

TConnd 维护使用文档



腾讯科技（深圳）有限公司

版权所有 侵权必究

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

修订历史记录

日期	备注	版本	作者
<2008-11-23>		<创建>	hardway
<2008-11-25>		TSF4G_TCONND_01_0001	hardway
<2008-12-22>		TSF4G_TCONND_01_0002	hardway
<2008-12-27>		TSF4G_TCONND_01_0003	hardway
<2009-01-06>		TSF4G_TCONND_01_0004	hardway
<2009-01-08>		TSF4G_TCONND_01_0005	hardway
<2009-02-17>		TSF4G_TCONND_01_0006	hardway
<2009-5-13>		TSF4G_TCONND_01_0007	hardway
<2009-9-30>		TSF4G_TCONND_01_0008	hardway
<2010-02-03>		TSF4G_TCONND_02_0000	hardway
<2010-06-09>		TSF4G_TCONND_02_0001	hardway

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

目录

1	引言	6
1.1	编写目的	6
1.2	定义	6
1.3	参考资料	6
2	安装配置	6
2.1	安装包	6
2.2	Tconnd 启动参数	7
2.3	日志说明	7
2.4	配置文件	7
3	配置说明	8
3.1	基本配置	8
3.2	DR 元数据描述库列表配置	8
3.3	协议数据单元列表配置	8
3.4	监听器配置列表	13
3.5	串行器（Serializer）配置列表	14
3.6	网络传输实例配置列表	14
4	运营注意事项	16
4.1	修改日志级别	16
4.2	修改共享内存大小	16
4.3	修改系统参数	16
4.3.1	进程允许打开最大连接数(必须)	16
4.3.2	修改内核未完成连接队列长度(推荐)	16
4.3.3	修改内核已完成连接队列长度(推荐)	17
5	Tconnd 版本变更	17
5.1	TSF4G_TCONND_02_0001	17
5.1.1	Bug 修正	17
5.1.2	配置变更	17
5.1.3	功能变更	17
5.1.4	接口变更	17
5.2	TSF4G_TCONND_02_0000	17
5.2.1	Bug 修正	17
5.2.2	配置变更	18
5.2.3	功能变更	18

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.2.4 接口变更	18
5.3 TSF4G_TCONND_01_0008	18
5.3.1 Bug 修正	18
5.3.2 配置变更	18
5.3.3 功能变更	18
5.3.4 接口变更	19
5.3.5 变更依赖	19
5.4 TSF4G_TCONND_01_0007	19
5.4.1 Bug 修正	19
5.4.2 配置变更	19
5.4.3 功能变更	19
5.4.4 接口变更	19
5.4.5 变更依赖	20
5.4.6 升级指导	20
5.5 TSF4G_TCONND_01_0006	20
5.5.1 Bug 修正	20
5.5.2 配置变更	20
5.5.3 功能变更	20
5.5.4 接口变更	20
5.5.5 变更依赖	20
5.6 TSF4G_TCONND_01_0005	20
5.6.1 配置变更	20
5.6.2 功能变更	21
5.6.3 接口变更	21
5.6.4 Bug 修正	21
5.6.5 变更依赖	21
5.7 TSF4G_TCONND_01_0004	21
5.7.1 配置变更	21
5.7.2 功能变更	21
5.7.3 接口变更	21
5.7.4 Bug 修正	21
5.7.5 变更依赖	21
5.8 TSF4G_TCONND_01_0003	21
5.8.1 配置变更	21
5.8.2 功能变更	22
5.8.3 接口变更	22
5.8.4 Bug 修正	22
5.8.5 变更依赖	22
5.9 TSF4G_TCONND_01_0002	22
5.9.1 配置变更	22
5.9.2 功能变更	22

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.9.3 接口变更	22
5.9.4 Bug 修正	22
5.9.5 变更依赖	22
5.10 TSF4G_TCONND_01_0001	22
5.10.1 配置变更	22
5.10.2 功能变更	22
5.10.3 接口变更	23
5.10.4 Bug 修正	23
5.10.5 变更依赖	23

