Basics of TCP/IP, Switching, Routing and Firewalling.

Why this article ?

After reading the following question at least a gazillion times: My DCC is not working ... can anyone help me pls ??, i have been thinking about the cause or causes of this 'problem' for quite some time now.
Most of the people asking this question did everything alright configuring the Chat-Client or other applications they are using to connect to the internet. DCC or other network-services should be working fine, but they don't.

The most common reason for the problems those people are facing is, in my honest opinion, a not properly configured piece of the network. Due to this, the 'information' needed by the 'other side' (remote host) is not being transmitted over the network (In this case the network is the 'bad, bad' Internet.), or the packages send by the remote host are not reaching the network in which the requesting computer (local host) resides.

To have a better understanding why this is happening, one has to know what the different networking devices are doing with the network traffic they send and receive.

The network that people think they are using will (simplified) basically look like this:

Local host Router1 Internet Router2 Remote Host

Most people connecting to the internet nowadays are using a nice little thing they call a Router or DSL-Router. This is where some of the problems start... Is this nice little nifty device only a Router? Or is there more behind it?
To understand what this wonderful piece of technique is capable of we need to know a bit more about the different pieces a little home network is made of and how they work.

Now where to start? Imagine a user somewhere on this world, sitting behind a computer, pushing the power-button, waiting for the OS coming up, then starting his or her favourite browser (which i hope is Mozilla Firefox ;)) and starts typing www.google.de after doing this hitting the enter-key. What happens next...? The German starting-page of the searching machine google appears on the screen in front of that user. Now that's easy? Isn't it :)

Hmmm... was this really as easy as it looked like? Definitely not ;)

To understand what happens we will need a bit of theory. I'll try to keep this as brief as possible.

TCP/IP-Networking

An Ethernet local area network (LAN) is essentially a (logical) bus based broadcast network; though the physical implementation may use hubs (with a physical star topology). As one would

expect, broadcast LANs must deal with collisions; either by preventing them or detecting them and taking appropriate action. Token based LANs avoid collisions by only allowing one host at time to transmit (the host that currently has the token may transmit).

Standards that relate to LANs are primarily the IEEE 802.x series. For instance, 802.3 is the Media Access Control (MAC) standard for (Carrier Sense Multiple Access with Collision Detection) CSMA/CD (the Ethernet standard); while 802.5 is the MAC standard for Token Ring. Just above the MAC level is the Logical Link Control (802.2) standard and above that it the High Level Interface (802.1) standard.

Within a LAN, addressing is done with a MAC address. Between LANs (connected over the Internet (WAN (Wide Area Network)), having Routers in between) using TCP/IP, addressing is done using IP addresses. If you are lost at this point, keep reading because much of this will be explained below ;)

The OSI-Model

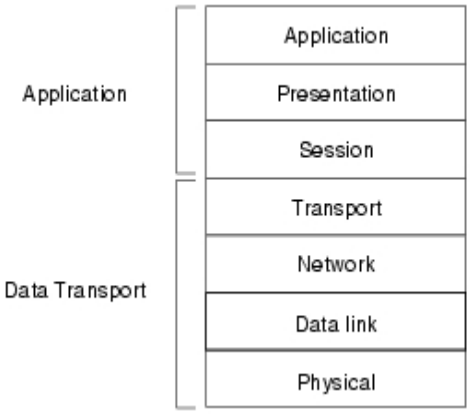
After TCP/IP was well-established and other networking protocols, such as DECnet and Novell's IPX were operational, the International Standardization Organization (ISO) developed the Open Systems Interconnection (OSI) seven layer reference model.

The following list details the seven layers of the Open System Interconnection (OSI) reference model:

OSI-Layer Name	Functional Description	Examples
• Layer 7 — Application	Interface between network and application software.	Telnet, HTTP, WWW-Browsers
• Layer 6 — Presentation	How data is presented, Encryption.	JPEG, ASCII
• Layer 5 — Session	Establishing maintaining and managing end-to-end bidirectional flows between endpoints.	Operating systems Application access
• Layer 4 — Transport	Reliable or unreliable delivery, Multiplexing.	TCP, UDP, SPX
• Layer 3 — Network	Logical addressing, which routers use for path determination	IP, IPX
• Layer 2 — Data link	Combination of bits into bytes, and bytes into frames. Access to media using MAC-address. Error detection and error recovery.	802.3/802.2 HDLC
• Layer 1 — Physical	Moving bits between devices. Specification of voltage, wire speed and cable pinouts.	EIA/TIA-232, V.35

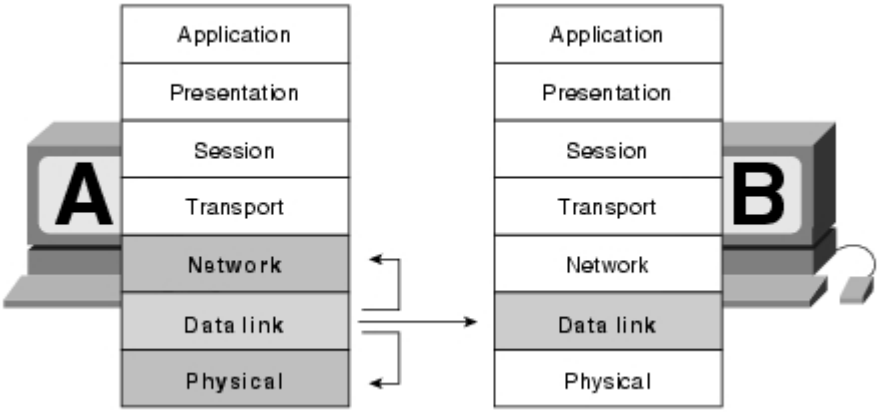
The seven layers of the OSI reference model can be divided into two categories: upper layers and lower layers.
The *upper layers* of the OSI model deal with application issues and generally are implemented only in software.
The *lower layer*s of the OSI model handle data transport issues. The physical layer and the data link

layer are implemented in hardware and software. The lowest layer, the physical layer, is closest to the physical network medium (the network cabling, for example) and is responsible for actually placing information on the medium.



