中国象棋对弈软件网络传输协议

CCNI

http://code.google.com/p/ccniserver/

修改记录

版本号	描 述	作者	Email	Date
0.10	创建	bjwf	bjwf2000@gmail.com	2007-2-14
0.11	更新，增加第 5 章	bjwf	bjwf2000@gmail.com	2007-2-26
0.12	更新多数据包，增加消息示例	bjwf	bjwf2000@gmail.com	2007-3-12
0.13	增加局域网支持	bjwf	bjwf2000@gmail.com	2007-4-3
0.14	增加注册消息/增加积分系统	bjwf	bjwf2000@gmail.com	2007-4-6
0.20	修改状态机，增加房间相关消息，扩展修改传输层协议为 TCP 和 UDP,并增加握手过程。	bjwf	bjwf2000@gmail.com	2008-11-20

Open Issues:

. 关于 LAN 支持，需要再考虑考虑.
. 需要设计一个新的用户等级/积分系统.

目录

1 导言

1.1 概述

本协议的主要设计目标是给中象棋软引擎提供一个网络对战环境的标准，使不同的引擎可登录对战服务器上自动搜索合适的对手软件，自动设定对局参数，自动对奕，以实方便引擎检测棋力，调整算法参数，调试代码或通过网络在进行比赛。
本协议的主要内容诣在提供规范化的象棋网络对战服务器以及客户端的行为和通讯协议。
本协义主要注重于通过网络（局域网/Internet）机机的自动对奕模式，同时兼顾人机，人人对奕模式。

1.2 参考文献

[1] IGS Commands http://www.pandanet.co.jp/English/command.html

[2] 中国象棋通用引擎协议 http://www.elephantbase.net/protocol/cchess_ucci.htm

[3] 中国象棋竞赛规则 1999 年版 http://www.elephantbase.net/protocol/rule.htm

[4] EOL 等级分计算公式 http://www.elephantbase.net/protocol/elostat.htm

[5] IGS Pandanet Rating System http://www.pandanet.co.jp/English/ratingsystem/

[6] Universal Chess Interface http://www.chessbase.com/download/engines/uci/uci-protocol.rtf

[7]

1.3 述语表

CCNI -- 中国象棋对弈软件网络传输协议

XML --

服务器

客户端

2 需求描述

2.1 实现客户端/服务器方式下的棋软自动对奕

登录服务器
获取房间列表
进入退出房间
获取房间中的用户列表
建立对局会话(指定规则，时间，等)
获取已建立的对局会话列表
加入已有的对局会话
退出对局会话
加入对局会话中观战
用户认证，用户等级分记录(*)

2.2 实现局域网内客户端之间点对点的棋软自动对奕

发起方做为局域网主机
加入方与指定 IP 或主机建立一个会话

协商规则，时间，对局数，开始状态（实现让子，残局等）
自动开始
支持第三方观战

3 CCNI 协议

3.1 用户状态机

服务器为每个用户（客户端）维护一个状态机，不同状态下用户可进行不同的操作。图 3.1.1 显示了状态机的状态转换图:

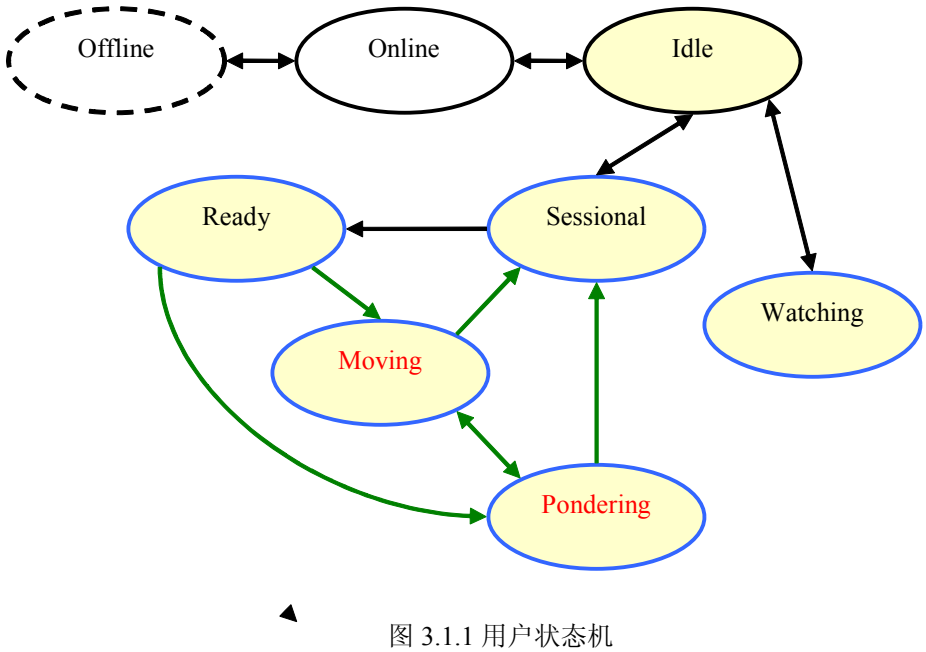


图 3.1.1 用户状态机

注：图中所有实线框表示在线，黄色底色表示已进入房间在线状态，蓝色边框表示已加入对局会话，红色字体表示已开始对局。黑色线表示状态迁移由客户端发起，绿色表示由服务器发起。

