

Re: peer to peer messaging

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2005-05/msg03028.html>

- *From:* "Chris Uppal" <chris.uppal@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 31 May 2005 12:45:05 +0100
-

[It's a bit late to reply, but maybe this'll be of interest to someone]

Tim Ward wrote:

- > If machine A which knows its IP address is X can talk to machine B via IP
- > address Y then we have no way of knowing
- >
- > (a) what IP address machine B thinks it has
- > (b) what IP address machine B should use to talk to machine A
- > (c) what IP address machine C should use to talk to machine B
- > (d) on which of its interfaces B would be best advised to listen for
- > connections from machine C
- >
- > and so on and so on and so on.

I think that we may be talking about different things, since what you say doesn't seem to make sense to me. I'll run through my understanding of NAT since that might be of some help to "Bond", or even of interest to yourself. (BTW, this has turned out much longer than I planned -- sorry about that.)

The thing to remember is that the Internet wasn't designed with NAT in mind. In fact NAT is an attempt to "fix" the way that the Net works without either of the end-point machines being aware of it.

Start with a simple case, with no NAT-ting involved anywhere. Say my laptop is connected to the Internet in the old-fashioned way, and that it attempts to the HTTP server at 209.249.116.141 (I'll ignore named IP addresses like 'java.sun.com' completely in this; they are not relevant since they are resolved before any of the stuff we are talking about happens). My machine attempts to open a connection to port 80 of the server at that IP address. The server responds by setting up a connection, and thereafter my machine and the server can communicate by sending IP packets with the appropriate port numbers embedded in them. For example a packet from my machine might have source IP aaa.bbb.ccc.ddd, and source port 3872 (randomly allocated when it attempts to make the connection) and destination address 209.249.116.141 and port 80. Packets from the sever to my laptop would have those reversed. Here, "my" IP address (aaa.bbb.ccc.ddd) is allocated by the relevant networking authorities, and has been assigned to me, and any machine anywhere on the network can address my machine using that address. (The same is true of the server's

Re: peer to peer messaging

address too, of course.)

A couple of things are worth noting since they'll become relevant later. Neither my machine nor the server are interested at all in the addresses of any routers in between us (the TCP/IP implementation on each machine does have to know about the /nearest/ router, so that it can route packets via it, but that knowledge is only used by the TCP/IP implementation in the OS kernel or wherever, it is not needed or used by ordinary network programmers). Secondly, the quadruple of <source-port, source-IP, destination-port, destination-IP> is sufficient to "label" packets as belonging to a specific TCP/IP connection between one program running on my machine and one program running on the server.

Now let's assume that my laptop is not connected directly to the whole Internet, but is instead on a NAT-ed subnet connected to the real Net by a NAT-ing router. In this case, the router itself "sits on" my public IP address (which may not be all that public if it's temporarily allocated by my ISP, or it may be fixed and well-known with DNS entries and everything). So it is at address aaa.bbb.ccc.ddd. My own laptop is probably given an IP address from the "private use" range. Say it's 192.168.0.20. Other machines on the same subnet have IP addresses in the same private use range, but none of them have (I hope!) the exact same address.

Now, when my laptop attempts to connect to any other machine it will use that machine's IP address (as supplied by DNS presumably), and it's own IP address, 192.168.0.20, as the source address. It does that because that's what it thinks its address is, and that's how TCP/IP works. For machines on my subnet that "just works", but when I attempt to connect to the server 209.249.116.141 then things would go wrong. If nothing intervened, then it would connect to 209.249.116.141 saying that the packets came from 209.249.116.141. If the server replied at all (which it probably has been configured not to) then the return packet would go missing, since the network as a whole has no idea how to route a packet addressed to 209.249.116.141 back to my laptop. But something /does/ intervene. What's more, it intervenes /transparently/. Somewhere between my laptop and the "real" Net is my NAT-ing router. When it sees a packet emanating from my laptop (or anywhere else in the NAT-ed subnet that it manages), it changes that packet so that it appears to come from its own IP address, aaa.bbb.ccc.ddd, and from a port number that it has temporarily allocated. It remembers that it is now managing a NATed connection for that IP/port. The server sees this packet just as before, and replies just as before. The reply ends up back at my router, which inspects the packet, and sees that the destination IP/port are for a connection that it knows about. It looks that data up in its internal list of connections, and finds the IP/port that my machine originally used. It changes the incoming packet to use that destination IP/port and sends the packet on to my machine.

