OpenRTM-aist (C++) - 機能 #2196

プライベートネットワーク内のRTCに外からポートフォワーディング等をつかってアクセスできるようオブジェクトリファレンスを書きかえる方法について検討する

2011/07/22 19:58 - n-ando

ステータス:	終了	**開始日:**	2011/07/22
優先度:	通常	**期日:**	
担当者:	n-ando	**進捗率:**	100%
カテゴリ:		**予定工数:**	0.00時間
対象バージョン:			

説明

プライベートネットワーク内のRTCに外からポートフォワーディング等をつかってアクセスできるようオブジェクトリファレンスを書きかえる方法について検討する

さて、質問ですが、片方がprivate IPで片方がglobal IPだとどう通信すればよいですか？RTMの問題ではないかも知れませんが、これまで開発してきたテレオペロボットを外に出そうと思い3Gルータを購入したけどロボット側の通信コンポーネントがゾンビオブジェクトになってしまいます。考えたら当然なことですが、どう解決すればよいのかがよくわかりません。

構成は次のとおりです。

PC（Global IP1）-----Internet----3G Router (Global IP2)----Robot (Private IP)

通信コンポーネントはPCからRobotに命令を送る、RobotからPCにセンシングデータなどを送るために使います。

考えたのは「Robotを固定Private IPにして3G Routerの設定で受信で使うポートをロボットにそのまま流すようにする」です。（ネットゲームでよく使う手です）

これで動くと思いますか？ほかにRTMを使ったもっといい例があれば教えてください。

履歴

#1 - 2011/07/24 06:07 - n-ando

- 担当者 を n-ando にセット

- 進捗率 を 0 から 90 に変更

一応、ルータ（NAT）の内外のRTC同士の通信は実現できるようになりました。

自宅のMac mini上（プロバイダのルータの内側）上のConsoleOutと、
www.openrtm.org （Ubuntu10.04）上で動いているConsoleIn同士を接続して、
通信できることを確認しました。
これで、おそらく李さんが実現したいことはできていると思うのですがどうでしょうか？

以下、手順を示します。

1) ルータでポートフォワーディングを設定してください。
僕のケースでは、Mac mini のアドレスの 2810-2820 まで
ポート番号をそのまま転送するように設定しました。
#範囲は適当です。RTCのプロセスが1個ならポートは1つでもOK。

router (2810-2820) ---> Mac mini (2810-2820)

2) Manager.cpp の以下の関数（initORB()）に #ifdef ORB_IS_OMNIORB
の部分を追加してOpenRTMをコンパイルし直してください。
＃この変更により、オブジェクト参照に含まれる本来のIPアドレスに
＃加えて、2番目（あるいは3番目）のオプションとして、rtc.conf に指定した
＃別のアドレスを追加することができます。

bool　Manager::initORB()

```
{
        RTC_TRACE(("Manager::initORB()"));
        //  Initialize  ORB
        try
        {
        std::vector&lt;std::string&gt;  args(coil::split(createORBOptions(),  "  "));
        //  TAO's  ORB_init  needs  argv[0]  as  command  name.
        args.insert(args.begin(),  "manager");
        char**  argv  =  coil::toArgv(args);
        int  argc(args.size());

//  ORB  initialization
        m_pORB  =  CORBA::ORB_init(argc,  argv);
        //  Get  the  RootPOA
        CORBA::Object_var  obj  =
                m_pORB->resolve_initial_references((char*)"RootPOA");
        m_pPOA  =  PortableServer::POA::_narrow(obj);
        if  (CORBA::is_nil(m_pPOA))
    {
                RTC_ERROR(("Could  not  resolve  RootPOA."));
                return  false;
            }
        //  Get  the  POAManager
        m_pPOAManager  =  m_pPOA->the_POAManager();
//  ここから
#ifdef  ORB_IS_OMNIORB
                const  char*  conf  =  "corba.additional_ior_addresses";
                if  (m_config.findNode(conf)  !=  NULL)
            {
                        coil::vstring  addr_list;
                        addr_list  =  coil::split(m_config[conf],  ",",  true);

for  (size_t  i(0);  i  <  addr_list.size();  ++i)
                {
                        coil::vstring  addr_port  =  coil::split(addr_list[i],  ":");
                        if  (addr_port.size()  ==  2)
                    {
                            IIOP::Address  iiop_addr;
                            iiop_addr.host  =  addr_port[0].c_str();
                            CORBA::UShort  port;
                            coil::stringTo(port,  addr_port[1].c_str());
                            iiop_addr.port  =  port;
                            omniIOR::add_IIOP_ADDRESS(iiop_addr);
                        }
                    }
                }
#endif  //  ORB_IS_OMNIORB
//  ここまで
            }
        catch  (...)
{
        RTC_ERROR(("Exception:  Caught  unknown  exception  in  initORB()."  ));
        return  false;
            }
        return  true;
    }
```

3) rtc.conf に以下のように記述します

corba.nameservers: グローバル側のネームサーバのアドレス
corba.endpoints: :2810 ← 使用するポート
corba.additional_ior_addresses: ルータのアドレス:2810 ← 上で指定したのと同じポート番号

実際にはこんな感じです ↓
corba.nameservers: my.global.nameserver.com
corba.endpoints: :2810
corba.additional_ior_addresses: w.x.y.z:2810

4) グローバル側とプライベート側のRTCをそれぞれ起動してRTsystemEditorでつなぎます
つなぐとき、少し時間がかかる可能瀬が有ります。これは、
まずはじめに、本来のアドレスに接続に行って、それがだめな場合、rtc.conf で指定した
別のアドレスに接続しに行くためです。

以上で、ルータ内部のRTCと外部のRTCの通信ができるようになります。
問題が有ればまたお知らせください。

#2 - 2013/04/15 17:46 - n-ando

- ステータス を 新規 から 終了 に変更

- 進捗率 を 90 から 100 に変更

corba.alternate_iiop_address オプションにて実現