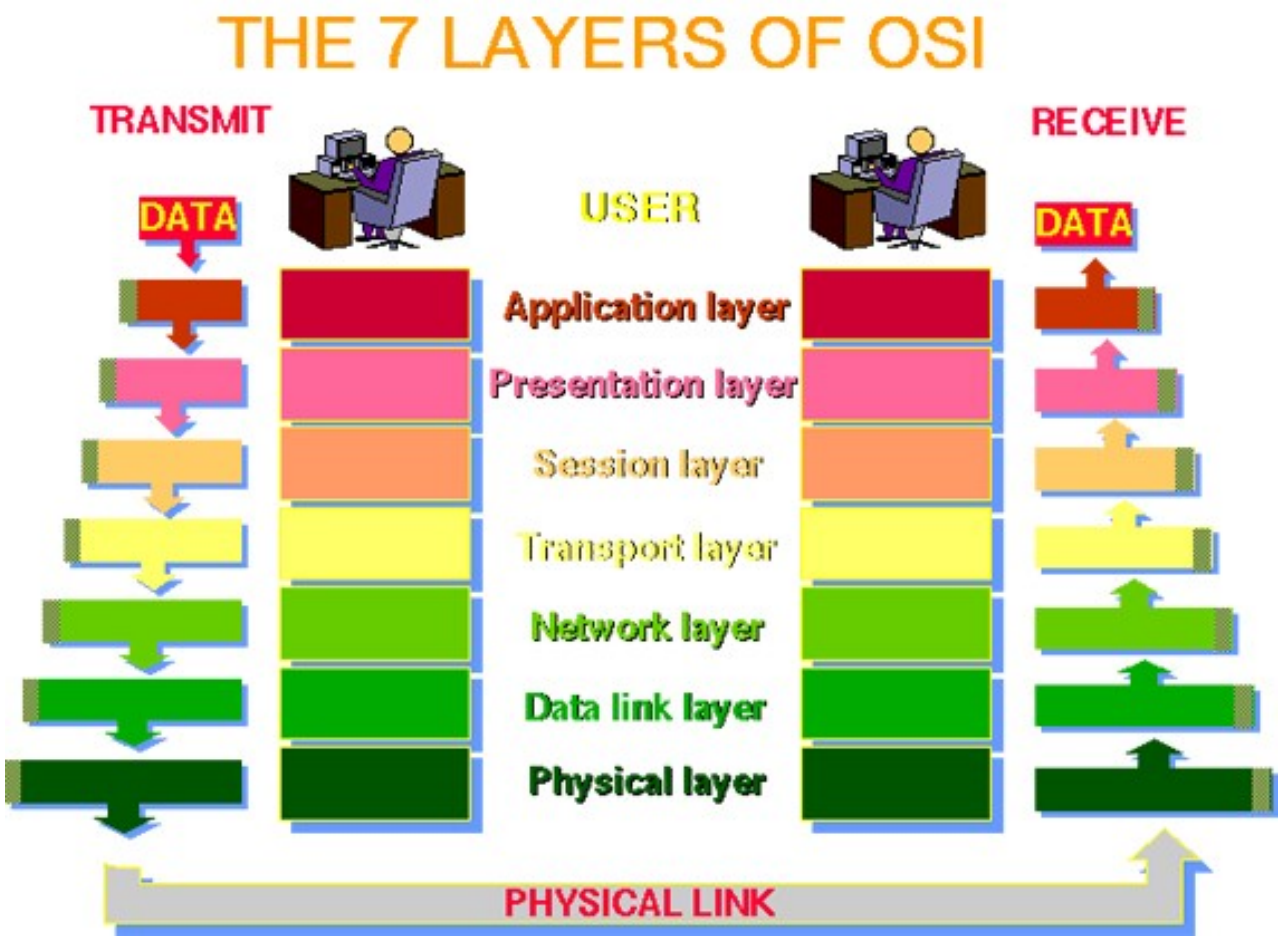
A wide variety of communication protocols exist. Some of these protocols include LAN (Local Area Network) protocols, WAN (Wide Area Network) protocols, network protocols, and routing protocols. *LAN protocols* operate at the physical and data link layers of the OSI model and define communication over the various LAN media. *WAN protocols* operate at the lowest three layers of the OSI model and define communication over the various wide-area media. *Routing protocols* are network layer protocols that are responsible for exchanging information between routers so that the routers can select the proper path for network traffic. Finally, *network protocols* are the various upper-layer protocols that exist in a given protocol suite. Many protocols rely on others for operation. For example, many routing protocols use network protocols to exchange information between routers.

A given layer in the OSI model generally communicates with three other OSI layers: the layer directly above it, the layer directly below it, and its peer layer in other networked computer systems. The data link layer in System A, for example, communicates with the network layer of System A, the physical layer of System A, and the data link layer in System B.



Now, lets go back to our little example. A user somewhere on this world pushes the power button of his computer. The computer starts his BIOS (Basic Input Output System) and does a POST (Power On Self Test) after doing a few more tests it searches for the OS (Operating System) and starts it. In other words, this computer is running through the OSI-Model from bottom to top. Starting with layer one and having reached layer five when the basics of the OS are running.
Now the GUI (Graphical User Interface) is started and this computer runs through the layers six and

seven. Why is this computer already running on layer seven of the OSI-Model? It's simple, hence: the GUI of Microsoft OS Windows is explorer.exe which is an application. Now the web browser is started, the user types www.google.de and hits the enter-key. This data is processed from top to bottom of the OSI-model and send over the network.