The end result of all that is that my machine and the server can communicate in traditional TCP/IP way, even though my machine is not at a public IP address. As I said the NAT-ing is transparent, neither the network program (Firefox, say) nor my machine's OS know anything at all about the NAT (and there's nothing that they could usefully do with the knowledge if they had it).

Re: peer to peer messaging

Re: peer to peer messaging

OK, let's make it a bit more complicated. Now I'm going to set up a public web server on my laptop. I want anyone on the web to be able to connect to <http://aaa.bbb.ccc.ddd/> and end up talking to my server. I can easily start an HTTP server on the laptop, but the problem is that it will be listening on port 80 at address 209.249.116.141. Not at aaa.bbb.ccc.ddd, which is a completely different address owned by my NAT-ing router. So what I have to do is go configure the NAT part of the router so that it knows that I want to allow connections to port 80 of my machine from the network as a whole (I'll probably have to change the firewall configuration too, but that's a different story). I tell it that when it sees an incoming packet aimed at port 80 of IP aaa.bbb.ccc.ddd, that it should do the same kind of packet modification as before to change the destination address to that of my own machine, and then forward the packet onwards. Similarly when it sees a packet going back on the same connection, it should re-write that package's source IP so that it seems to be coming from aaa.bbb.ccc.ddd.

Again, once I've done a little manual configuration, the whole process is transparent to both the webserver running on my laptop, and to the client machine somewhere on the Web.

There are a couple of potential problems with this. In the above description, the NAT can keep track of which connections it is modifying because it can track the TCP protocol as it opens and closes the connection. For connectionless protocols such as UDP/IP that isn't possible. So it has to use some sort of heuristic to attempt to keep the re-write information around for as long as it will be needed, but not clog up memory by keeping it for too long. (Actually, there's a similar problem with TCP/IP traffic, since the connection may be dropped without the normal shutdown messages being exchanged --- e.g. if one machine crashes). In theory this is a problem, but it doesn't seem to be too hard to solve in practice.

The other class of problem is caused by a handful of rather weird protocols (FTP comes to mind) which embed IP addresses and port number in the body of the protocol. In such cases there are only two answers. Either the NAT-ing router can "understand" that protocol, and know how to modify the bodies of the messages too. E.g. my router understands FTP, so FTP (in either active or passive mode) works fine even though there's a NAT in the way. The other possibility is that the NAT-er /doesn't/ understand the protocol, in that case the application programmers will have to do the equivalent of NAT-ing themselves. Such cases are (as far as I know) rare. There is little reason to embed IP numbers and ports into the body of TCP/IP messages. The biggest exception that I know of, is tunnelling TCP/IP connections over SSH, and similar. To be honest I don't know nearly enough about how that works to say more about it, nor to speculate on how it is managed without co-operation from the application programmers.

--- chris

- *Follow-Ups:*

- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Bond

- *References:*

- ◆ **[peer to peer messaging](#)**
 - ◇ *From:* Bond
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Thomas Weidenfeller
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Bond
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Guy Noir
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Bond
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Guy Noir
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Tim Ward
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Chris Uppal
- ◆ **[Re: peer to peer messaging](#)**
 - ◇ *From:* Tim Ward

- Prev by Date: **[Re: Java WSDP and Tomcat 5](#)**
- Next by Date: **[Java program architecture \(Threads...\) : Advices needed](#)**
- Previous by thread: **[Re: peer to peer messaging](#)**
- Next by thread: **[Re: peer to peer messaging](#)**
- Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**