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

1 引言

1.1 编写目的

本文档主要用于介绍 tconnd 使用配置,变更历史已经运营注意的配置事项,适用于 tconnd.使用维护者。

1.2 定义

- ◆ TCONND TSF4G connector daemon 服务器接入进程
- ◆ LogicSvr 逻辑服务器,tconnd 提供接入服务的对象。

1.3 参考资料

1. [TDR 参考文档](#)
2. [TBUS 参考文档](#)
3. [TAPP 参考文档](#)
4. [TCONND 概要设计文档](#)。

2 安装配置

2.1 安装包

tconnd 发布包是一个压缩文件 tsf4g_conndYYYYMMDD.tgz, 使用

```
tar -zxvf tsf4g_tconndYYYYMMDD.tgz
```

解开压缩包,得到 tconnd 目录,其子目录结构如下:

目录	说明	备注
./	安装包根目录	
sample/	例子程序	
sample2/	例子程序2	
docs/	文档	
include	tconnd API 头文件	
lib/	库文件	
win32_lib	windows库文件	
bin	tconnd可执行文件目录	
cfg	tconnd配置文件模板目录	
tools/	tconnd示例程序工具目录	

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

version	版本文件	
tsf4g_connd.install	linux安装脚本	
tsf4g_connd.dependency	linux安装依赖文件	
tsf4g_connd.bat	windows安装脚本	

2.2 Tconnd 启动参数

--id: tconnd 进程实体号，此 ID 将做为 tbus 系统中 tconnd 的地址。
--conf-file: 进程的配置文件路径，可以使用相对路径，--use-bus: 指示使用 tbus 组件。
--bus-key: tbus 系统路由表存储共享内存 key，缺省值是 1688。
--daemon, -D: specify whether start this process as a daemon。
--tlogconf: 进程日志配置文件路径。
start: 启动进程。
stop: 给已经启动的进程发送一个 SIGUSR1, 停止进程。
更多启动参数说明查阅 [TAPP 参考文档](#)。

Tconnd 启动:

```
./tconnd --id=6.1.1.1 --conf-file=../conf/tconnd.xml --bus-key=2009 --daemon start
```

Tconnd 的停止:

```
./tconnd --id=6.1.1.1 stop
```

2.3 日志说明

日志配置文件可以通过启动参数--tlogconf 指定,如果没有指定进程启动的日志配置文件,默认会在进程当前运行目录生成 tconnd_log.xml 日志配置,同时在/tmp 目录自动生成进程运行的日志文件。日志文件名一般格式为进程名_进程 ID.log 和进程名_进程 ID.error.如:

```
/tmp/tconnd_6.1.1.1.log    进程 trace 级别日志文件
/tmp/tconnd_6.1.1.1.error  进程 error 级别日志文件
/tmp/tconnd_6.1.1.1_rundata.log 进程运行统计数据
```

注意,一般在非正式运营环境下一般需要将日志配置文件的全局日志级别基本设为 DEBUG,正式环境下由于日志太多会影响性能,一般可以设为 INFO 级别。如图下,更多日志文件配置可以参考 [tlog 文档](#)



2.4 配置文件

tconnd.xml 是 tconnd 缺省的文件名，通过 tconnd 的命令行参数--conf-file，可以其内部公开

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

他名字的配置文件。

Tconnd 配置文件为 xml 格式的文件，其中的数据必须遵循 DR 约定。

2.5 示例程序

发布包的 Sample 目录是使用 tconnd QQ 统一登录方式通信所使用接口 API 示例。

发布包的 Sample2 目录是使用 tconnd 透传通信方式所使用接口 API 示例。

3 配置说明

3.1 基本配置

◆ Threading(缺省值为零):

指示是否按多线程模式启动，其值为 0 表示按照单线程模式启动，**目前不支持多线程, 不要修改此值。**

◆ EnableViewer(缺省值为零):

是否打开旁路功能，其值为 0，表示关闭旁路功能，否则打开旁路功能。所谓旁路监视是指 tconnd 收到客户端数据时，除了将数据放入与 logicsvr 通信 之间 tbus 通道外，还容许开一条旁路，tconnd 将数据放入此旁路中；

◆ MaxFD(缺省值为 10240):

tconnd 进程允许最大网络连接数；

配置实例如下：

```
<Threading>0 </Threading>
<EnableViewer>0 </EnableViewer>
<MaxFD>10240 </MaxFD>
```