The Home Network

Now that we have come this far, it's time to break up our little home network into piece's and have a closer look at the single components.
Lets start with the thing your are probably sitting in front of, reading this document.

The Computer

As we already know this is a nice piece of technique that is capable of running on all seven layers of the OSI-model. Why is this important? You will see in a few.
This device allows you to run applications (OSI-layer 7) and is capable of sending data over the network using the NIC (Network Interface Card) which is running on OSI-layer 2.
Why has the NIC to be a OSI-layer 2 device? It's simple, it has a hardware decoded MAC-address and has, when it's not a wireless one, a piece of cable plugged in, connecting it to a hub or switch. It does not know anything about IP-addresses, port numbers and the protocol that is probably running in the little home network... TCP/IP.

TCP/IP (Transport Control Protocol/Internet Protocol) is a protocol suite that is implemented in the

OS you are using and is running on OSI-layer 3 and 4 the network and transport layer. TCP/IP is responsible for you having an IP-address and delivers the possibility to the computer to communicate with other systems using IP-addresses, a subnetmask, portnumbers and a default gateway.
The ability that your computer can run at OSI-layer 4 using TCP/IP makes it possible to use this device to run a firewall on it. Most of you will have at least one firewall running on the computer when having installed Windows XP SP2. This firewall is active by default and could be a possible cause for blocking traffic you'd rather like to get. (See we are getting to the point now ;))

Most users connecting to the internet don't know that one firewall, though it isn't a good one in my opinion, is already up and running, so they download a third-party product, like Zone Alarm. They install this firewall, and not really knowing what they are doing, they click on allow or block this or that application or traffic. Having a second, probably misconfigured, firewall up and running.

Due to the fact that most people know that viruses can be really nifty things and are, without proper knowledge, not easy to remove from an infected system. They purchase or download an antivirus product (like Bitdefender, NortonAV), not knowing that nowadays most of these software packages have a build in firewall that is installed and... up and running by default. To be sure that no virus can infect the computer, they use several products from different vendors, or they install different products from cd's/dvd's coming with computer magazins. Now already 3-4 (or even more) firewalls are running on the computer!

When scrolling back up to the little network at the beginning of this article i think this one already needs a revision;)

The next part of the home network is the network-cable connecting the computers NIC to the hub or switch. Could this piece of the network be responsible for the connectivity problems? Yup !! When it's broken or not connected, because it's a OSI-layer 1 piece of it ;)
OK let's presume the cable isn't causing the problems. The next piece of our network would be the hub or switch.

Hubs and Switches

What is the difference between a hub and a switch, how do they work and could they be responsible for parts of the data transmission not working like it should?
Most people think of hubs and switches being kind of a multiple socket outlet for connecting computers. Basically this is true for most of the devices used in small home/office networks.

When a hub receives a data packet, it 'shouts' it out of all ports, beside the one he received the packet on. So all computers connected to the hub are receiving that packet. It does not matter if that packet had that destination or not. How does a computer know if that little packet was determined for him or not? That's very simple... the computers NIC reads out the destination MAC-address in the packets header and accepts the packet when it's his MAC, otherwise the NIC drops the packet. Because of this a hub is working on layer 2 of the OSI-model. It cannot be responsible for not letting through any packets received by this device.
Due to the fact that hubs are causing a lot of unnecessary networking traffic switches were invented.

A switch 'learns' the MAC-addresses of the computers NIC's connected to his ports, writing them down in a MAC-address table. When a switch receives a data packet on one port, it reads out the destination MAC-address from the packet header and then forwards the packet to the port on which the NIC having this address is connected to. Thus reducing the network traffic a very good amount. The only traffic transported out of every port of the switch, apart from the port the switch is

receiving this packet, is the broadcast traffic the computers connected to the switch are causing. This 'one-to-one' communication still does not need IP-addresses, it is based on MAC-addresses only. That's why a switch is a OSI-layer 2 device too. It cannot be responsible for rejecting any network traffic and therefore can not be a part of the trouble shooting for users not be able to DCC or trying to use any other networking services.

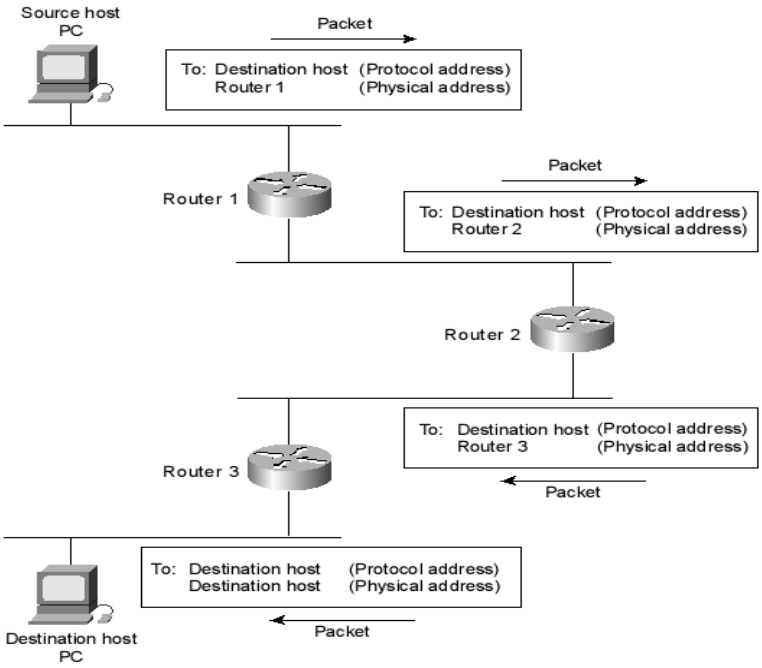
Now i can see a bit of glooming coming into your eye's. There's only one device left in our little home network, beside of that dang computer that already has a number of firewalls running on it, that could cause the problems we are so eager to solve;)) I can see you thinking... HAH! It's gotta be the Router!!!

