Contents

Introduction

The internet works using two main address units: the IP Address and the port. When your computer makes a call on the internet--trying to load www.dd-wrt.com for example--it starts by asking the IP address of www.dd-wrt.com for the webpage. However, it can't just ask dd-wrt.com's IP address for the webpage files--that would be like leaving off the apartment number when mailing someone. It needs to ask dd-wrt.com's IP address on port 80--the universal webpage port. Your computer instructs the response to be sent back to your IP address on some port that you opened to receive that data. By using ports, your computer can keep track of which stream of data belongs to what.

For example, when requesting a webpage with some text and 1 image, your web browser might ask that the main text content be sent back to it on port 10345 and the image be sent back on 10548. It might also be receiving instant messages on other ports and e-mails on yet other ports. The ports here don't matter because your computer just makes them up on the fly. What does matter are server ports. A web server doesn't announce that it's waiting for webpage requests on port 80, that's just known. It's a standard. In fact, all of the ports from 1-1024 are set aside for such standardization.

On the internet there are two types of addresses: public IP addresses and private IP addresses. Public IP addresses are those addresses that are routable on the internet. These consist of the bulk of all IP addresses. Addresses that begin with 10.x.y.z or 192.168.x.y (where x, y, and z can be anything 0-255), and the range from 172.16.0.0 to 172.31.255.255 are strictly private addresses and cannot exist on the internet. Additionally, 127.x.y.z is set aside as the local loop back address and, depending on your computer, 127.x.y.z will reach your own system. On most computers this is limited to 127.0.0.1, but that needn't be so.

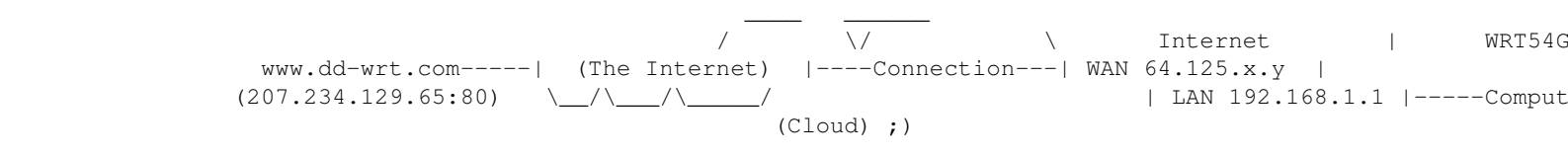
There are only 2^{32} *possible* IP addresses, and worldwide there are many more than 2^{32} devices (computers, printers, scanners, etc) that would like to have internet IP addresses. Private Addresses and, more specifically, NAT were setup to solve the problem of a limited number of IP addresses. The new IP Protocol specification, IPv6 intends to solve this problem by increasing the number of addresses.

Network Address Translation (NAT) works by making an entire network of privately addressed devices appear as just 1 device on the internet. NAT is usually done inside of a router, like the WRT54G, but can also be done on a computer running Linux, MAC OS, or Windows provided you have more than 1 network adapter.

When a computer behind a NAT device makes a call to the internet, it sends it's packet to the NAT device just as though it were going to send it directly to it's destination. To the privately addressed device, the NAT device appears to be any ordinary router.

When the NAT device receives a packet destined for the internet, it reconstructs the packet taking note of the original sender IP and Port and then resends that packet as though it were making the request. When the information comes back, it sends it back to the original device as though it had come from the computer on the internet.

Example:



1. Computer A tries to connect to www.dd-wrt.com, it sends a packet like:
   To=207.234.129.65:80 From=192.168.1.100:16848 Get index.html

2. The WRT54G intercepts this packet and sends the following:
   To=207.234.129.65:80 From=64.125.15.256:15846
   and makes a note:
   Anything received on port 15846 goes to 192.168.1.100 on port 16848

3. www.dd-wrt.com responds with
   To=64.125.15.256:15846 From=207.234.129.65:80 <Contents of Index.html>

4. The WRT54G sees the information is addressed to 15846, so it sends:
   To=192.168.1.100:16848 From=207.234.129.65:80 <Contents of Index.html>

In this way neither www.dd-wrt.com nor Computer A knew that they weren't talking directly to each
(all ports except 80 were completely made up and were simply for illustration
64.125.15.256 is an invalid ip address on purpose)

Primer on Port Forwarding

In the example above, during step 2 the WRT54G makes a note that all information received on port whatever should be relayed to a certain IP address on a certain port. This is essentially port forwarding, but this happens automatically.

