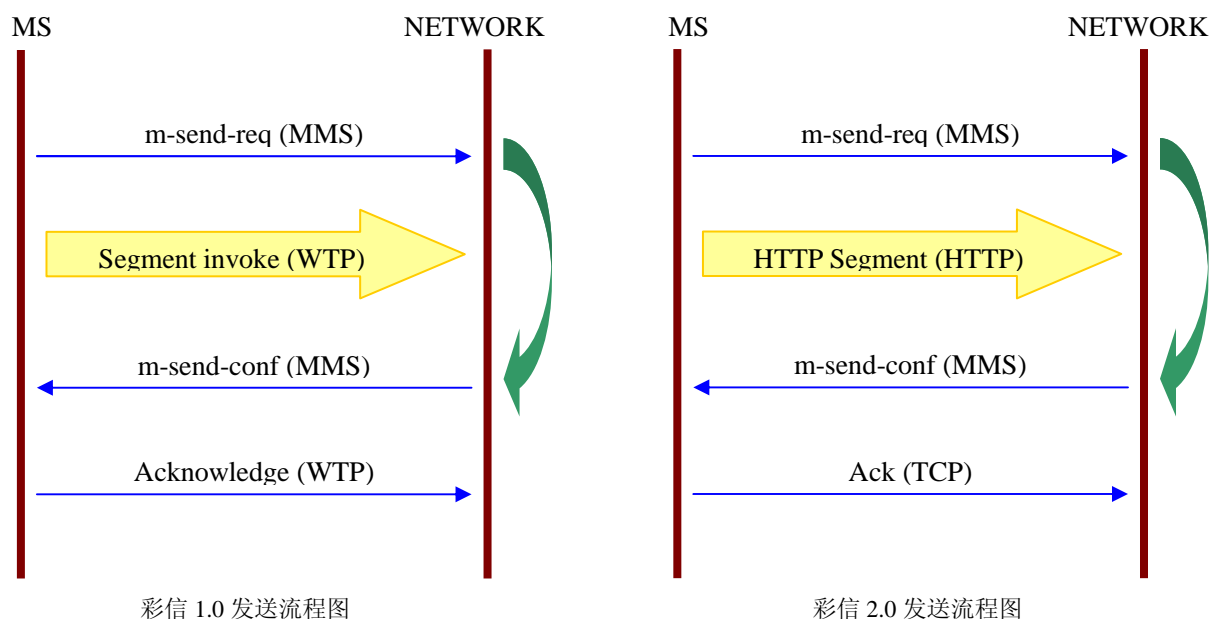
1.2.4 彩信（2.0）接收用户原因区分

与彩信（1.0）接收分析结果类似，用户原因同样在彩信（2.0）接收失败结果中占有很大比重。简单将用户原因的失败做如下定义：

Reset, PDP context deactivation, Status code(400 Bad Request)

二、彩信发送分析

与彩信接收相同，彩信发送也有 1.0 与 2.0 之分，不过在信令流程方面彩信的发送要相对简单，并且主要依靠 MMS 应用协议。具体如下：



彩信发送的过程中，同样可能出现 abort、disconnect、PDP context deactivation 的中断操作，也会存在无任何消息标识的突然中断最终引起超时失败，另外，承载 m-send-conf(MMS)消息的 WSP/HTTP 协议中包含 status code 信息来返回传输状态标识。