3.2 DR 元数据描述库列表配置

此配置节点配置 tconnd 可能会用到的元数据描述库列表信息，每个元数据描述库信息保存在 TDRs 节点中，此节点的配置信息如下：

- Count(缺省值为 1):元数据描述库的个数
- Name(缺省值为 default)：元数据描述库配置节点的名字，tconnd 查找元数据库配置时用
- Path：元数据描述库保存路径；

配置实例如下：

```
<TDRList type="TDRList">
  <Count>1</Count>
  <TDRs type="TDR">
    <Name>default</Name>
    <Path>./sample.tdr</Path>
```


TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

</TDRs>

</TDRList>

3.3 协议数据单元列表配置

对应 logicsvr 协议消息包我们称之为协议数据单元（PDU，Proctocl Data Unit），每个协议数据单元配置保存在 PDU 节点下，此节点下的配置信息如下：

- ◆ Count(缺省为 1):PDU 单元的个数；
- ◆ Name(缺省值为 default)：协议数据单元配置名，可用于区分其他协议数据单元配置；
- ◆ UpSiz（缺省值为零）：上行数据包最大字节数,若取缺省值则不同解析器按不同的策略分配上行数据包缓存区,By_TDR 和 By_QQ 方式按照计算的协议单元元数据大小分配缓存。其它按照缺省大小(4096)分配,最好配置上行数据包大小。
- ◆ DownSize（缺省值为零）：下行数据包最大字节数,By_TDR 和 By_QQ 方式按照计算的协议单元元数据大小分配缓存。其它则按照缺省大小(65536)分配,
- ◆ LenParsertype(缺省值为 By_TDR 方式)：将数据流分解成数据包的解析器类型,目前支持 BY_TDR,BY_NULL,By_QQ,By_CHAR,By_STR,By_None6种解析器。

BY_TDR=0

BY_NULL=1

By_QQ=2

By_CHAR=3

By_STR=4

By_Non=10

BY_TDR 解析器是将协议数据单元的数据结构用 TDR 描述出来，通过 TDR 描述中描述的数据包长度信息进行分包，此解析器配置结构如下：

<LenParser type="PDULenParser">

<TDRParser type="PDULenTDRParser">

- ◆ <TDR>：此协议数据单元数据结构所在的[元数据描述配置节点](#)的名字。
- ◆ <Pkg>：此协议数据单元数据结构在元数据描述库中的描述名称
- ◆ <PkgLen>：PDU 数据结构中记录数据包总长度信息成员的元数据描述路径名,若该字段存在则下面 HeadLen 和 BodyLen 字段可以不填。
- ◆ <HeadLen>：PDU 数据结构中记录数据包头部长度信息的成员的元数据描述路径名；
- ◆ <BodyLen>：PDU 数据结构中记录数据包消息体长度信息的成员的元数据描述路径名；

</TDRParser>

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

</LenParser>

配置实例:

```
<PDUs type="PDU">
    <Count>1</Count>
    <Name>default </Name>
    <UpSize>0</UpSize>
    <DownSize>0</DownSize>
    <LenParsertype>PDULENPARSERID_BY_TDR </LenParsertype>
    <LenParser type="PDULenParser">
        <TDRParser type="PDULenTDRParser">
            <TDR>default </TDR>
            <Pkg>samplepdu</Pkg>
            <HeadLen>head.headlen</HeadLen>
            <BodyLen>head.bodylen</BodyLen>
        </TDRParser>
    </LenParser>
</PDUs>
```

BY_NULL 解析器是当数据中出现了 NULL (‘\0’) 字符就认为收到了一个完整的协议数据包。

配置实例:

```
<PDUs type="PDU">
    <Name>default </Name>
    <UpSize>0</UpSize>
    <DownSize>0</DownSize>
    <LenParsertype>PDULENPARSERID_BY_NULL</LenParsertype>
</PDUs>
```

BY_CHAR 解析器是当数据中出现了特定字符就认为收到了一个完整的协议数据包。
配置结构如下:

- ◆ Separator: 根据指定字符分包, 使用字符的 ASCII 码表示

配置实例: 根据‘\n’分包

```
<PDUs type="PDU">
    <Name>default </Name>
    <UpSize>0</UpSize>
    <DownSize>0</DownSize>
    <LenParsertype>PDULENPARSERID_BY_CHAR</LenParsertype>
    <LenParser type="PDULenParser">
        <PDULenCharParser>
```