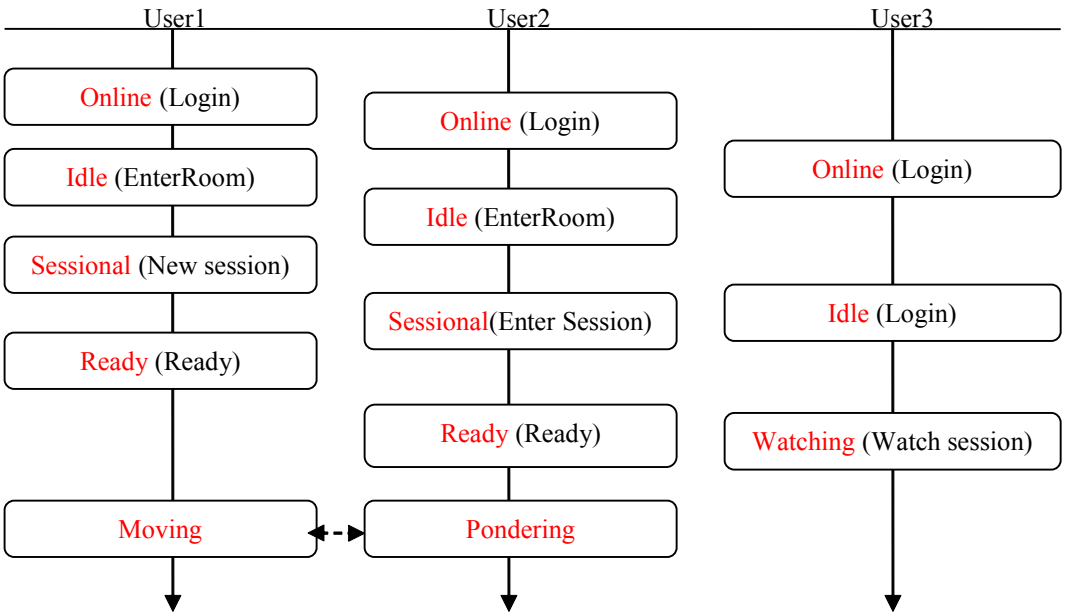
状态	说明
Offline	离线
Online	在线
Idle	在线，已进入房间
Sessional	在线，已进入房间，加入对局会话，空闲（等待对手或协商规则）
Ready	在线，已进入房间，已加入对局会话，已准备好
Watching	在线，已进入房间，已加入对局会话，观战
Moving	在线，已进入房间，已加入对局会话，已开始棋局，走棋
Pondering	在线，已进入房间，已加入对局会话，已开始棋局，等待对方走棋

状态转换说明：
Offline 可以通过 Login 的方法转成 Online.
所有在线状态(实线框)都可能由于断线而转成 Offline 状态(为方便画图，图中并未标出这些线)。
用户通过进入，退出房间的方法大 Online 和 Idle 之间转换。
用户可以通过创建或加入对局会话把状态从 Idle 转到 Sessional。

所有状态由服务器端进行维护，但客户端应该记录并确切的知道自己的当前状态。

3.2 用户状态转换时序图

客户端登录服务器后进行对局的流程及状态转换示意图:



User1 登录服务器，进入 Online 状态，进入房间后转为 Idle 状态，创建对局会话后进入 Sessional 状态，准备好后发送 Ready 消息进入 Ready 状态，User1 进入 Ready 状态后，对局是否真正开始，取决于是否有对手加入对局会话并是否设置 Ready 状态，一旦 User2 加入，并设置 Ready 之后，对局开始。User1 和 User2 分别转入 Moving 或 Pondering 状态（取决于对局会话的设置）。

User3 可以在对局会话被创建之后到会话结束之前的任何时间以观战者身份加入会话。

3.3 数据转输协议

客户端与服务器通过网络通讯协议进行交互，每次交互产生一个或多个数据包。每个数据包描述一个或多个客户端请求或一个或多个服务器回应。CCNI 协议数据包由两层协议组成，下层是用二进制表达的传输协议，由包头和数据段组成，上层为 XML 表达的文本协议。

CCNI 同时使用 TCP 协议和 UDP 协议做为下层的网络通讯协议。对客户端主动发起的请求和回复基于 TCP 协议转输，服务器主动发出的通知消息采用 UDP 协议传输。

基于 TCP 协议的数据包使用的缺省端口号为 20081, 基于 UDP 协议的数据包使用的缺省端口号为 20082。

3.3.1 数据包格式

数据包由两部分组成，一是包头信息，另一部分是数据信息，其中包头用二进制描述，数据信息根据包头的不同可以是 XML 描述的 CCNI 消息或其它格式的数据。表 3.3.1 描述了包头定义。

定义	标识符	长度(字节)	说明
包头长度	hdlen	1	数据包包头的长度，值为 32
版本号	ver_major	1	主版本号，值为 1
副版本号	ver_minor	1	副版本号，值为 0
数据包类型	type	1	数据段的类刑，0 表示连接请求，1 表示数据段用 xml 表示的正常数据包。
数据长度	data_len	2	包头后面的数据信息的长度，最大为 64K
数据包序列号	seq	4	本数据包的序列号
用户数据	udata	4	用户数据
数据包 Key	secret	16	用于加密身份验证的 Key
保留	reserved	2	保留，值为 0
			共 32 字节

