

Re: incoming connection port 80

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-03/msg00234.html>

- *From:* Christian <fakemail@xxxxxx>
 - *Date:* Tue, 04 Mar 2008 21:05:41 +0100
-

Erik schrieb:

On 4 mrt, 20:35, "Peter Duniho" <NpOeStPe...@xxxxxxxxxxxxxxxxxxxx> wrote:

On Tue, 04 Mar 2008 10:55:05 -0800, Mark Space
<marksp...@xxxxxxxxxxxxxxxxxxxx>
wrote:

Erik wrote:

I'm wondering how to accept connections (socket) if you are behind a router. Skype and uTorrent can handle this (by using port 80 or 443). How do these programs manage to accept connections if ports (accept port 80 and 443) are blocked?
Thanks
Erik

uTorrent is just a Bit Torrent client. Bit Torrent connects out to a server, it does not accept incoming connections. Its incoming connections are not low number ports (80 or 443) and have to be specifically enabled on the router/firewall or it won't work well.

If it doesn't accept incoming connections, then why do its incoming connections "have to be specifically enabled"? :)

To the original poster: an application that has a listening TCP socket does indeed require that the router _somehow_ be configured to forward connection requests to that socket. The two most common techniques involve manually configuring the router or using the "universal plug-and-play" protocol (by which a network application can obtain

Re: incoming connection port 80

specific information from the router and/or configure the router to do specific forwarding).

Many routers support "port triggering", by which the router watches outbound traffic and if it notes a network client using some particular port (either locally or, more commonly, in the remote address), it automatically enables forwarding to that client temporarily on some other specified port or ports (which may include the original outbound port). Specifics on this vary from router to router.

You may also want to Google "nat hole punching". It's more reliable when used with UDP than TCP (different techniques are used with each, and doing it using TCP requires lower-level access than sockets normally give you), and in either case it's not 100% reliable as it depends on undocumented, arbitrary behavior on the part of the router. But depending on how important it is to solve the problem, it's something you might consider.

Note that if the router has literally "blocked" the ports, then the answer is "you don't". Typically, the ports are only "blocked" in that the router doesn't know who to forward the traffic to. This is addressable as described above. But if someone's actually configured the router to not allow traffic on those ports to pass, then the only thing that will allow traffic on those ports through is to reconfigure the router so that it's no longer blocking traffic on those ports.

Finally note that a "router" is not the same as a "firewall". Sometimes the two functions are combined into a single device, but a firewall's job is specifically to block traffic. Either it's blocking traffic on specific ports or it's not. If it's not, you have nothing to do, and if it is, nothing you can do short of changing the firewall configuration is going to unblock the ports. Obviously, changing the firewall configuration is not something that would be done automatically by a software client without any user intervention. Otherwise it wouldn't be much of a firewall. :)

Pete

Thanks for your answer :).

What I'm trying to do is to create a Java applet which can receive an incoming connection, so a connection is established between a program and the applet (which must be able to accept the connection). Is this possible (to bypass the router (not the firewall!)) and if it's possible, how?

Thanks

Erik

Re: incoming connection port 80

A standard way would be to use UPnP protocol to open up a port.

Everything else is more sophisticated.. like tricking the router into believing a connection was created .. probably not what you want (hole punching like told above)

As a sidenote port 80 and alike are often used to trick admins/isps that don't work with level-7 filters but just block unwanted traffic by port.

.