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

```

        <Separator>10</Separator>
    </PDULenCharParser>
</LenParser>
</PDUs>

```

BY_STR 解析器是当数据中出现了特定字符串就认为收到了一个完整的协议数据包。配置结构如下：

- ◆ Separator: 字符数组, 使用字符的 ASCII 码表示

配置实例：根据"\r\n"分包

```

<PDUs type="PDU">
    <Name>default </Name>
    <UpSize>0</UpSize>
    <DownSize>0</DownSize>
    <LenParsertype>PDULENPARSERID_BY_STR</LenParsertype>
    <LenParser type="PDULenParser">
        <PDULenStringParser>
            <Separator>13 10</Separator>
        </PDULenStringParser>
    </LenParser>
</PDUs>

```

By_QQ 解析器提供一个预定义的消息协议, 在此消息协议的基础上, 可以运载应用自定义的消息。使用预定义的消息, 可以支持 QQ 号码方式的用户认证和加密, 解析器配置结构如下：

```

<LenParser type="PDULenParser">
    <PDULenQQParser>
        ◆ <TDR>: PDU 数据结构所在元数据描述库文件
        ◆ <SendPkg>: 上行数据包元数据名
        ◆ <RecvPkg>: 下行数据包元数据名。
        ◆ <SigValidSec>: 签名有效时间, 默认值为 7200 秒
        ◆ <CheckCltIP>: 是否验证 客户端 IP, 默认值为 0.
        ◆ <CheckSequence>: 验证上行包序列号是否递增, 默认为零, 不验证
        ◆ <EncMethod>: 客户端和 connd 之间加密方法 , 一般需要配为 3
            默认为 2.
            0: 表示消息体不加密
            1: 暂不支持
            2: QQ 方式, tea
            3: TAES

```

4: TAES2 安全算法

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

- ◆ <AuthType>: 验证方式(QQV1,QQv2),一般需要配为 3,
默认为统一签名方式。
 - 0:不验证签名
 - 1:0x37 签名方式,现在已不支持 0x37,如果配置了
1,验证方式也同 0x82.
 - 2:0x82 签名方式
 - 3:统一签名方式
- <CheckSig2>: 是否验证附加签名,默认为 0, 目前不需要。
- ◆ <SvrSKey>: 默认静态密钥(16 字节字符串,由即通分配)
- ◆ <SvrSKey2>: 密钥二(16 字节字符串,暂不用)
- ◆ <TestNumEnable>:是否支持测试号码,默认为 0.
- ◆ <TestSvrSKey>: 测试密钥一
- ◆ <TestSvrSKey2>: 测试密钥二,暂不用
- ◆ <TestBeginNum>:测试号码段开始号
- ◆ <TestEndNum>:测试号码段结束号
- ◆ <ThreewayHandshake>:是否需要三次握手,默认为 1,需要
- ◆ <UseVas>:对于正式运营必须置为一,接入公司 QQ 签名动态更改 agent,具体参考[官网](#)
- ◆ <Vasid>:动态索引 id, 由即通安全中心分配。
- ◆ <AppID> :业务 id,即通分配。单业务验证模式暂未用。
- ◆ <EnableMultiSvc>: 是否支持多业务验证,默认为零不使用
- ◆ <MultiSvrList>: 多业务验证结构列表
 - <SvcCount> :个数,默认为零
 - <SvcConf>: 结构如下
 - ◆ <AppID>:业务 id,即通分配。多业务验证模式必须配
 - ◆ <AuthType>:业务验证签名格式
 - ◆ <VasId>:动态索引 id, 由即通分配。
 - ◆ <SvrSKey>: 默认静态密钥(16 字节字符串,由即通分配)

</PDULenQQParser>

</LenParser>

配置实例:

<PDUs type="PDU">

<Count>1</Count>

<Name>default </Name>

<UpSize>0</UpSize>

<DownSize>0</DownSize>

<LenParsertype>PDULENPARSERID_BY_QQ </LenParsertype>

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

```

<LenParser type="PDULenParser">
    <QQParser type="PDULenQQParser">
        <TDR>default</TDR>
        <SendPkg>TPDUFrame</SendPkg>
        <RecvPkg>TPDUFrame</RecvPkg>
        <CheckCltIP> 0</CheckCltIP>
        <SigValidSec>1800</SigValidSec>
        <EncMethod>3</EncMethod>
        <AuthType>3</AuthType>
        <SvrSKey>aaaabbbbccccdddd</SvrSKey>
        <UseVas>1</UseVas>
        <VasId>28</VasId>
    </QQParser>
</LenParser>
</PDUs>