其中数据包序列号，用户数据，保留这几个字段在客户端与服务器的一次对话过程中保持不变，即服务器会保证回应数据包的这几个字段与客户端发送请求时的这几个字段一致。 数据包序列号用于需要同步回应的数据包时实现回应数据包和请求数据包的配对，请参考消息的详细描述。

3.3.2 消息格式

数据包中包头后面的数据信息是使用 XML 描述的 CCNI 消息，一个消息数据包中可以包含一个或多个 CCNI 消息，一个 CCNI 消息具有如下的格式:

```
<消息标识符>
    <消息参数1>参数1</消息参数1>
    <消息参数2>参数2</消息参数2>
    <消息描述>这是一个CCNI消息</消息描述>
</消息标识符>
```

一个完整的消息数据包可包含一个或多个 CCNI 消息，具有如下的格式：

```
<?xml version="1.0" encode="UTF-8"?>
<CCNIMessages>
  <消息1>
    <消息参数1>参数1</消息参数1>
    <消息参数2>参数2</消息参数2>
    <消息描述>这是一个CCNI消息</消息描述>
  </消息1>
  <消息2>
    <消息参数1>参数1</消息参数1>
    <消息参数2>参数2</消息参数2>
    <消息描述>这是另一个CCNI消息</消息描述>
   </消息2>
</CCNIMessages>
```

注：一个消息数据包的最大长度被限制在 64 K 字节以内。客户端可以在一个数据包里发送多个消息，服务器依次处理每个请求的消息，并将所有应答消息在一个消息包中返回。（如回应数据包大于 64K，超长的问部分可能被丢掉，所以客户端应避免在一个消息数据包里发送过多的 CCNI 消息。）

所有的消息据包由< CCNIMessages >开头，以</CCNIMessages >结束，其中包括一个或多个 CCNI 消息。每个 CCNI 消息包括两部分，消息主体和消息描述。消息主体说明本消息的请求的功能及所需的参数，消息描述可有可无，用于对当前消息进行文字说明，接收方可忽略或用于调试参考。以下是一个消息数据包只包含一条登录消息的例子：

```
<?xml version="1.0" encode="UTF-8"?>
<CCNIMessages>
  <Login>
    <UserName>bjwf</UserName>
    <Password>12345</Password>
    <Description>login request</Description>
  </Login>
</CCNIMessages>
```

本例子中的<Login>与</Login>之间的内容为消息主体，<Login>称为消息的标识，CNNI 所有支持的消息标识参见后面的消息定义。

注：消息中所有 XML 标识均严格区分大小写。非 ASC 字符用 UTF8 编码。

3.3.3 消息类型及流向

本协议中所有的消息分为三类，一是客户端请求消息，如登录，加入对局会话等，另一是服务器针对客户端请求的回应消息，第三类是服务器主动向客户端发出的通知消息。

客户端的请求消息又分两种，一种是需要服务器的同步返回，另一种是不需要。所谓需要同步返回是指客户的请求发出后，需要等服务器的回应消息之后才能继续运行，等待过程中客户端发送线程将被挂起。回应数据包中的包头部分的序列号，用户数据，保留字段与请求数据包的包头保持相同。回应消息的标识为相应的请求消息的标识加后缀 Res，例为下面为 Login 的回应消息:

```
<CCNIMessages>
  <LoginRes>
    <ReturnCode> -1</ReturnCode>
    <ReturnInfo>unknown user name or password error</ReturnInfo>
    <Description>login error </Description>
  </LoginRes>
</CCNIMessages>
```

所有的通知消息（前缀为 Notify）将通过 UDP 传输。

3.4 客户端和服务器建立连接

出于性能和安全的考虑，CCNI 在下层同时采用 TCP 和 UDP 作为传输层协议，在一次完整的客户端登录到退出的流程中，有些数据包是通过 TCP 传输，有些是通过 UDP 传输，这就要求在客户端与服务器建立连接时要建立两个 socket，一个用于 TCP，一个用于 UDP。所有的连接均由客户端发起，完成一个连接的过程如下：

1. 客户端通过 UDP 向服务器发送请求连接数据包，数据包类型 0,只有包头，无数据。
2. 服务器通过 UDP 向客户端回应允许连接，并返回一个 Key 和有效期，secret 字段包括 Key, Reserved 字段表示有效期，单位为秒。
3. 客户端向服务器发起 TCP 连接，连接成功后必须在有效期之内向服务器发送 Login 消息，并在包头中包括刚才得到的 Key，否则需要重新申请 Key。

3.5 数据加密和安全

[TBD]

3.6 局域网支持

CCNI 协议的数据传输协议同时适用于 Internet 的 Server/Client 模式和局域网的点对点格式。大部分 CCNI 消息的定义也同时适用于这两种情况，第 4 章的消息定义中会列出有区别的地方。

在局域网连线过程中，连线的发起方应建立后台运行的小型 CCNI 网络服务，监听并响应其它客户端的连接请求，同时发起方也应做为一个客户端连接到自己建立的 CCNI 服务器上。这个小型的 CCNI 服务器我们称为 CCNIServerLite。

