**The Basics**

To really explain port forwarding, you first need to understand a little more about what your [router](http://superuser.com/tags/router/info) does. Your internet service provider assigns **one** IP address to your internet connection. All computers on the internet need a unique IP address, but you have multiple computers in your house and only one address. So how does this work?

***If you know what it is and just want to know how to do it***: <http://portforward.com/> has a how-to with screenshots for literally hundreds of different routers. The documentation is there hidden behind an ad-page for their automatic portconfig tool. (Just click around a bit an you'll find it.)

**NAT - What is it? Why do we use it?**

Your home router has a function called Network Address Translation, or NAT, built in. Inside your network, computers have addresses like 192.168.1.100. All addresses in the 192.168.\* range (or in the 10.\*) range are *"*[*private*](http://en.wikipedia.org/wiki/Private_IP_addresses#Private_IPv4_address_spaces)*"* or *"*[*reserved*](http://en.wikipedia.org/wiki/Reserved_IP_addresses)*"* addresses. These addresses are officially assigned by IANA to be used inside of private networks. Your router automatically assigns such an address to each computer connected via [DHCP](http://en.wikipedia.org/wiki/DHCP). These addresses are how computers in your network communicate with the router and with each other.

Your router has a separate network interface that connects it to the internet. This interface has a very different address which is assigned by your ISP. This is the one address that I mentioned before, and your router uses it to communicate with other computers on the internet. Computers inside of your network have *non-routable* private IP addresses, meaning that if they send packets directly to the internet the packets will automatically be dropped (packets with private addresses are not allowed to traverse the internet for stability reasons). But your router has a *routable* address. Network Address Translation, as its name suggests, *translates* between these two kinds of addresses, allowing the multiple computers inside of your network to appear to the internet as one computer with one address.

**The Details**

Although this might sound complicated, it's actually pretty simple how your router does it. Every time a computer inside your network wants to connect to a computer on the internet, it sends the connection request to the router (it knows to send it to the router because its *Default Gateway* parameter is set to the router's address). The router then takes that connection request (a "SYN request" in TCP/IP) and changes the source address (the "reply-to" or return address) and changes it from the private IP of the computer to the public IP of the router, so that the response will be sent to the router. It then takes note in a database (called the *NAT table*) that the connection was initiated, so that it remembers it later.

When the response comes back from the remote computer (a "SYN-ACK"), the router looks in its NAT table and sees that a connection to that host on that port was previously initiated by a private computer on your network, changes the destination address to the private address of the computer, and forwards it inside your network. In this way, packets can continue to transit back and forth between networks, with the router transparently changing the addresses so that it works. When the connection is terminated, the router just removes it from the NAT table.

**Or think of it this way**

This might be a little easier to visualize with a metaphor - let's say you're a freight forwarder in the US working with Chinese clients. They need to send packages to many customers in the US, but it's easier for customs/paperwork reasons to only send packages to one place. So, a package comes to you from one of your clients in China (the private network, in this example) with an actual destination somewhere in the US (the internet). You change the address label on the box to the US (public) address, and you change the return address to your own public address (since it can't be returned straight to China without inconveniencing the customer) and hand it to the postal service. If the customer returns the product, it comes to you. You look it up in your records and see what company in China it came from, and change the destination to that company (its private address) and the return address to your private address, so that they can send back a replacement through you.

**And finally we get to Port Forwarding**

**Port forwarding allows remote computers (for example, computers on the Internet) to connect to a specific computer or service within a private local-area network (LAN)**

This works great, but there's a bit of a problem. What if a customer needs to send something to the company, let's say a money order in payment for something? Or, let's say that a computer on the internet initiates a connection with the router (a SYN request), say to a web server that is in the network. The letter/packet only has the router's public address on it, so the router actually doesn't know where to send it! it could be destined for any of the computers on the private network, or for none of them. You might have experienced this problem when you call someone's home phone - when they call you it's no problem, but when you call them there's no way for them to know who's the call for, so the wrong person might answer.

While it's easy enough for humans to sort this out, it's a lot trickier for computers, because not every computer on your network knows all the other computers.