Port Forwarding is generally considered when you manually define a rule in the router to send all data received on some range of ports on the internet side (WAN Jack) to a port and IP address on the LAN side (LAN Jacks or Wireless Antennas).

You will need to do this whenever your computer opens a port to receive connections without first connecting to a machine on the internet. This happens if you're running a Web server (80), FTP Server (21), SSH server(22) etc on one of your local computers that you would like to be visible on the internet. Many games and instant messaging clients also open ports without trying to connect out first, and these ports may need to be specifically defined as well.

Configuring Port Forwarding

It's a good idea to set static IP addresses for any computers you would like to forward ports to. This can be done using Static DHCP or by manually configuring IPs in your OS. Be sure to set your static IPs outside of your automatic (DHCP) address range. This range is 192.168.1.100-192.168.1.149 by default.

For more instructions and a list of further ports, see the external links.

Port Forwarding using the Webinterface

Some of this is tricky until you get used to the "sequence" required.

For an example, if you need 5 port forward entries, click "add" 5 times and you will have 5 blank lines to fill in. When you have your lines, click "save"

Enter the info in one line, click save, enter another, click save, etc.

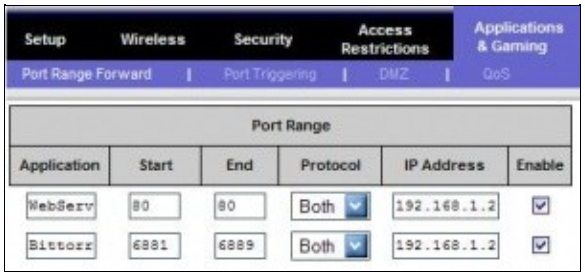
When finished, click save one more time for fun then click apply settings.

Same goes for port range forwarding.

Warning

UPnP port forwards seem to overwrite static port forwards set here. If your static port forwarding is important, turn off UPnP.

Port Range Forward



Portforwarding for a web server and a bittorrent client

This is the most common port forwarding and always forwards ports to the same machine (LAN IP) on the network.

- Browse to the web interface.
- Click on the "NAT / QoS" tab.
- Click on the "Port Range Forward" subtab.
- Enter any **Application** name you'd like. This is for your own reference and does not matter to to the router.
- Enter the **Start** port in the range you'd like to forward.
- Enter the **End** port in the range you'd like to forward. If you're just forwarding 1 port, set them both equal.
- Enter the local **IP Address** of the machine you'd like to forward the port to.
- Check the "Enable" checkbox to enable forwarding for this port range.

If the computer you're forwarding to is configured by for automatic IP address by DHCP, you will periodically have to update the last field to reflect the machine's current Local IP Address.

Here you can see that connecting to <WAN_IP_Address>:80 will bring you to the webserver on 192.168.1.2 and connecting to any port between 6881 and 6889 on <WAN_IP_Address> will bring you to the bittorrent client on 192.168.1.2.

Triggered Port Forwarding

Triggered port forwarding for a AIM file transfers
Triggered port forwarding is not the same as port range forwarding, and works by forwarding requests to a range of ports to the machine that first connected to a remote host on the trigger port. Port triggering is nice because it's semi-automatic and doesn't care about static IP addresses. For example, this could be used to forward the File Transfer ports that AIM uses to *any* computer that connects to the AIM servers, not just the computer you hard code.

- Browse to the Web Interface.
- Click on the "NAT/QoS" tab.
- Click on the "Port Triggering" subtab.
- Enter any **Application** name you'd like. This if for you and makes no difference.
- Enter the **Start** and **End** ports in the range that needs to be triggered. If you just want 1 trigger port, set them equal.
- Enter the protocol of the traffic you want forwarded
- Enter the **Start** and **End** ports in the range you'd like to forward. If you're just forwarding 1 port, set them both equal.

When a local computer on your LAN connects to any remote IP address with the destination port 5190 then WAN traffic received on ports 4117-4443 will be forwarded to the local computer. Since the AIM servers listen for connections on 5190, this means that as soon as a local computer connects to the AIM servers, the ports AIM uses for file transfers will be opened on the router and forwarded to the local computer automatically.