2.1 彩信（1.0）发送结果分析

2.1.1 成功率分析

对彩信发送成功的定义仍然是判断彩信数据包是否已经全部发送出去以及网络侧返回的 status code 是否正常。

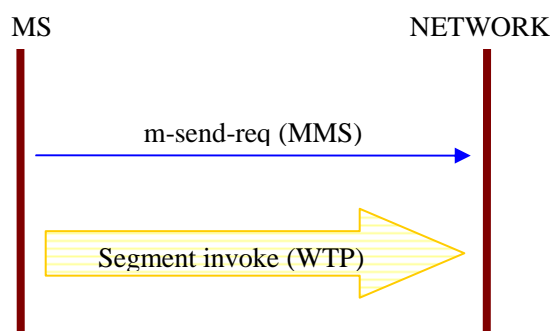
2.1.2 彩信发送失败原因分析

彩信（1.0）发送失败可以分为如下 5 种类型：

- 接收过程中断，且长时间无响应导致超时失败
- 接收过程中用户发出 abort(WTP)，且再无重新请求导致失败
- 接收过程中用户发出 disconnect(WSP)，且再无重新请求导致失败
- 接收过程中用户发出 Deactivate PDP context request(SM)信令终止
- status code 异常值导致失败

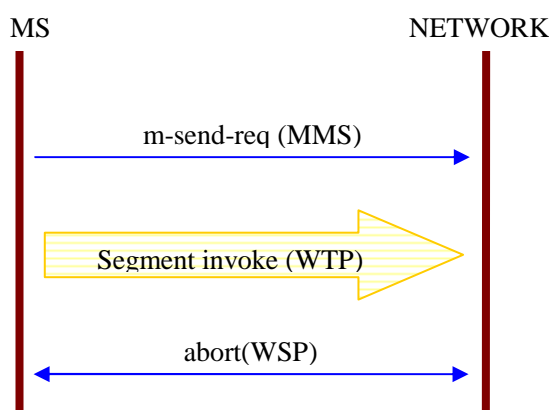
2.1.2.1 timeout 失败原因分析

彩信（1.0）发送超时可分为两种情况，一是在手机终端发出 m-send-req 后就再也没有任何数据包，二是手机终端在发送 segment invoke 的过程中突然停止且再无响应。具体流程如下：



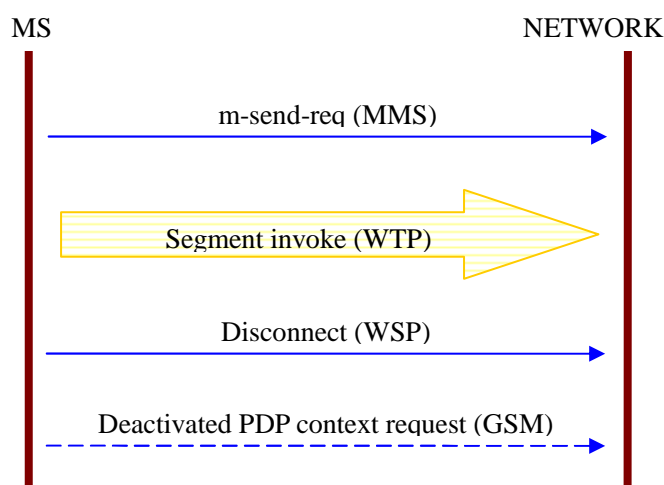
2.1.2.2 abort 失败原因分析

Abort 消息出现后，数据传输再未恢复，则定义为彩信（1.0）发送 abort 失败。Abort 失败大都因用户原因导致，具体信令流程及细分统计结果如下：



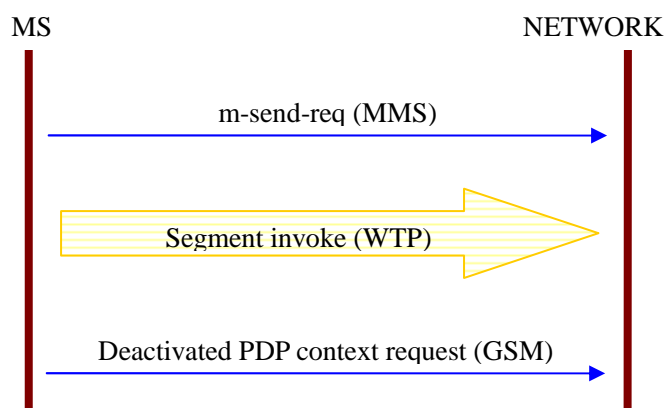
2.1.2.3 disconnect 失败原因分析

Disconnect 是用户操作的一种拆线行为，属于 WSP 协议，在本次分析结果中定义的 disconnect 失败并不包含用户 disconnect 后重新在发起 connect 进行续传并成功的事件。因此数量较少，仅有 2 次。具体流程如下：