CCNIServerLite 所支持的 CCNI 消息及流程与完整的 CCNI Server 有所区别，主要表现在：

- 仅支持一个 Session，只能是发起方创建。
- 不支持数据库，对于客户端的 Login 请求，仅认为是用户向自己传递相关的用户信息，对于所有的 Login 请求仅保证用户名唯一。
- 支持的在线人数受限于单个 Session 的容量。
- 发起方是 Session 的拥有者，发起方退出后，所有连线的人都断开连接。

4 CCNI 消息定义

4.1 获取服务器信息(CCNI)

➢ **消息标识：**

CCNI

➢ **流向：**

C=>S。

➢ **说明：**

获取服务器信息。<SeriverInformation>中的Type字段说明服务器的类型，目前有两类，CCNI Server 和 CCNIServerLite，其中CCNIServerLite说明该server是局域网连线的发起方。

➢ **发送端状态要求：**

登录后所有的状态

➢ **消息参数**

无

➢ **接收方动作**

➢ **消息示例**

```
<CCNI/>
```

➢ **回应示例**

```
<CCNIRes>
  <SeriverInformation>
    <ServerType>CCNIServer</ServerType>
    <ServerVersion>1.0</ServerVersion>
    <CCNIVersion>1.0</CCNIVersion>
    <Copyright>bjwf Copyright@2006-2010</Copyright>
```

```
    <Author>bjwf bjwf2000@gmail.com</Author>
    <Description>CCNI Chinese Chess network server version 0.1 for Windows NT</Description>
  </SeriverInformation>
</CCNIRes>
```

4.2 注册用户(Register)

➢ **消息标识:**
Register

➢ **流向:**
C=>S。

➢ **说明:**
在服务器上注册新用户。

➢ **发送端状态要求:**
Offline

➢ **消息参数**
UserName 用户名，字母开头，只能是字母，数字，下划线, 长度 8-20
NickName 昵称, 可以是中/英文/数字/字母
Email 电子邮件
Password 密码， 客户端应负责把密码用 MD5 方式加密,CCNI 消息中所有出现的 password 都应以 MD5 方式加密

➢ **接收方动作**
检查用户名是否唯一，并返回注册是否成功

➢ **消息示例**

```
<Register>
   <UserName >bjwf</UserName >
   <NickName>wfwf</NickName>
   <Email>bjwf2000@gmail.com</Email>
   <Password>827ccb0eea8a706c4c34a16891f84e7b</Password>
   <Level>4</Level>
</Register>
```

➢ **回应示例**

```
<CCNIRes>
  <RegisterRes>
    <ReturnCode> -1</ReturnCode>
    <ReturnInfo>user name already been used</ReturnInfo> //可选字段
  </RegisterRes>
</CCNIRes>
```

4.3 用户登录(Login)

➢ **消息标识:**
Login

➢ **流向:**
C=>S (客户端发送，服务器接收)

➢ **说明:**

用户登录。

➢ **发送端状态要求:**

Offline

➢ **消息参数**

UserName 用户名
Password 密码

➢ **接收方动作**

向发送方发送应答消息，表示登录是否成功，如成功则把当前用户转为 Idle 状态并向其它所有在线用户发送 NotifyLogin 的通知消息。如果成功返回代码为 0, 如果出错则返回代码小于 0，可能的出错代码有：
-1 用户名不存在或密码错误
-2 用户已经登录(状态错)
如果登录成功，应答消息中还包括当前用户的描述信息，目前用户的描述信息格式如下(有待于扩展)：

```
<User>
<UserName >bjwf</UserName >
<NickName>wfwf</NickName>
<Score>2000</Score>
<Level>4</Level>
</User>
```

对于 SeverLite 来讲，不检查用户名和密码，只要不存在同名的用户，则一定返回成功。

➢ **消息示例**

```
 <Login>
    <UserName>bjwf</UserName>
    <Password>12345</Password>
    <Description>login request</Description>
  </Login>
```

➢ **回应示例**

```
<LoginRes>
    <ReturnCode> -1</ReturnCode>
    <ReturnInfo>unknown user name or password error</ReturnInfo> //可选字段
    <Description>login error </Description>      //可选字段
</LoginRes>
```

当用户成功登录时回应如下消息:

```
<LoginRes>
    <ReturnCode> 0 </ReturnCode>
    <User>
       <UserName >bjwf</UserName >
       <NickName>wfwf</NickName>
       <Score>2000</Score>
       <Level>4</Level>
    </User>
    <Description>login success<Description>      //此字段可选
</LoginRes>
```

4.4 用户退出登录(Logout)

➢ **消息标识:**
Logout

➢ **流向:**
C=>S (客户端发送，服务器接收)。

➢ **说明:**
用户退出登录。

➢ **发送端状态要求:**
除 Offline 之外所有的状态。

➢ **消息参数**
UserName 用户名
Password 密码

➢ **接收方动作**
向发送方发送应答消息，表示退出登录是否成功，如果用户是从房间中直接退出，则服务器向发房间中其它用户广播退出房间消息。

➢ **消息示例**

```
<Logout>
  <UserName>bjwf</UserName>
  <Password>12345</Password>
  <Description>logout request</Description>
</Logout>
```

➢ **回应示例**

```
<LogoutRes>
    <ReturnCode>0</ReturnCode>
    <ReturnInfo>user logout</ReturnInfo> //可选字段
    <Description>logout success</Description>      //可选字段
</LogoutRes>
```

4.5 用户进入房间(EnterRoom)

➢ **消息标识:**
EngerRoom

➢ **流向:**
C=>S (客户端发送，服务器接收)