```

By_None 解析器收到数据直接转发,预留

3.4 监听器配置列表

ListenerList 节点下配置监听器列表, 每个监听器配置在 Listeners 节点下, 此节点配置信息如下:

- ✧ Count:监听器的个数,缺省值为 1。
- ✧ Name:监听器的名字,缺省值为 default,
- ✧ Url: 监听 URL, 在 URL 中指定监听地址, 端口等信息, 缺省值为 tcp://127.0.0.1:6666
- ✧ SendBuff: 发送缓冲区大小, 缺省值为 128k(131072).
- ✧ RecvBuff: 接收缓冲区大小,缺省值为 128k(131072).
- ✧ MaxIdle: 连接最大空闲时间, 时间单位为秒。tconnd 在这个时间间隔内没有收到客户端数据就断开客户端的连接,0 为不断开, 缺省值为 0.
- ✧ Backlog: 监听队列的大小,一般需要设为内核默认最大值 128.

配置实例如下:

```

<ListenerList type="ListenerList">
    <Count>1 </Count>
    <Listeners type="Listener">
        <Name>default</Name>
        <Url>tcp://127.0.0.1:6666</Url>
        <SendBuff> 131072</SendBuff>
        <RecvBuff> 131072</RecvBuff>
    </Listeners>
</ListenerList>

```

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

```

        <MaxIdle>0 </MaxIdle>
        <Backlog>128</Backlog>
    </Listeners>
</ListenerList>

```

3.5 串行器（Serializer）配置列表

Tconnd 将客户端的并发数据串行化放入 tbus 通道中，与 tconnd 通信的 tbus 通道对端进程就称为串行器（Serializer）。每个串行器配置在 Serializers 节点下，其配置信息结构如下：

- ✧ Count: 串行器的个数, 缺省值为 1.
- ✧ Name: 串行器的名字, 缺省值为 default
- ✧ Url: 串行器实体 ID, 此 ID 作为 tbus 通信地址;

配置实例

```

<SerializerList type="SerializerList">
    <Count>1 </Count>
    <Serializers type="Serializer">
        <Name>default</Name>
        <Url>10.1.2.1</Url>
    </Serializers>
</SerializerList>

```

3.6 网络传输实例配置列表

网络传输实例列表配置在 NetTransList 节点下，每个网络传输实例配置下 NetTrans 节点下，每个传输实例的信息结构如下：

- ✧ Name: 网络传输实例的名字;
- ✧ UseTimeStamp: 发送到 GameServer 数据包是否带时间戳，值为 0 时表示不带时间戳；否则接收数据包将附带接收时刻的时间戳；缺省值为 0；
- ✧ UseWaitQueue: 是否使用排队机制，0 表示客户端连接不排队, 默认为 1 使用排队机制。;
- ✧ QueueNotifyInterval: 排队通知时间间隔。UseWaitQueue 为 1 时有效, 默认为 1 秒。
- ✧ PDU: [协议数据单元配置](#) 节点的名称，通过此名称找到协议数据单元的具体配置，缺省值为 default
- ✧ Listener: [监听器配置节点](#) 的名称，通过此名称找到监听器的具体配置, 缺省值为 default。
- ✧ Serializer: [串行器配置节点](#) 的名称，通过名称找到串行器具体配置，缺省值为 default。
- ✧ LisViewer: Listener 旁路接收器的名称，如果配了此配置项，且旁路功能打开后，

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

tconnd 除了将收到的客户端消息发送给 Serializer 外，还会发送给 LisViewer。

✧ SerViewer: Serializer 旁路接收器名称，如果配了此配置项，且旁路功能打开后，tconnd 从 Serializer 中取出数据发给客户端后，还会发一份给 SerViewer;

✧ ConnLimit:连接限制结构,结构如下;

<MaxConn>: 网络实例允许最大连接数,默认为零,允许连接数等于进程允许最大网络连接数。

<Permit>: gameserver 能处理最大连接数，缺省 0 按 tconnd 允许最大连接数处理并且连接不需要排队。

<Speed>:每 100 毫秒从队列放行连接数，缺省 0,放行当前队列剩余连接数。

✧ TransLimit:传输限制结构,结构如下;