What is Routing?

Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The primary difference between the two is that bridging occurs at Layer2 (the link layer) of the OSI-reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways.

Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork. In the context of the routing process, the latter of these is referred to as packet switching. Although packet switching is relatively straightforward, path determination can be very complex.

Switching algorithms is relatively simple; it is the same for most routing protocols. In most cases, a host determines that it must send a packet to another host. Having acquired a router's address by some means, the source host sends a packet addressed specifically to a router's physical (Media Access Control (MAC)-layer) address, this time with the protocol (network layer) address of the destination host. As it examines the packet's destination protocol address, the router determines that it either knows or does not know how to forward the packet to the next hop. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, however, it changes the destination physical address to that of the next hop and transmits



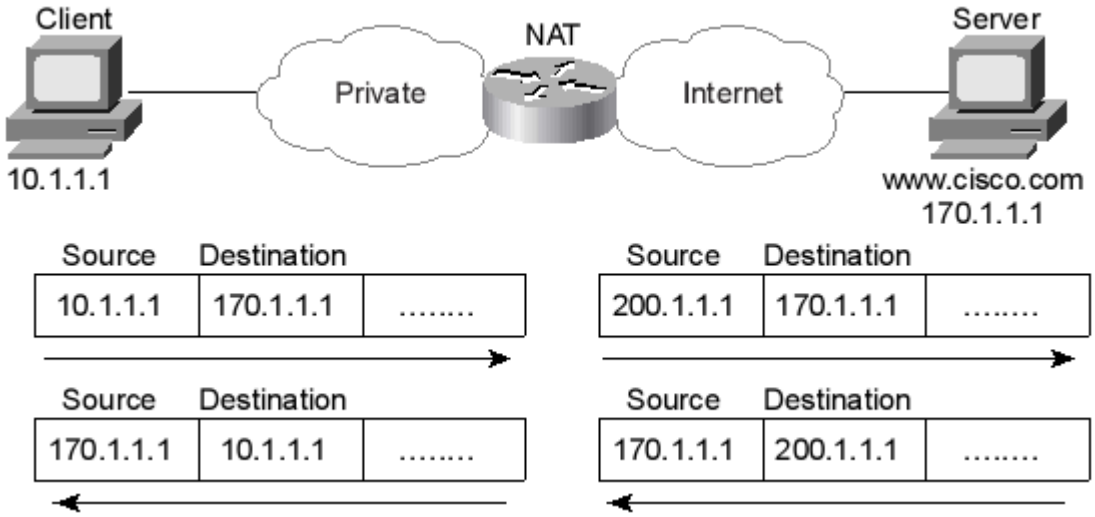
the packet.

The example above shows two hosts communicating with each other using three routers between them. If the three routers are part of the Internet, it will only work this way when both hosts have valid public IP-addresses assigned to them.

Network Address Translation

NAT, defined in RFC 1631, allows a host that does not have a valid registered IP address to communicate with other hosts through the Internet. The hosts might be using private addresses or addresses assigned to another organization. In either case, NAT allows these addresses that are not Internet-ready to continue to be used and still allows communication with hosts across the Internet.

NAT achieves its goal by using a valid registered IP address to represent the private address to the rest of the Internet. The NAT function changes the private IP addresses to publicly registered IP addresses inside each IP packet.



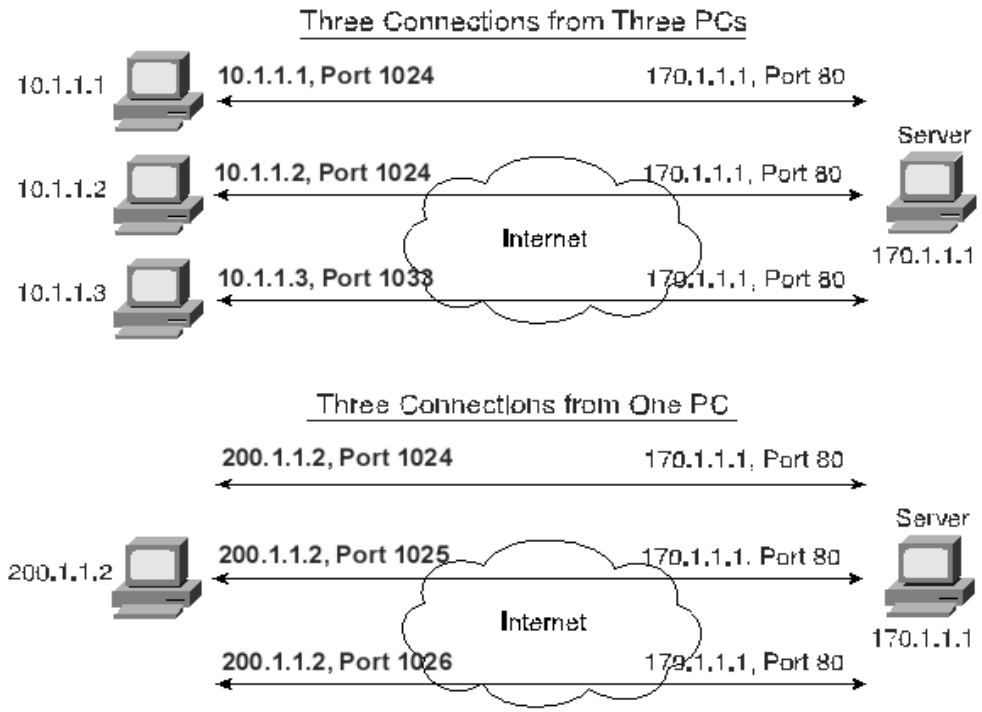
Notice that the router, performing NAT, changes the packet's source IP address when leaving the private organization and the destination address in each packet forwarded back into the private network. (Network 200.1.1.0 is registered in this figure) The NAT feature, configured in the router labeled NAT, performs the translation.