➢ **说明:**
用户进入房间。

➢ **发送端状态要求:**
Online

➢ **消息参数**
RoomNumber 房间编号

➢ **接收方动作**

向发送方发送应答消息，表示进入房间是否成功，如成功则把当前用户转为 Idle 状态并向房间其它所有在线用户发送 NotifyEnterRoom 的通知消息。如果成功返回代码为 0, 如果出错则返回代码小于 0，可能的出错代码有：

-1 用户名不存在或密码错误

-2 用户已经登录(状态错)

如果进入房间成功，应答消息中还包括当房间的描述信息，目前房间的描述信息格式如下 (有待于扩展)：

```
<Room>
 <ID>1</ID>
 <Name>room1</Name>
 <Description>room description</Description>
 <DefaultSession>
    <!—session descriptions-->
 </DefaultSession>
</Room>
```

对于 SeverLite 来讲，不检查用户名和密码，只要不存在同名的用户，则一定返回成功。

➢ **消息示例**

```
 <EnterRoom>
    <RoomID>10</RoomID>
    <Description>enter room</Description>
  </ReterRoom>
```

➢ **回应示例**

```
< EnterRoomRes>
    <ReturnCode> -1</ReturnCode>
    <ReturnInfo>unknown roomid</ReturnInfo> //可选字段
</ EnterRoomRes>
```

4.6 用户退出房间(LeaveRoom)

➢ **消息标识:**

LeaveRoom

➢ **流向:**

C=>S (客户端发送，服务器接收)。

➢ **说明:**

用户退出房间。

➢ **发送端状态要求:**

Idle

➢ **消息参数**

➢ **接收方动作**

向发送方发送应答消息，表示退出房间是否成功，如成功则把当前用户转为 offline 状态，并向房间中其它用户广播 NotifyLeaveRoom 的消息。

➢ **消息示例**

```
  <LeaveRoom>
  </LeaveRoom>
```

➢ **回应示例**

```
<LeaveRoomRes>
    <ReturnCode>0</ReturnCode>
    <ReturnInfo>user leave room</ReturnInfo> //可选字段
    <Description>leave room success</Description>      //可选字段
</LeaveRoomRes>
```

4.7 通知进入房间(NotifyEnterRoom)

➢ **消息标识:**

NotifyEnterRoom

➢ **流向:**

S=>C

➢ **说明:**

服务器向某房间内用户通知某个用户进入。

➢ **发送端状态要求:**

➢ **消息参数**

UserName 用户的用户名
NickName 用户的昵称

➢ **接收方动作**

刷新用户列表

➢ **消息示例**

TBD

➢ **回应示例**

无

4.8 通知退出房间(NotifyLeaveRoom)

➢ **消息标识:**

NotifyLeaveRoom

➢ **流向:**

S=>C

➢ **说明:**

服务器向某房间内用户通知某个用户离开。

➢ **发送端状态要求:**

➢ **消息参数**

UserName 用户的用户名
NickName 用户的昵称

➢ **接收方动作**

刷新用户列表

➢ **消息示例**

TBD

➢ **回应示例**

无

4.9 创建对局会话(NewSession)

➢ **消息标识:**

NewSession

➢ **流向:**

C=>S (客户端发送，服务器接收)。

➢ **说明:**

用户请求创建一个对局会话。

➢ **发送端状态要求:**

Idel

➢ **消息参数**

对局会话的描述信息，用 SessionDescription 标识，其子字段如下（有待扩展）：

Title	本会话的标题
Type	会话的类型(比赛/训练)只有比赛类型的棋局才更新记录等级分。
Fen	本会话中每局棋的初始棋盘状态，可选参数，缺省为开局
Red	持红棋的一方，可选参数，缺省为空,表示先加入会话的一方(创建者)
Innings	期望的对局数，可选参数，缺省为 0,表示无数局，直到某一方退出
Rule	规则，可选参数，缺省为亚洲规则 *
TotalTime	一局的总时间，单位为秒，可选参数，缺省为 900(15 分钟)
StepTime	每步的时间，单位为秒，可选参数，缺省为 60(1 分钟)
BonusPerStep	每步加的时间,单位为秒，可选参数，缺省为 0
RequiredScore	加入会话所需的等级分，可选参数，缺省为 0
Ante	本对局会话中每局的额外的赌注分数，可选参数，缺省为 0 *
WatcherCount	允许观战的人数，可选参数，缺省最大 255(不限人)

➢ **接收方动作**

服务器发送应答消息表示成功或失败，如成功，返回所创建会话的 ID（SessionID 用于唯一标识一个对局会话）并把当前用户的状态转到 Sessional 状态。并向其它所有 Idle 的用户发送 NotifyNewSession 消息。

➢ **消息示例**

```
<NewSession>
   <Session>
     <Title>xxx引擎调试估值参数</Title>
     <Fen>r1bakabr1/9/1c4nc1/4p1p1p/pnp6/6P2/P1P1P3P/N1C1C1N2/R8/2BAKABR1 b</Fen>
     <TotalTime>600</TotalTime>
     <RequiredScore>2000</RequiredScore>
   </Session>
   <Description>new session<Description>      //此字段可选
</NewSession>
```