<PkgSpeed>: 最大收包速度(单位:包/秒,0 表示不限制),缺省值为 0

<ByteSpeed>: 最大收字节流速度(单位:字节/秒,0 表示不限制),缺省值为 0

<LimitAction>: 超过客户端限速是否断开连接，PkgSpeed 和 ByteSpeed 不为零是有效,默认为 1 断开连接,0 不断开只是丢包。

配置实例:

```
<NetTransList type="NetTransList">
  <Count> 1</Count>
  <NetTrans type="NetTrans">
    <Name>default</Name>
    <UseTimeStamp>0 </UseTimeStamp>
    <UseWaitQueue>1 </UseWaitQueue>
    <QueueNotifyInterval>1 </QueueNotifyInterval>
    <PDU>default</PDU>
    <Listener>default</Listener>
    <Serializer>default</Serializer>
    <ConnLimit>
      <MaxConn>0</MaxConn>
      <Permit>0</Permit>
      <Speed>0</Speed>
    </ConnLimit>
    <TransLimit>
      <PkgSpeed>0</PkgSpeed>
      <ByteSpeed>0</ByteSpeed>
      <LimitAction>0</LimitAction>
    </TransLimit>
  </NetTrans>
</NetTransList>
```

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

</NetTransList>

4 运营注意事项

4.1 修改共享内存大小

正式运营一般需要修改接入进程和逻辑进程间的共享内存大小,一般设置不低于 20M,使用 tbusmgr 修改共享内存大小方法如下:

- 删除内存
./tbusmgr -k 2010 -D 1
- 重写内存
./tbusmgr -C tbusmgr.xml.cus -W
- 查看共享内存大小
./tbusmgr -S 10.2.1.1 -k 2010
- 获得帮助
./tbusmgr -h

4.2 修改系统参数

4.2.1 进程允许打开最大连接数(必须)

接入进程默认支持进程打开最大连接数为 10240,需要将系统设置的进程允许打开最大连接数设置为 10240 以上.

- 查看允许打开文件描述符
ulimit -n
- 修改允许打开文件描述符
使用 root 权限
1: 在/etc/profile 中加上 ulimit -n 10240
2: 执行: . profile
3: 修改生效:service ssh2 restart

4.2.2 修改内核未完成连接队列长度(推荐)

- 查看
sysctl -a|grep tcp_max_syn_backlog (默认 1024)
- 修改
echo "10240" > /proc/sys/net/ipv4/tcp_max_syn_backlog

4.2.3 修改内核已完成连接队列长度(推荐)

- 查看

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

sysctl -a|grep net.core.somaxconn (默认 128)

➤ 修改

echo "10240" > /proc/sys/net/core/somaxconn

5 Tconnd 版本变更

5.1 TSF4G_TCONND_02_0001

5.1.1 Bug 修正

- 修正 tconnd2.0版本在处理低版本客户端验证方式跟服务器不一致的情况下记录错误日志会 core 的问题。
- 修正 tconnd2.0版本在 AES 算法返回负值可能导致 TCONND core 的问题。

5.1.2 配置变更

- N/A.

5.1.3 功能变更

- tconnd 支持 linux 64 位版本。
- tconnd 支持 reload 静态配置数据。
- tconnd 支持控制通道模式查看和修改内部数据。
- tconnd 增加连接释放原因统计(QQ 仙侠)。
- tconnd 检查上行包序号不是递增断开连接(幻想世界)。
- tconnd 新增了 TAES2 算法支持。
- tconnd 新增了不排队支持。

5.1.4 接口变更

接口头文件和库依赖 tsf4g_base_02_0001 以上版本。

- server 端:
N/A
- client 端
TQQAPI 新增了 tqqapi_set_log | tqqapi_set_log_callback 日志接口

5.2 TSF4G_TCONND_02_0000

5.2.1 Bug 修正

- 修正跳转路由后排队统计不正确的问题
- 修正 udp 收发包异常后,收包不正确的问题
- 修正了文本消息在粘包的情况下分包不正确的问题

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.2.2 配置变更

- 增加了按特定字符分包和特定字符串分包的配置.
- 增强了日志记录,支持日志按照 uin 过滤
- 增强了 BUS 来源检查,只支持接收配置在 BUS 列表的下行包。
- 增强了连接统计数据。