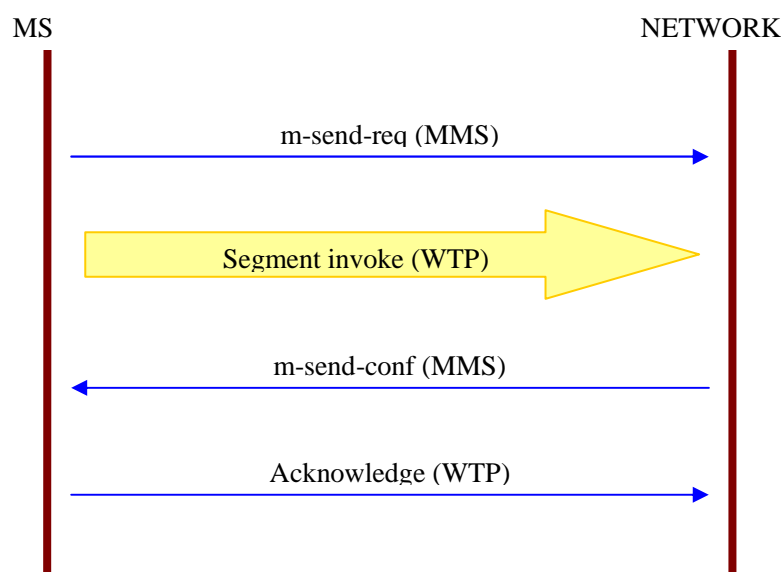
2.1.2.4 PDP context deactivation 失败原因分析

与彩信接收情况相同，该失败是指彩信发送过程中无任何过程消息而直接触发 Deactivated PDP context request 的事件，具体流程如下：



2.1.2.5 status_code 异常值失败分析

彩信（1.0）发送过程中出现 status code 异常时，从信令流程本身来看是正常的，并且绝大多数的 WSP 协议层中的 status 也返回的是“200 ok”。而这里的 status code 异常是存在于 MMS 协议中的 response status 消息之中。具体流程如下：



2.1.3 彩信（1.0）发送用户原因区分

总结彩信（1.0）发送失败原因，其中用户原因同样占有很大的比重。在 status code 异常原因值分析结果中，就过程本身而言，实际上已经完成了彩信的发送流程，而对于用户而言则是网络侧返回了错误提示，对于类似错误提示，很有可能是用户发送的彩信数据不规范或者非法发送导致。因此，将用户原因的发送失败作如下定义并排除：

用户原因定义：abort(peer request/user request), disconnect, pdp context deactivation, status code (unsupported-message/service-denied)

2.2 彩信（2.0）发送结果分析

2.2.1 成功率分析

彩信（2.0）发送的结果主要考虑信令流程的完整性和返回的 mms status 是否正常。

2.2.2 彩信发送失败原因分析

彩信（2.0）发送失败原因可分为 3 种：

- 接收过程中断，且长时间无响应导致超时失败
- 接收过程中用户发出 Deactivate PDP context request(SM)信令终止
- status code 异常值导致失败

与彩信（1.0）发送失败原因基本相同，本次分析结果中的彩信（2.0）发送失败原因除了超时之外，既是用户主动拆线和发送数据不规范或者乱码导致，并且大都集中在少数用户中。具体的 status 内容一般有两种：即 “Error-service-denied” 和 “Error-unsupported-message”

2.2.3 彩信（2.0）发送用户原因区分

彩信（2.0）的发送量相对较少，参考彩信（1.0）的失败原因分类，将其用户原因定义为：

pdp context deactivation, status code (unsupported-message/service-denied)