➢ **回应示例**

```
<NewSessionRes>
   <ReturnCode>0</ReturnCode>
   <SessionID>13517</SessionID>
   <Description>session 13517 create success<Description>      //此字段可选
</NewSessionRes>
```

4.10 通知新对局会话(NotifyNewSession)

➢ **消息标识:**
NotifyNewSession

➢ **流向:**
S=>C

➢ **说明:**
服务器向在线用户通知新的对局会话已建立。

➢ **发送端状态要求:**

➢ **消息参数**
对局会话的描述信息，参见创建对局会话。
对局会话的状态信息(对战双方用户信息，观战者信息等)

➢ **接收方动作**
刷新对局列表

➢ **消息示例**
TBD

4.11 加入指定的对局会话(EnterSession)

➢ **消息标识:**
EnterSession

➢ **流向:**
C=>S (客户端发送，服务器接收)。

➢ **说明:**
用户请求做为玩家加入一个指定的对局会话。

➢ **发送端状态要求:**
Idle

➢ **消息参数**
要加入的对局会话的 ID

➢ **接收方动作**
服务器发送应答消息表示成功或失败，如成功，把当前用户转为 Sessional 状态，并向 Session 中的其它用户发送 NotifyEnterSession 消息。

➢ **消息示例**

```
<EnterSession>
  <SessionID>13521</SessionID>
</EnterSession>
```

➢ **回应示例**

```
<EnterSessionRes>
  <ReturnCode>-1</ReturnCode>
  <ReturnInfo>error user status</ReturnInfo>
```

</EnterSession>

4.12 通知用户加对局会话(NotifyEnterSession)

- **消息标识:**

NotifyEnterSession

- **流向:**

S=>C。

- **说明:**

服务器向对局会话中的其它用户通知有新玩家加入。

- **发送端状态要求:**

无。

- **消息参数**

加入者的用户描述信息，参见 Login。

- **接收方动作**

无

- **消息示例**

TBD

- **回应示例**

TBD

4.13 加入某个对局会话观战(WatchSession)

- **消息标识:**

WatchSession

- **流向:**

C=>S (客户端发送，服务器接收)。

- **说明:**

用户请求做为观战者加入某个对局会话。

- **发送端状态要求:**

Idle。

- **消息参数**

要加入的对局会话的 ID。

- **接收方动作**

服务器向发送者返回加入是否成功。如成功，把当前用的状态转为 Watching，并向会话中所有其它用户发送 NotifyWatchSession 消息。

- **消息示例**

参见 EnterSession。

- **回应示例**

参见 EnterSession。

4.14 通知观战(NotifyWatchSession)

- **消息标识:**

NotifyWatchSession

- **流向:**

S=>C。

- **说明:**

当有人观战时向会话中的其它用户发出通知。

- **发送端状态要求:**

无。

- **消息参数**

观战者的用户信息。

- **接收方动作**

- **消息示例**

- **回应示例**

4.15 用户状态转为 Ready(Ready)

- **消息标识:**

Ready

- **流向:**

C=>S (客户端发送，服务器接收)。

- **说明:**

已加入会话的用户把自己的状态转为 Ready。

- **发送端状态要求:**

Sessional。

- **消息参数**

无

- **接收方动作**

把用户的状态转为 Ready

- **消息示例**

- **回应示例**

4.16 通知对局开始(NotifyStartPlay)

- **消息标识:**

NotifyStartPlay

- **流向:**

S=>C。

➢ **说明:**
当加入会话的双方都 Ready 之后，服务器向会话中所有人发送此消息，根据会话的设置把对战双方转为 Moving 或 Pondering 状态，并开始记时。

➢ **发送端状态要求:**
无。

➢ **消息参数**
棋局状态:
Fen 描述棋局初始状态
Black 持黑棋的用户
Red 持红棋的用户

➢ **接收方动作**
对于观战者应显示棋盘，对局者应根据自己的棋子先择前台或后台思考。

➢ **消息示例**
```
<NotifyStartPlay>
    <Fen> xxxx </Fen>
    <Black>bjwf</Black>
</NotifyStartPlay>
```

➢ **回应示例**
无

4.17 走棋(Move)

➢ **消息标识:**
Move

➢ **流向:**
C=>S (客户端发送，服务器接收)。

➢ **说明:**
处于 Moving 状态的用户发出走棋指令。

➢ **发送端状态要求:**
Moving

➢ **消息参数**
Step 当前的走步，采用于 UCCI 相同的方法描述一个走法，走法描述参见附录 1。
TimeStamp 从轮到自己走棋后到现在经过的秒数，可选参数，仅用于当服务器校准由于网络延迟产生的记时误差时的参考。

➢ **接收方动作**
服务器对此走法进行检验，并回应是否成功，如是合理走法，回应成功，并更新对话中对局双方的状态，然后发送 NotifyMove 消息给对局会话中其它用户。如走法不合理（违反会话中指定的规则，如送将，长将，长捉等），则回应不成功，并对走棋方记录违归一次。三次违归则会被服务器自动判负。

➢ **消息示例**

```
<Move>
    <Step>b0c2</Step>
    <TimeStamp>5</TimeStamp>
    <Description>马八进七</Description>
</Move>
```

➢ **回应示例**

```
<MoveRes>
  <ReturnCode>-1</ReturnCode>
</MoveRes>
```

4.18 通知用户走棋(NotifyMove)