5.2.3 功能变更

- tconnd 与服务器通信增加心跳检测协议
- tconnd 增加按特定字符分包和特定字符串分包功能。
- tconnd 增强了内部连接统计数据。
- tconnd 增加了多业务签名验证功能。

5.2.4 接口变更

- server 端:
 - 1:新增了心跳检测协议。
 - 2:新增了获取连接关闭原因的 API tconnapi_get_closestring
- client 端
 - 1:tqqapi_create_initial_connection 接口增加了业务 Service ID 字段,tconnd 兼容 1.8 版本接口。

5.3 TSF4G_TCONND_01_0008

5.3.1 Bug 修正

- 修正 tmempool 的 idx 负值 core 的 bug。
- 修正跳转路由上限的问题。
- 修正了排队协议可能出现负数的问题

5.3.2 配置变更

- TransLimit 结构增加客户端上行包超过连接限制是否断开连接开关,默认为零,不断开
- PDULenQQParser 结构增加 CheckSequence 开关,对应是否验证上行包序列号递增,默认为零,不验证.

5.3.3 功能变更

- tconnd 支持 AES 通信加密算法
- tconnd 支持明文通信,即不加密(测试)
- tconnd 增加调整服务器连接限制接口(轩辕需求)
- 增加客户端超过连接限制断开连接开关(幻想世界需求)

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.3.4 接口变更

➤ server 端:

- 1:新增了跳转连接限制接口(adjustconlimit) (轩辕需求)
- 2:Stop 消息 reason 增加了客户端超过连接的一个宏

➤ client 端

1.8 版本之后客户端 API,建议项目组直接使用基础开发组 GamesvrManage 组件,不再单独对项目组提供接口支持。

- 1:tqqapi_create_initial_connection 接口增加了加密方法字段
- 2:tqqapi_create_relay_connection 接口增加了加密方法字段
- 3:tqqapi_set_gamekey 接口增加了加密方法字段
- 4:新增了 tqqapi_write_msg/tqqapi_write_buffer 接口
- 5:新增了 tqqapi_read_msg/tqqapi_read_buffer 接口

5.3.5 变更依赖

tconnd 客户端 API 依赖于 tsf4g_base_01_0010 版本以上的基本库文件.包括 libpal, libtdr,libtsec 等

tconnd 服务端 API 依赖于 tsf4g_base_01_0007 版本以上的基本库文件.包括 libpal, libtdr*和 libtbus*等

5.4 TSF4G_TCONND_01_0007

5.4.1 Bug 修正

N/A

5.4.2 配置变更

- 新增了动态签名和测试号码配置
- 新增了 QQ 签名验证不三次握手流程的配置。

5.4.3 功能变更

- tconnd 登录验证流程支持不三次握手的流程
- tconnd 支持组播包不验证包头连接信息(幻想世界需求)
- tconnd 支持路由跳转及预占协议带路由(轩辕需求)

5.4.4 接口变更

➤ client 端:

- 1:新增了 tqqapi_send_buffer 和 tqqapi_recv_buffer(仙侠传需求)

➤ server 端:

- 1:连接建立包(start)增加了连接等待时间字段
- 2:新增了路由跳转协议(setrouting)
- 3:预占协议支持带路由(notify)

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.4.5 变更依赖

tconnd 客户端 API 依赖于 tsf4g_base_01_0007 版本以上的基本库文件.包括 libpal, libtdr 等
tconnd 服务端 API 依赖于 tsf4g_base_01_0007 版本以上的基本库文件.包括 libpal, libtdr*和 libtbus*等

5.4.6 升级指导

➤ client 端

与上一个版本相比,协议和接口没有变化,不需要替换和编译库

➤ server 端

与上一个版本相比,start 和 notify 协议有增加字段,同时增加了 setrouting 协议,需要重新编译 tconnapi.a 库.

5.5 TSF4G_TCONND_01_0006

5.5.1 Bug 修正

N/A

5.5.2 配置变更

N/A

5.5.3 功能变更

- tconnd 登录验证流程增强.需要与客户端三次握手才能建立连接
- tconnd 消息处理性能优化.

