**对原生Webrtc/Libjingle的Changelog**

**webrtc-server的修改**

**1、server逻辑、打印优化**

===================================================================  
**--- server/main.cc      (revision 348)  
+++ server/main.cc      (working copy)**  
**@@ -130,9 +133,12 @@**  
-              ChannelMember\* target = clients.IsTargetedRequest(s);  
+              ChannelMember\* target = clients.IsTargetedRequest(member,s);  
               if (target) {  
-                member->ForwardRequestToPeer(s, target);  
+                if(target->id() == member->id())  
+                  s->Send("500 Error", true, "text/plain", "","Can't call self");  
+                else  
+                  member->ForwardRequestToPeer(s, target);  
  
**@@ -157,7 +160,6 @@**  
-        printf("Disconnecting socket\n");  
  
**@@ -177,7 +179,6 @@**  
-        printf("New connection...\n");

**--- server/data\_socket.cc       (revision 348)  
+++ server/data\_socket.cc       (working copy)**  
**@@ -163,6 +163,7 @@**  
   buffer += "\r\n";  
   buffer += data;  
  
+  printf("[send]:%s\n",buffer.c\_str());  
   return Send(buffer);  
}

===================================================================

**2、会议名称的识别**

===================================================================  
**--- server/peer\_channel.cc      (revision 348)  
+++ server/peer\_channel.cc      (working copy)**  
**@@ -30,12 +30,18 @@**

**一些都文件和命名空间**+#include <sys/types.h>  
+#define MCU\_SESSION\_NAME\_PREFIX "mcu-se"  
+#define REGEX McuSession::regexHandler  
  
+using namespace std;  
  
**@@ -67,11 +73,15 @@**

**新增变量在构造函数的初始化**   assert(socket->PathEquals("/sign\_in"));  
+  session\_id = -1;  
+  remote\_peer = 0;  
   name\_ = socket->request\_arguments();  // TODO: urldecode  
  
**@@ -88,14 +98,19 @@**

**修改**GetPeerIdHeader()**使其能够生成正常ID或者会议ID**-std::string ChannelMember::GetPeerIdHeader() const {  
-  std::string ret(kPeerIdHeader + int2str(id\_) + "\r\n");  
+std::string ChannelMember::GetPeerIdHeader(bool isReally) const  
+{  
+  std::string ret;  
+  if(isReally || session\_id == -1)  
+      ret = kPeerIdHeader + int2str(id\_) + "\r\n";  
+  else  
+      ret = kPeerIdHeader + int2str(session\_id) + "\r\n";  
   return ret;  
}

**添加会议类型的判断**

bool ChannelMember::NotifyOfOtherMember(const ChannelMember& other) {  
   assert(&other != this);  
-  QueueResponse("200 OK", "text/plain", GetPeerIdHeader(),  
+  QueueResponse("200 OK", "text/plain", GetPeerIdHeader(true),  
                 other.GetEntry());  
   return true;  
}

**@@ -103,7 +118,11 @@**

**添加会议类型的判断**std::string ChannelMember::GetEntry() const {  
   char entry[1024] = {0};  
-  sprintf(entry, "%s,%i,%i\n", name\_.c\_str(), id\_, connected\_);  // NOLINT  
+  if(session\_id == -1)  
+      sprintf(entry, "%s,%i,%i\n", name\_.c\_str(), id\_, connected\_);  // NOLINT  
+  else  
+      sprintf(entry, "%s,%i,%i\n", session\_name.c\_str(), session\_id, connected\_);  // NOLINT  
+  
   return entry;  
}  
  
**@@ -111,9 +130,10 @@ ChannelMember::ForwardRequestToPeer**

**添加会议类型的判断**

-  std::string extra\_headers(GetPeerIdHeader());  
+  std::string extra\_headers(GetPeerIdHeader(false));

**当为bye消息时,清楚remote peer的记录，下次就不会路由了**

+    if(ds->data().substr(0,3) == "BYE")  
+    {  
+        remote\_peer = 0;  
+        peer->remote\_peer = 0;  
+    }  
  