Overloading NAT with Port Address Translation (PAT)

Some networks need to have most, if not all, IP hosts reach the Internet. If that network uses private IP addresses, the NAT router needs a very large set of registered IP addresses. With static NAT, for each private IP host that needs Internet access, you need a publicly registered IP address.

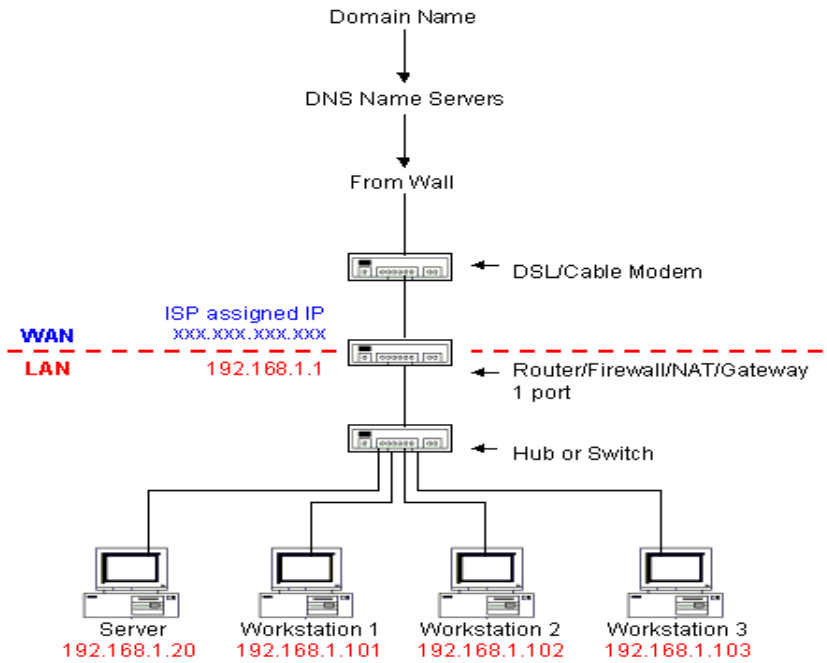
Overloading allows NAT to scale to support many clients with only a few public IP addresses. The key to understanding how overloading works is to recall how ports are used in TCP/IP.
The figure below details an example that helps make the logic behind overloading more obvious.

The top part of the figure shows a network with three different hosts connecting to a web server using TCP. The bottom half of the figure shows the same network later in the day, with three TCP connections from the same client. All six connections connect to the server IP address (170.1.1.1) and WWW port (80, the well-known port for web services). In each case, the server differentiates between the various connections because their combined IP address and port numbers are unique.



NAT takes advantage of the fact that the server really doesn't care if it has one connection each to three different hosts or three connections to a single host IP address. So, to support lots of inside private IP addresses with only a few global, publicly registered IP addresses, NAT overload uses Port Address Translation (PAT). Instead of just translating the IP address, it also translates the port number.
NAT overload can use more than 65,000 port numbers, allowing it to scale well without needing very many registered IP addresses, in many cases, like in small Office/Home Networks, needing only one.

Taking the device called a 'router' by most users apart, it contains different components. The following figure pictures the different components out. These are a hub/switch, the router and a DSL/Cable modem.



Now having a deeper insight in the basic things the router part of the device, connecting you to the Internet, in your small home network is doing, it´s time to ask the reader of this document a question ;) What protocol type is used by the router and what OSI-layer is that protocol running on?

The answer to this question should be quite simple for you now. Because of the fact, that the router performs a port and an address translation using NAT overload combined with PAT it has to be TCP/IP. This protocol suite works at OSI-layers 4 and 3, so there has to be a possibility to apply filter rules.

Applying filter rules can be done on the interfaces of the router. These filter rules can be applied on both interfaces, the internal and external. Network traffic on the interfaces can occur in two directions, incoming and outgoing.

Now having a deeper insight in what different networking devices are doing and how they work it's time to pick up the last topic.

Firewalling

Setting up a firewall seems to be easy and pretty straightforward for most users. It's nothing more than installing a piece of software, then allowing or blocking network traffic caused by applications running on the users computer by means of a few mouseclicks. Now the user feels safe behind the nice little brick wall running on his or her computer.

Before being able to understand a discussion of firewalls, it’s important to understand the basic principles that make firewalls work.

What is a Firewall?

A firewall is a system or group of systems that enforces an access control policy between two or more networks. The actual means by which this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one which exists to block traffic, and the other which exists to permit traffic. Some firewalls place a greater emphasis on blocking traffic, while others emphasize permitting traffic. Probably the most important thing to recognize about a firewall is that it implements an access control policy. If you don’t have a good idea of what kind of access you want to allow or to deny, a firewall really won’t help you. It’s also important to recognize that the firewall’s configuration, because it is a mechanism for enforcing policy, imposes its policy on everything behind it.

Why should i want a firewall?

The Internet, like any other society, is plagued with the kind of jerks who enjoy the electronic equivalent of writing on other people’s walls with spraypaint, tearing their mailboxes off, or just sitting in the street blowing their car horns. Some people try to get real work done over the Internet, and others have sensitive or proprietary data they must protect. Usually, a firewall’s purpose is to keep the jerks out of your network while still letting you get your job done.

What can a firewall protect against?