Port Forwarding is how we fix this problem: it's a way to tell your router what computer inside the network incoming connections should be directed to. We have three different ways we can do this:

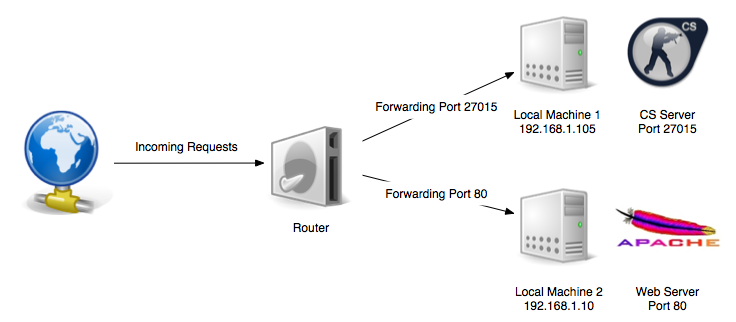
* Faux-DMZ: a lot of routers have a feature called DMZ. This stands for Demilitarized Zone, which is a kind of network security configuration. The DMZ on home routers is often referred to as faux-DMZ because it lacks the features of an actual DMZ. What it does do is the simplest kind of incoming connection handling: all incoming connection requests will be sent to one specified inside your network. It's dead simple - you type an IP address in to your router's configuration, and all incoming connections go there. This doesn't always work, though, because you might have multiple computers that need to accept incoming connections. For that, we have...
* Port forwarding: All network connection requests include a "port". The port is just a number, and it's part of how a computer knows what the packet is. IANA has specified that Port 80 is used for HTTP. This means that an incoming packet that says port number 80 must be a request intended for a web server. Port forwarding on your router allows you to enter a port number (or possibly a range or combination of numbers, depending on the router), and an IP address. All incoming connections with a matching port number will be forwarded to the internal computer with that address.
* UPnP port forwards: UPnP forwarding works the exact same way as port forwarding, but instead of you setting it up, software on a computer inside the network automatically sets the router to forward traffic on a given port to it.

**An example**

Let's look at a usage example. A lot of multiplayer video games (as an example, Counter Strike) allow you to run a game server on your computer that other people can connect to in order to play with you. Your computer doesn't know all the people that want to play, so it can't connect to them - instead, they have to send new connection requests to your computer from the internet.

If you didn't have anything set up on the router, it would receive these connection requests but it wouldn't know which computer inside the network had the game server, so it would just ignore them (or, more specifically, it would send back a packet indicating that it can't connect). Luckily, you know the port number that will be on connection requests for the game server. So, on the router, you set a port forward with the port number that the game server expects (for example, 27015) and the IP address of the computer with the game server (for example, 192.168.1.105).  
The router will know to forward the incoming connection requests to 192.168.1.105 inside the network, and computers outside will be able to connect in.

Another example would be a local network with two machines, where the second one with the IP 192.168.1.10 hosts a website using Apache. Therefore the router should forward incoming port 80 requests to this machine. Using port forwarding, both machines can run in the same network at the same time.



Video games are perhaps the most common place everyday users will encounter port forwarding, although most modern games use UPnP so that you don't have to do this manually (instead, it's fully automatic). You'll need to do this whenever you want to be able to connect directly to something in your network though (rather than through some intermediary on the internet). This might include running your own web server or connecting via Remote Desktop Protocol to one of your computers.

**A note on security**

One of the nice things about NAT is that it provides some effort-free, built-in security. A lot of people wander the internet looking for machines that are vulnerable... and they do this by attempting to open connections with various ports. These are incoming connections, so, as discussed above, the router will drop them. This means that in a NAT configuration, only the router itself is vulnerable to attacks involving incoming connections. This is a good thing, because the router is much simpler (and thus less likely to be vulnerable) than a computer running a full operating system with a lot of software. You should keep in mind, then, that by DMZing a computer inside your network (setting it as the DMZ destination) you lose that layer of security for that computer: it is now completely open to incoming connections from the internet, so you need to secure it as if it was directly connected. Of course, any time you forward a port, the computer at the receiving end becomes vulnerable on that specific port. So make sure you run up-to-date software that is well configured.