The key to port triggering is that you must have an application on the inside of your network which can "Trigger" the router to open the port. Triggering has two main benefits. First, two computers on your LAN can time-share the same ports on the router. So for example if you use AIM on one computer in the morning and another in the afternoon this would work great. You cannot however use both computers at the same time. The router will only forward the ports to the last computer that sent it a trigger. The second advantage is the security benefit of minimizing the time a port is opened. DD-WRT will maintain the open ports until there has been no traffic for 10 minutes so ill intentioned rabble on the WAN will only have a limited time to try and exploit these open ports. After that time the ports close automatically and can only be reopened by another trigger.

Port triggering has be around for awhile and has largely be replaced by uPnP. As a case in point AIM used as the example above now supports uPnP so it is not really necessary to do any of the above. But as some applications do not yet support uPnP there may still be a place for its use.

Port Forwarding using UPnP

Automatic

Using UPnP, applications that support it can automatically tell the router to open the port they're listening on and close them when their done listening. Automatic port forwarding with UPnP means you don't have to worry about IP address, ports, or anything like that. UPnP is the easiest way to Port Forward.

On router side

- On the Router's Web Interface, open tab **NAT/QoS -> UPnP**
- Select **enable UPnP Service**
- Select **Clear port forwards at startup**
- Click **Apply**

In Windows

- In Windows XP open **Control Panel -> Add/Remove Programs**
- On the left click on **Add/Remove Windows Components**
- Select **Networking Services** and click the **Details** button
- Mark the box next to **UPnP User Interface**. If available, also mark the box **Internet Gateway Device Discovery and Control Client**
- Click OK, then Click Next. The installer will execute.

- NOTE: You may need to insert your Windows CD

Windows ME and above include uPnP and it can be turned on in this manner. Windows 98 can have it installed from the Windows XP CD using the network configuration wizard.

Also, don't expect to see Ports opening immediately in the DD-WRT Web Interface -> NAT/QoS. Also, make sure the application that you are trying to Port Forward is in the list of Windows Firewall exceptions!

Manual

From *Network Neighborhood* or *Network Connections*, depending on your version of windows, you should see an icon for "Internet Gateway" or "Internet Connection" If not, reboot your router, followed by your computer.

Chances are the "Internet Connection" says it's disconnected, and if you double click on it nothing you expect will happen. If you right click on it, however, you can access its Properties. From there click on the "Settings" icon. Not that you cannot add ranges of ports, but you can do port mapping. From my experience, port mapping via UPnP has not been successful. I always set both internal and external to the same port and do normal port forwarding. This seems to work the best.

One thing to note: you don't need to enter an IP address for Manual UPnP Forwarding. You can simply enter the Windows Computer Name and your ports will be forwarded to your computer regardless of its current IP address. Other OSs that support UPnP *probably* allow the Samba name or other host name through some method. I'm not sure, though.

Port Forwarding using the console

Here I will specifically show port mapping, but the same thing can be done for port forwarding if you set the internal and external ports equal.

iptables is an entirely different topic, but here's the commands you'll want to use:

```
iptables -t nat -I PREROUTING -p tcp --dport <EXTERNAL_PORT> -j DNAT --to <INTERNAL_IP>:<INTERNAL
iptables -I FORWARD -p tcp -d <INTERNAL_IP> --dport <INTERNAL_PORT> -j ACCEPT [-s <EXTERNAL_IP>]
```

Replace:
<EXTERNAL_PORT> with the external port you wish to map
<INTERNAL_PORT> with the internal port you wish to map to
<INTERNAL_IP> with the internal IP you wish to map to
[-s <EXTERNAL_IP>] is completely optional (don't include the [ ]). This option enables you to limit the rule to only accept connections from <EXTERNAL_IP>, which can be a network name, a hostname, an IP address or an IP address range.

Example: To map port 81 to internal port 80 on 192.168.1.2 issue the command

```
iptables -t nat -I PREROUTING -p tcp --dport 81 -j DNAT --to 192.168.1.2:80
iptables -I FORWARD -p tcp -d 192.168.1.2 --dport 80 -j ACCEPT
```

The down side with this is that it's not as obvious as the other methods but you can use a program called Firewall Builder to make this simpler. This setting will also disappear on router reboot if you don't place these

commands inside of a Startup Script.

External links

http://www.portforward.com