Generally, firewalls are configured to protect against unauthenticated inter- active logins from the

“outside” world. This, more than anything, helps prevent vandals from logging into machines on your network. More elaborate firewalls block traffic from the outside to the inside, but permit users on the inside to communicate freely with the outside. When it's a piece of hardware, the firewall can protect you against any type of network-borne attack if you unplug it.

What can't a firewall protect against?

Firewalls can’t protect against tunneling over most application protocols to trojaned or poorly written clients. Tunneling “bad” things over HTTP, SMTP, and other protocols is quite simple and trivially demonstrated.

Lastly, firewalls can’t protect against bad things being allowed through them. For instance, many Trojan Horses use the Internet Relay Chat (IRC) protocol to allow an attacker to control a compromised internal host from a public IRC server. If you allow any internal system to connect to any external system, then your firewall will provide no protection from this kind of attack.

What are the basic types of firewalls?

Basically there are three types of firewalls: Network Layer, Application Layer and Hybrid Firewalls. (Remember the 7 layered OSI-Model?)

A good example for a Network Layer firewall is a router. This device is capable of examining the packets header and reading out the information contained there.

The information that can be filtered on is: The source IP-Address, source port, destination IP-Address, destination port and protocol type (TCP,UDP aso). An Access Control List (ACL), containing different filter rules, could be implemented on the internal interface permitting or denying outgoing traffic based on this information.

The same could be done on the external interface permitting or denying the incoming traffic.

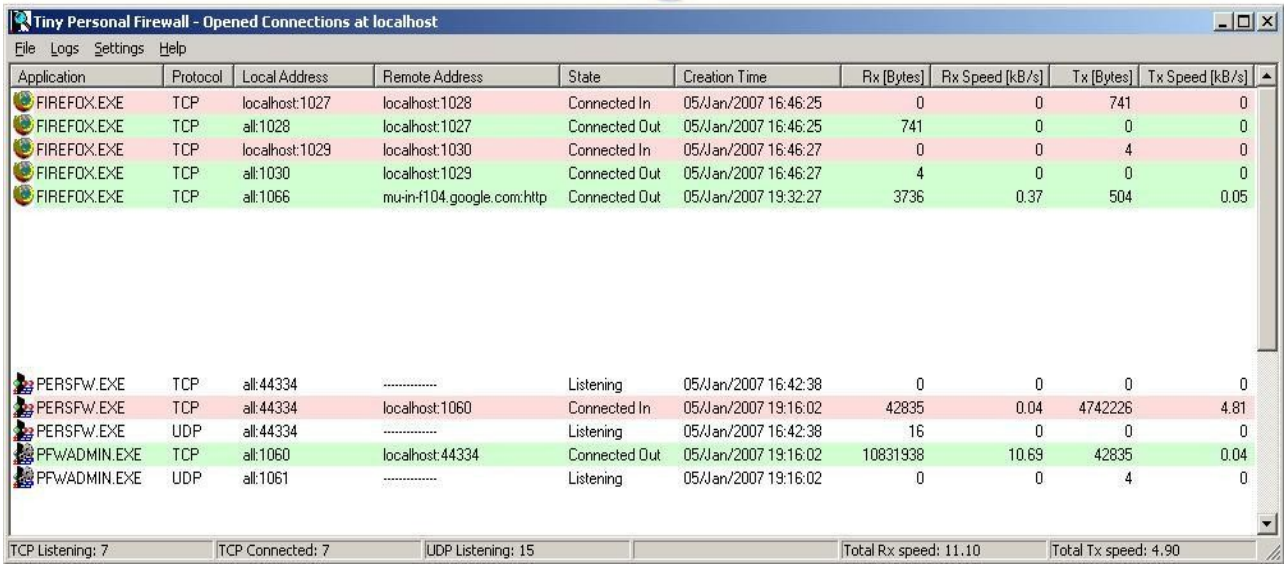
Application layer firewalls generally are hosts running proxy servers, which permit no traffic directly between networks. Application layer firewalls can be used as network address translators, since traffic goes in one “side” and out the other, after having passed through an application that effectively masks the origin of the initiating connection.

Most of you will ask now, proxy server, NAT... how can this be? My computer has only one NIC build in, or only one NIC connected to the switch and router connecting me to the Internet, i've got only one IP-Address, and this guy is talking about traffic between networks?? Still there is an application firewall running on my computer???

The answer is simple. The proxy server and NAT are running on the local loopback address (127.0.0.1) of the computer you are using. Have a look at the routing table used by your computer, typing route print in the command line, it should look something like this.

```
C:¥>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 90 99 32 bb 11 ...... corega Ether PCI-T Ethernet Adapter.
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.2       1
        127.0.0.0        255.0.0.0        127.0.0.1      127.0.0.1       1
      192.168.0.0    255.255.255.0      192.168.0.2    192.168.0.2       1
      192.168.0.2  255.255.255.255        127.0.0.1      127.0.0.1       1
    192.168.0.255  255.255.255.255      192.168.0.2    192.168.0.2       1
        224.0.0.0        224.0.0.0      192.168.0.2    192.168.0.2       1
  255.255.255.255  255.255.255.255      192.168.0.2    192.168.0.2       1
Default Gateway:       192.168.0.1
===========================================================================
```

And here is a screenshot of the connections open for Firefox just having refreshed the starting page of google.de and the firewall running on the computer at my apartment. You will see that the firewall and Firefox both are communicating over the local loopback interface of the computer.



Most firewalls now lie some place between network layer and application layer firewalls. As expected, network layer firewalls have become increasingly “aware” of the information going through them, and application layer firewalls have become increasingly “low level” and transparent.

What Application Services have to be Supported?

Before setting up the firewall on the router or computer, you have to decide what services are needed for the users and computers that are behind that firewall. Some of the most common

services are listed below.