➢ **消息标识:**

NotifyMove

➢ **流向:**

S=>C。

➢ **说明:**

向会话中处于 Pondering 状态的用户和观战者发送对手的走棋步法。

➢ **发送端状态要求:**

无。

➢ **消息参数**

同 Move

➢ **接收方动作**

客户端显示棋子的移动，处于 Pondering 状诚的客户端状态改变到 Moving。

➢ **消息示例**

➢ **回应示例**

4.19 向对手求和(Draw)

➢ **消息标识:**

Draw

➢ **流向:**

C=>S (客户端发送，服务器接收)。

➢ **说明:**

向对手请求和棋或同意和棋。如果某棋手发送 Draw 之后未发生对局双方未发生过状态改变（没走过棋），另一棋手回应 Draw，则服务器判和局，更新会话中所有人的状态,然后向会话中所有人发送 NotifyEndRound 消息，否则消息被忽略。

➢ **发送端状态要求:**

Moving 或 Pondering。

➢ **消息参数**
无

➢ **接收方动作**
服务器收到此消息后发送 NotifyDraw 通知消息给对手,对局双方在一局棋中各有三次求和机会，如在一局中发送超过三次 Draw 消息，则被服务器忽略。

➢ **消息示例**

➢ **回应示例**

4.20 通知对手求和(NotifyDraw)

➢ **消息标识:**
NotifyDraw

➢ **流向:**
S=>C

➢ **说明:**
服务器转发求和消息。

➢ **发送端状态要求:**
除 Offline 之外所有的状态。

➢ **消息参数**
无

➢ **接收方动作**
收到此消息的棋手可先择同意，或忽略（不同意），如同意可发送 AgreedDraw 消息给服务器

➢ **消息示例**

➢ **回应示例**

4.21 认输(GiveUp)

➢ **消息标识:**
GiveUp

➢ **流向:**
C=>S (客户端发送，服务器接收)。

➢ **说明:**
用户认输。

➢ **发送端状态要求:**
Moving 或 Pondering。

➢ **消息参数**
无

➢ **接收方动作**

向会话中所有人发送 NotifyEndRound 消息，并更新会话中所有人的状态

➢ **消息示例**

➢ **回应示例**

4.22 通知棋局结束(NotifyEndRound)

➢ **消息标识:**

NotifyEndRound

➢ **流向:**

S=>C。

➢ **说明:**

服务器通知棋局结束。

➢ **发送端状态要求:**

无。

➢ **消息参数**

终局描述信息

Result 对局结果

Reason 产生结果的原因，将死，超时，协商和棋，长将等等

Info 对局信息（双方的走法），用时等，参考 PGN 格式

➢ **接收方动作**

➢ **消息示例**

➢ **回应示例**

4.23 离开对局会话(LeaveSession)

➢ **消息标识:**

LeaveSession

➢ **流向:**

C=>S (客户端发送，服务器接收)。

➢ **说明:**

用户离开对局会话。

➢ **发送端状态要求:**

Sessional/Moving/Pondering/Watching。

➢ **消息参数**

无

➢ **接收方动作**

如果发送者是 Moving 或 Pondering 状态，则把当前局判负，向其它人发送 NotifyEndRound 通知消息，然后向 Session 中其它人发送 NotifyLeaveSession 消息。如果发送者是 Sessional/Watching 则仅向其它人发送 NotifyLeaveSession 消息。

- **消息示例**

- **回应示例**

4.24 通知离开对局会话(NotifyLeaveSession)

- **消息标识:**

NotifyLeaveSession

- **流向:**

S=>C。

- **说明:**

通知某用户离开对局会话, 如果离开者是对局双方中的一人，服务器会把此人对局结果判负，另一人状态置为 Sessional。

- **发送端状态要求:**

无。

- **消息参数**

无

- **接收方动作**

无

- **消息示例**

- **回应示例**

4.25 通知对局会话结束(NotifyEndSession)

- **消息标识:**

NotifyEndSession

- **流向:**

S=>C。

- **说明:**

当一个会话结束时，服务器向会话中的人发送消息通知。只有对战的双方都离开，时观战者时才会收到此消息。

- **发送端状态要求:**

无。

- **消息参数**

无

- **接收方动作**

无

- **消息示例**
- **回应示例**

4.26 通知用户断线(NotifyBreakLine)

- **消息标识:**

NotifyBreakLine

- **流向:**

S=>C。

- **说明:**

某用户断线时服务器向其它相关用户发出的通知。

- **发送端状态要求:**

无。

- **消息参数**

UserName 用户名

- **接收方动作**
- **消息示例**
- **回应示例**

4.27 列出所有房间(ListRooms)

- **消息标识:**

ListRooms

- **流向:**

C=>S (客户端发送，服务器接收)。

- **说明:**

列出服务器上所有房间。

- **发送端状态要求:**

除 Offline, 之外所有的状态。

- **消息参数**
- **接收方动作**

返回房间列表给发送者。

- **消息示例**
- **回应示例**

4.28 列出所有在线的用户(ListUsers)

- **消息标识:**

ListUsers

- **流向:**

C=>S (客户端发送，服务器接收)。

- **说明:**

列出本房间内所有在线的用户，如果发送端不在房间中，则返回状态错误。

- **发送端状态要求:**

除 Offline, Online 之外所有的状态。

- **消息参数**

SessionID 可选参数，如给出 SessionID，表示列出指定 SessionID 中的用户，否则列出所有在线用户。

- **接收方动作**

返回用户列表返回给发送者。

- **消息示例**

- **回应示例**

4.29 列出对局会话(ListSessions)

