

Re: Problem using Remoting with 2 NICs

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2006-05/msg00003.html>

- *From:* KidO <KidO@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 2 May 2006 05:11:02 -0700
-

I agree with you on the 2 NIC's with same subnet being a bad idea. In this case, the end-customer's internal LAN subnet (connected to NIC1) was the same as the subnet we normally use on NIC2 for a private workgroup. The subnet on NIC2 is determined by a piece of hardware, which I am currently not able to change the subnet on.

Fortunately, the same subnet for each NIC was not the cause of the problem. I tested in both cases where the NIC's had the same subnet, and different subnets. Each case resulted in the same error.

Setting the http channels useIpAddress property to false on the server fixed the error.

Thanks,
--
- John

"Chaz" wrote:

Having 2 NIC's configured with IP's in the same subnet on any PC is generally a bad idea, it does not perform load balancing as you would expect. Is there some reason you have set it up this way ?

Chaz

"KidO" <KidO@xxxxxxxxxxxxxxxxxxxx> wrote in message
<news:1BF04534-7988-424C-A0A9-C371D19F39DF@xxxxxxxxxxxxxxxxxxxx>

Chaz,

Thanks for the reply.

Yes, the 2 NIC's do share the same subnet. My first attempt at correcting the problem was to change the subnet of one of the NICs. One side (static IP

Re: Problem using Remoting with 2 NICs

– is a small workgroup LAN) Unfortunately, this did not correct the problem.

I have found that adding the property

```
props["useIpAddress"] = false;
```

to the channel configuration (see RegisterServerChannel method from my original post) when creating the Http Channel(s) allows me to successfully connect to the server from either NIC.

The server itself is currently running in a dual NIC configuration at another installation with no problems. This installation was performed about 10 months ago. I know that no OS level changes have been made on that server since the software was installed. The PC in which the server is installed now is a fully up to date WinXP Pro_SP2 box. I am suspecting that something has changed at the OS (API) level between then and now. I currently have a case open with MS support in hopes to get to the bottom of the problem.

Thanks,

--

– John

"Chaz" wrote:

Do the 2 NIC's ever share the same IP Subnet ?

e.g. NIC#1 DHCP Assigned 192.168.10.20
NIC#2 Static 192.168.10.1

Chaz

"KidO" <KidO@xxxxxxxxxxxxxxxxxx> wrote in message
news:CBF9BB69-D939-406B-928A-0038A5A382D1@xxxxxxxxxxxxxxxxxx

I have a remoting server (Class Factory)
running on a PC with 2 NICs.
NIC
1
is DHCP, NIC 2 is fixed IP Address.

I have clients connecting successfully on
NIC 1, but clients connecting
to
NIC 2 fail when accessing the methods of

Re: Problem using Remoting with 2 NICs

the inner classes of the Class
Factory.

For each NIC (IP Address), I create a Server
Channel as follows, using

a
different channel name for each NIC (IP
Address), and binding the
channel
to
the IPAddress of the NIC:

```
private void RegisterServerChannel(string  
name, int port, IPAddress  
ipAddress)  
{  
IDictionary props = new Hashtable();  
props["name"] = name;  
props["port"] = port;  
props["bindTo"] = ipAddress.ToString();  
props["strictBinding"] = true;
```

```
BinaryServerFormatterSinkProvider  
serverProv =  
new BinaryServerFormatterSinkProvider();  
serverProv.TypeFilterLevel =  
System.Runtime.Serialization.Formatters.TypeFilterLevel.Full;  
BinaryClientFormatterSinkProvider  
clientProv =  
new BinaryClientFormatterSinkProvider();
```

```
HttpChannel chnl = new HttpChannel(props,  
clientProv, serverProv);
```

```
ChannelServices.RegisterChannel(chnl);  
}
```

After each Server Channel is registered, I
call:

```
RemotingServices.Marshal(classFactory,  
objURI,  
typeof(IClassFactory));
```

I have a shared assembly that contains the
interfaces for the classes
(simplified):

```
namespace SharedInterfaces  
{  
public interface IClassFactory  
{
```

Re: Problem using Remoting with 2 NICs

```
string SomeString{ get; set; }  
InnerClass InnerClass { get; }  
}
```

```
public interface InnerClass  
{  
    bool SomeValue { get; }  
}
```

When both NICs are enabled (clients connected to NIC 1 work perfectly). Clients on NIC 2 can connect and successfully call `IClassFactory.SomeString`, but when the Client on NIC 2 calls `IClassFactory.InnerClass.SomeValue`, I get the following exception:

`System.Net.WebException: The underlying connection was closed: Unable to connect to the remote server.`

Server stack trace:

```
at  
System.Net.HttpWebRequest.CheckFinalStatus()  
at  
System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult  
asyncResult)  
at  
System.Net.HttpWebRequest.GetRequestStream()  
at  
System.Runtime.Remoting.Channels.Http.HttpClientTransportSink.ProcessAndSend(I  
msg, ITransportHeaders headers, Stream  
inputStream)  
at  
System.Runtime.Remoting.Channels.Http.HttpClientTransportSink.ProcessMessage(I  
msg, ITransportHeaders requestHeaders,  
Stream requestStream,  
ITransportHeaders& responseHeaders,  
Stream& responseStream)  
at  
System.Runtime.Remoting.Channels.BinaryClientFormatterSink.SyncProcessMessage  
msg)
```

Exception rethrown at [0]:

```
at  
System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage
```

Re: Problem using Remoting with 2 NICs

```
reqMsg, IMessage retMsg)  
at  
System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData&  
msgData, Int32 type)  
at  
SharedInterfaces.IInnerClass.get_SomeValue()
```

If I disable NIC1, the clients can access the inner classes of the class factory on NIC 2. With both NIC 1 and NIC 2 enabled, the clients on NIC 2 can access "top level" methods of the class factory, but cannot access the methods referencing the inner classes.

In short, the application works fine when there is only 1 NIC, but something is going wrong when using 2 NICs. I suspect that I have something missing in my configuration, but I am not able to find it.

Any input is greatly appreciated.

Thanks in advance.

--
- John