Service	Definition
Basic TCP Protocols	Generic connected TCP Protocols, such as HTTP, POP3, Telnet, SSL, etc.
Other UDP	Generic UDP-Services such as DNS, NTP, TFTP, IKE, SNMP, etc.
FTP	Control connection on TCP Port 21, Data on TCP Port > 1024
Mail (SMTP)	Connect TCP Protocol on Port 25
H.323 (Netmeeting)	H.323 video conference protocol over UDP
RealAudio (RTSP)	Real-Time Streaming Protocol over UDP or TCP

For example, a reasonable list of desired services for many installations is: DNS, NTP, HTTP, FTP, and Telnet, plus SMTP and POP3 to the mail server only. For many of us using IRC a “few” ports have to be opened for this service too.

For a full list of services and the related ports (TCP/UDP) assigned by the IANA (Internet Assigned Numbers Authority) pls. refer to this webpage: http://www.iana.org/assignments/port-numbers

What´s a Port?

A “port” is a “virtual slot” in your TCP and UDP stack that is used to map a connection between two hosts, and also between the TCP/UDP layer and the actual applications running on the hosts. They are numbered 0–65535, with the range 0–1023 being marked as “reserved” or “privileged”, and the rest (1024–65535) as “dynamic” or “unprivileged”.

There are basically two uses for ports:

“Listening” on a port.

This is used by server applications waiting for users to connect, to get to some “well known service”, for instance HTTP (TCP port 80), Telnet (TCP port 23), DNS (UDP and sometimes TCP port 53).

Opening a “dynamic” port.

Both sides of a TCP connection need to be identified by IP addresses and port numbers. Hence, when you want to “connect” to a server process, your end of the communications channel also needs a “port”. This is done by choosing a port above 1024 on your machine that is not currently in use by another communications channel, and using it as the “sender” in the new connection.

Dynamic ports may also be used as “listening” ports in some applications, most notably FTP, and for us this one is true for DCC too. Ports in the range 0–1023 are almost always server ports. Ports in the range 1024–65535 are usually dynamic ports (i.e., opened dynamically when you connect to a server port). However, any port may be used as a server port, and any port may be used as an

“outgoing” port.

So, to sum it up, here’s what happens in a basic connection (scroll back to the overloading NAT with PAT and have another look at the 3 PC´s connecting to one server to have a picture of it):

At some point in time, a server application on host 170.1.1.1 decides to “listen” at port 80 (HTTP) for new connections.

You (10.1.1.3) want to surf to 170.1.1.1, port 80, and your browser issues a connect call to it.

The connect call, realising that it doesn’t yet have a local port number, goes hunting for one. The local port number is necessary since when the replies come back some time in the future, your TCP/IP stack will have to know to what application to pass the reply. It does this by remembering what application uses which local port number. (This is very, very much simplified, no flames from TCP/IP experts and programmers, please.)

Your TCP stack finds an unused dynamic port, usually somewhere above 1024. Let’s assume that it finds 1033.

Your first packet is then sent, from your local IP, 10.1.1.3, port 1033, to 170.1.1.1, port 80.

The server responds with a packet from 170.1.1.1, port 80, to you, 10.1.1.3, port 1033.

This procedure is actually much longer than this, but it points out the basics of your computer contacting the HTTP-service running on 170.1.1.1, “listening” on port 80.

What are Listening Ports ?

Suppose you did “netstat -a” on your machine and ports 1025 and 1030 showed up as LISTENing. What do they do? Right, let’s take a look in the assigned port numbers list.

blackjack 1025/tcp network blackjack
iad1 1030/tcp BBN IAD

Wait, what’s happening? Has my workstation stolen my VISA number and decided to go play blackjack with some rogue server on the internet? And what’s that software that BBN has installed? This is NOT where you start panicking. In fact, this question has been asked maybe a gazillion times, and every time it’s been answered. Not that THAT keeps people from asking the same question again.
If you are asking this question, you are most likely using a windows box. The ports you are seeing are (most likely) two listening ports that the RPC subsystem opens when it starts up.
This is an example of where dynamicly assigned ports may be used by server processes.
Applications using RPC will later on connect to port 135 (the netbios “portmapper”) to query where to find some RPC service, and get an answer back saying that that particular service may be contacted on port 1025.
Now, how do we know this, since there’s no “list” describing these ports? Simple: There’s no substitute for experience. And using the mailing list search engines also helps a hell of a lot.

How do i determine what Service the Port is for?

Since it is impossible to learn what port does what by looking in a list, how do i do it?
The old hands-on way of doing it is by shutting down nearly every service/daemon running on your machine, doing netstat -a and taking note of what ports are open. There shouldn’t be very many listening ones. Then you start turning all the services on, one by one, and take note of what new ports show up in your netstat output.

Another way, that needs more guess work, is simply telnetting to the ports and see what comes out. If nothing comes out, try typing some gibberish and slamming Enter a few times, and see if something turns up. If you get binary garble, or nothing at all, this obviously won't help you. :-)
However, this will only tell you what listening ports are used. It won't tell you about dynamically opened ports that may be opened later on by these applications.
There are a few applications that might help you track down the ports used. On Unix systems, there's a nice utility called lsof that comes preinstalled on many systems. It will show you all open port numbers and the names of the applications that are using them. This means that it might show you a lot of locally opened files as well as TCP/IP sockets. Read the help text. :-)
On windows systems, nothing comes preinstalled to assist you in this task. (What's new?) There's a utility called "Inzider" which installs itself inside the windows sockets layer and dynamically remembers which process opens which port. The drawback of this approach is that it can't tell you what ports were opened before inzider started, but it's the best that you'll get on windows (to my knowledge). http://ntsecurity.nu/toolbox/inzider/

What Ports are safe to pass through a Firewall?