- **消息标识:**

ListSessions

- **流向:**

C=>S (客户端发送，服务器接收)。

- **说明:**

列出本房间内所有符合要求的对局会话。

- **发送端状态要求:**

除 Offline, Online 之外所有的状态。

- **消息参数**

- **接收方动作**

返回对局会话列表给发送者。

- **消息示例**

- **回应示例**

4.30 发送文字消息(SendMessage)

- **消息标识:**

SendMessage

➢ **流向:**
C=>S (客户端发送，服务器接收)。

➢ **说明:**
发送文字消息，用于在线用户之间的聊天。

➢ **发送端状态要求:**
除 Offline 之外所有的状态。

➢ **消息参数**
ReceiverList 接收消息的用户列表
Content 消息内容

➢ **接收方动作**
转发消息给指定的接收者,，如果接收者不在线，则忽略。

➢ **消息示例**

```
<SendMessage>
    <ReceiverList>bjwf u1 u2</ReceiverList>
    <Content>hello, I’ m coming. </Content>
</SendMessage>
```

➢ **回应示例**
无

4.31 通知文字消息(NotifyMessage)

➢ **消息标识:**
NotifyMessage

➢ **流向:**
S=>C。

➢ **说明:**
转发文字消息。

➢ **发送端状态要求:**
无。

➢ **消息参数**
同 SendMessage

➢ **接收方动作**

➢ **消息示例**

➢ **回应示例**

5 积分系统

[TBD]

6 CCNI 相关的数据结构定义

6.1 用户描述数据

UserName 字符串型, 4 -20 字节, 字母开头，可以是英文字母，数字，下划线。
Password 字符串型，8-20 字节
NickName 字符串型，8-20 字节
Score 数字型
Level 数字型

6.2 房间描述

Title
UserCount 房间可容纳的人数，最大为 255

6.3 对局会话描述

Title 本会话的标题
Fen 本会话中每局棋的初始棋盘状态，可选参数，缺省为开局
Red 持红棋的一方，可选参数，缺省为 0,表示先加入会话的一方(创建者)
Innings 期望的对局数，可选参数，缺省为 0,表示无数局，直到某一方退出
Rule 规则，可选参数，缺省为亚洲规则 *
TotalTime 一局的总时间，单位为秒，可选参数，缺省为 900(15 分钟)
StepTime 每步的时间，单位为秒，可选参数，缺省为 60(1 分钟)
BonusPerStep 每步加的时间,单位为秒，可选参数，缺省为 0
RequiredScore 加入会话所需的等级分，可选参数，缺省为 0
Ante 本对局会话中每局的额外的赌注分数，可选参数，缺省为 0 *
WatcherCount 允许观战的人数，可选参数，缺省最大 255(不限人)

6.4 对局结果

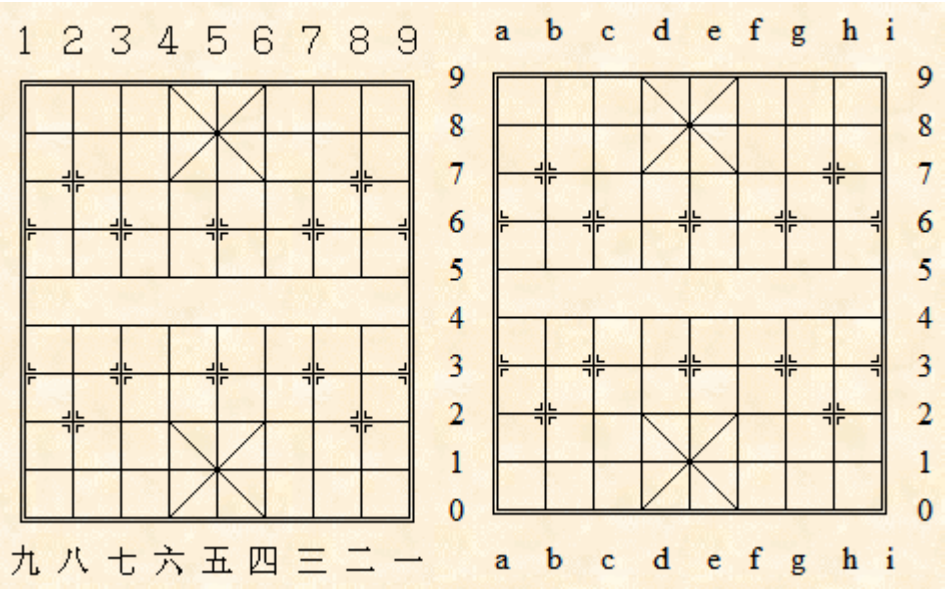
Result 对局结果
Reason 产生结果的原因，将死，超时，协商和棋，长将等等
Info 对局信息（双方的走法），用时等，参考 PGN 格式

6.5 增强型文字消息*

[TBD]

7 附录 1

着法表示，本协议采用坐标方式表示一个着法。



棋盘如上图所示，纵线从左到右(红方)依次为 a b c d e f g h i，横线从下到上(红方)依次为 0 1 2 3 4 5 6 7 8 9。
在初始盘面下，着法 b0c2 表示红方马八进七

[完]