5.5.4 接口变更

- client 端:tqqapi_create_initial_connection 接口有变化
- client 端:tqqapi_create_relay_connection 接口有变化
- client 端:增加了服务器断开连接通知消息
- client 端:增加了 tqqapi_get_errorstring 接口
- server 端:重连协议增加了重连类型定义

5.5.5 变更依赖

tconnd 客户端 API 依赖于 tsf4g_base_01_0007 版本以上的基本库文件.包括 libpal, libtdr 等
tconnd 服务端 API 依赖于 tsf4g_base_01_0007 版本以上的基本库文件.包括 libpal, libtdr*和 libtbus*等

5.6 TSF4G_TCONND_01_0005

5.6.1 配置变更

N/A

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.6.2 功能变更

N/A

5.6.3 接口变更

- 修改 TPDUHEAD 结构里 TPDU_MAX_ENCRYPTIDENT_LEN 的宏定义.EncryptIdent 成员 buffer 的长度有变化。
- 新增了 tconnd client API 两个接口,封装了 tconnd 握手流程。
tqqapi_create_initial_connection 通过签名信息建立握手连接。
tqqapi_create_relay_connection 通过重连.信息建立握手连接。

5.6.4 Bug 修正

- 修改了 tconnd 收到 logicserver 消息出 core 的 bug

5.6.5 变更依赖

Tconnapi/Tcltapi:依赖于 tsf4g_base_01_0005 版本以上的基本库文件.包括 libtdr*和 libtbus*等

5.7 TSF4G_TCONND_01_0004

5.7.1 配置变更

N/A

5.7.2 功能变更

N/A

5.7.3 接口变更

- 新增了 tconnd client API 两个接口,并增强了 tqqapi_recv_msg 接口,client 可以忽略 tconnd 握手和更改密钥流程,直接发送和接收消息。
tqqapi_build_auth_msg:生成签名信息,用于测试号码连接验证。
tqqapi_build_relay_msg:生成重连信息,用于跳转或断线重连。

5.7.4 Bug 修正

N/A

5.7.5 变更依赖

Tconnapi/Tcltapi:依赖于 tsf4g_base_01_0005 版本以上的基本库文件.包括 libtdr*和 libtbus*等

5.8 TSF4G_TCONND_01_0003

5.8.1 配置变更

N/A

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

5.8.2 功能变更

- 修正了新签名格式的加解密算法

5.8.3 接口变更

- 新增了 tconnd client API

5.8.4 Bug 修正

N/A

5.8.5 变更依赖

Tconnapi/Tcltapi:依赖于 tsf4g_base_01_0001 版本以上的基本库文件.包括 libtdr*和 libtbus*等

5.9 TSF4G_TCONND_01_0002

5.9.1 配置变更

- QQParser 类型增加了是否支持测试号码的开关,增加了对应签名是否验证第二段加密结构的开关

5.9.2 功能变更

- 支持新的 QQ 统一登录签名验证
- 重连协议增强,支持跨服跳转和断线重连

5.9.3 接口变更

- 签名验证请求消息有变化

5.9.4 Bug 修正

N/A

5.9.5 变更依赖

Tconnapi/Tcltapi:依赖于 tsf4g_base_01_0001 版本以上的基本库文件.包括 libtdr*和 libtbus*等

5.10 TSF4G_TCONND_01_0001

5.10.1 配置变更

- NetTrans 数据定义增加了 connlimit 和 TransLimit 成员,
- PDULenParser 类型增加了 PDULenQQParser 类型的定义
- PDU 数据增加了上下行数据包的定义

5.10.2 功能变更

- 对客户端提供了预定义的消息协议,使用预定义的消息,可以支持 QQ 号码方式的用户认内部公开

TSF4G-TCONND	Version: <1.00>
配置管理系统概要设计说明书	Date: <2008-11-23>
<互娱研发中心架构组>	

证和加密,在此消息协议的基础上,支持断线重连,排队通知功能,同时可以运载应用自定义的消息。

- 新增了流量控制功能
- 新增了连接排队功能
- 新增了运行时日志和统计日志
- 新增了测试号码段支持。

5.10.3 接口变更

N/A

5.10.4 Bug 修正

N/A

5.10.5 变更依赖

Tconnapi/Tcltapi:依赖于 tsf4g_base_01_0001 版本以上的基本库文件.包括 libtdr*和 libtbus*等