**@@ -169,7 +194,91 @@**

**添加McuSession的实现，该类主要用于描mcu中会议的概念**+Tu\_Regex \* McuSession::regexHandler = new Tu\_Regex;  
+bool McuSession::IsMcuClient(std::string &clientName)  
+bool McuSession::IsMatchTheSession(std::string &clientName, std::string &sesName)  
+const char \* McuSession::getMcuSessionNameByClient(std::string &clientName)  
+McuSession \* McuSession::createMcuSession(std::string &sessionName, int sessionId)  
+bool McuSession::addMcuClient(ChannelMember \* mbr)  
+bool McuSession::removeMcuClient(ChannelMember \* mbr)  
  
**@@ -217,7 +326,7 @@**

**重写PeerChannel::IsTargetedRequest,使其能够查找会议类型的client**

     args = found + ARRAYSIZE(kTargetPeerIdParam) - 1;  
   } while (true);  
   int id = atoi(&path[found]);  
-  Members::const\_iterator i = members\_.begin();  
-  for (; i != members\_.end(); ++i) {  
-    if ((\*i)->id() == id) {  
-      return \*i;  
-    }

+...

**@@ -251,8 +410,49 @@**

**重写PeerChannel::AddMember,使其能够添加会议类型的client**   assert(IsPeerConnection(ds));  
   ChannelMember\* new\_guy = new ChannelMember(ds);  
   Members failures;  
-  BroadcastChangedState(\*new\_guy, &failures);  
-  HandleDeliveryFailures(&failures);  
+ ...  
  
**@@ -274,21 +478,18 @@ @@ -298,10 +495,7 @@  @@ -345,8 +593,7 @@  
修改PeerChannel::OnClosing,CheckForTimeout,HandleDeliveryFailures通过新函数deleteOne删除会话**

-      Members failures;  
-      BroadcastChangedState(\*m, &failures);  
-      HandleDeliveryFailures(&failures);  
-      delete m;  
+      deleteOne(m);

**@@ -314,13 +511,67 @@**

**新增PeerChannel::deleteOne,主要用于删除会话时做会议类型判断，必要时通知其他客户端**

**@@ -356,14 +603,32 @@**

**重写PeerChannel::BuildResponseForNewMember,为新用户生成用户列表时,不再将它自己放在第一条,添加会议类型的判断**

**--- server/peer\_channel.h       (revision 348)  
+++ server/peer\_channel.h       (working copy)**  
**@@ 37 @@  头文件添加**  
+#include <map>  
+#include "Tu\_Regex.h"

**@@ 44 @@  McuSession**

+class ChannelMember;  
+class McuSession{  
+public:  
+    McuSession(int \_id, const char \*\_name):  
+        id(\_id),name(\_name){}  
+  
+    ~McuSession(){}  
+  
+    static Tu\_Regex \*regexHandler;  
+    static bool IsMcuClient(std::string &clientName);  
+    static bool IsMatchTheSession(std::string &clientName, std::string &sesName);  
+  
+    static const char \* getMcuSessionNameByClient(std::string &clientName);  
+    static McuSession \* createMcuSession(std::string &sessionName, int sessionId);  
+    bool addMcuClient(ChannelMember \* mbr);  
+    bool removeMcuClient(ChannelMember \* mbr);  
+    bool isEmpty(){return ClientsMap.empty();}  
+  
+public:  
+    int id;//id<-1  
+    //name='mcu-session-name@computerName'  
+    std::string name;  
+    std::map<int, ChannelMember \*> ClientsMap;  
+  
+};  
+  
 **@@ 70 @@ ChannelMember修改**-  std::string GetPeerIdHeader() const;  
+  std::string GetPeerIdHeader(bool isReally) const;  
+  static int s\_member\_id\_;  
-  static int s\_member\_id\_;  
+public:  
+//  McuSession \*mySession;  
+  int session\_id;  
+  std::string session\_name;  
+  int remote\_peer;

**@@ 121 @@PeerChannel 修改**

+      s\_session\_id\_ = -1; // l = 125  
-  ChannelMember\* IsTargetedRequest(const DataSocket\* ds) const;  
+  ChannelMember\* IsTargetedRequest(ChannelMember\* srcMbr, const DataSocket\* ds) ;  
+  void deleteOne(ChannelMember \* member, Members\* failures = NULL);  
+  int s\_session\_id\_;  
+  std::map<int, McuSession \*> sessions\_;

===================================================================