ALL.
No, wait, NONE.
No, wait, uuhhh... I've heard that all ports above 1024 are safe since they're only dynamic??
No. Really. You CANNOT tell what ports are safe simply by looking at its number, simply because that is really all it is. A number. You can't mount an attack through a 16-bit number.
The security of a "port" depends on what application you'll reach through that port.
A common misconception is that ports 25 (SMTP) and 80 (HTTP) are safe to pass through a firewall. *meep* WRONG. Just because everyone is doing it doesn't mean that it is safe.
Again, the security of a port depends on what application you'll reach through that port.
If you're running a well-written web server, that is designed from the ground up to be secure, you can probably feel reasonably assured that it's safe to let outside people access it through port 80. Otherwise, you CAN'T.
The same it true for "inside" users visiting a compromised website on port 80 (HTTP). This website will send you the virus, or other NOT wanted data, to the application that requested this data on the "dynamically" assigned port on the local computer.
The problem here is not in the network layer. It's in how the application processes the data that it receives. This data may be received through port 80, port 666, a serial line, floppy or through singing telegram. If the application is not safe, it does not matter how the data gets to it. The application data is where the real danger lies.
If you are interested in the security of your application, go subscribe to bugtraq http://www.securityfocus.com or try searching their archives.
This is more of an application security issue rather than a firewall security issue. One could argue that a firewall should stop all possible attacks, but with the number of new network protocols, NOT designed with security in mind, and networked applications, neither designed with security in mind, it becomes impossible for a firewall to protect against all data-driven attacks.

How do most "modern" application Firewalls work?

After having installed an application Firewall on your computer this piece of software, that we know now, is a proxy-server performing NAT/PAT on the local loopback interface of the computer, normally has only one single filter rule.

Deny Any-Direction Any-Local-IP Any-Local-Port Any-Remote-IP Any-Remote-Port

This means, that the Firewall will DENY all traffic from all local Ports, all local IP´s to any remote

IP´s and any remote Ports (Protocols/Services) and visa versa. Thus letting no traffic out of the computer nor letting any traffic into the machine. This IS pretty save? Isn't it?

When you start your Web Browser, let's say this one is Mozilla Firefox, located at "c:\Application Directory\firefox.exe", this application will try to reach the starting page, for example www.google.com, you use on the Internet on port 80. What happens? The firewall will ask you if you want to permit or deny traffic from this application and if the filter rule, that´s going to be created, should be remembered or not. You DO want to surf the web of course, now clicking on Permit and Remember this Rule, never ask me again. Bingo!!!... you´r online, you are able to surf the web, you can go everywhere you want to :-)

O.K. a new filter rule has been created on the firewall. How does that one look like?
Because of the fact that the application firewall does´t know anything about ports, protocols, IP-addresses and directions that are required for this connection, the set of filter rules running on the firewall will now look like this.

Permit Any-Direction Any Any Any Any "c:\Application Directory\firefox.exe" MD5-Checksum
Deny Any-Direction Any-Local-IP Any-Local-Port Any-Remote-IP Any-Remote-Port

!!! Yes, DO reread that **Permit-Rule** a few times !!! And, YES, think that one over !!!

This filter rule permits the application firefox.exe, located in the "Application Directory" on the hard disk labelled c:\, to send data from any local port and IP-address to any remote IP-Address and to any remote port.
This part of the filter rule allows the WWW-Browser to send a request for data (outgoing IP-traffic) from a local (dynamically assigned >1024) port (this is still what we like to have) to any remote host (that´s o.k. too www.google.com and www.cisco.com will have a different IP-addresses) regardless to the port (service) that is requested. And this isn't what we like to have!! The requested service was HTTP, so the port-number the remote host is listening on is 80. The only outgoing IP-traffic we need, in this case, will go to port 80.
The IP-traffic that is send back will come from the remote host, having the source port 80, and having our dynamically assigned local port and the external IP-address of our router, as destination. This traffic will be handled by the NAT/PAT part of the router, and firewall, passing it to the port the application that requested it is listening on.
It also allows this application to receive data from any remote IP-address and any remote port to any local IP-Address and any local port.
This would only be needed if the application, in this case firefox.exe, is listening on a certain port, running a server-service. So this part of the filter rule is not needed at all.

The MD5-Checksum is a so called "hash-value" that is calculated on the basis of the current "state" of the application. This value looks something like this: "7EE7897AC29382BU89K2346", and this value is specific for any application running on your computer.
The first thing you are protected from is, an "attacker" changing a part of the application and this causing the MD5-Checksum to change.(Have you ever wondered why the firewall isn't asking you to create a new filter rule after loading a new script-file into your mIRC?)

The firewall will ask you if the changed application, still having the same name and still located at the same place on the hard disk, should be allowed to connect to the internet in the future.
Most users will think that this is o.k., because they don´t exactly know what was the cause, and will allow the application, now running some mallware too, to connect to the internet in the future. The piece of mallware could now use ANY remote port it want´s, connecting to ANY remote IP-Address, receiving data on ANY local port and IP-Address.

The second thing you are protected from is: an “attacker” installing a new application that tries to connect to the internet. If that application tries to do so, the firewall will ask you if that application should be allowed to do so or not.

The third thing that the firewall is protecting you from is: an “attacker” trying to connect to an application that has no entry in the filter rules list that the firewall is working with. Cause of the fact that the firewall still has his last rule: Deny anything else coming in or going out of the machine that i do NOT know of, and/or give the user a warning if just this happens.

Services, Port-number, Type, Direction overview

This is only a short overview of the most common services used by most of the people using a computer connecting them to the internet, the protocol-type, and the direction that should be opened in the firewall running on the computer used for that.

Service	**Port-number**	**Type**	**Direction, Incoming**	**Outgoing**
DNS	53	UDP		X
HTTP/HTTPS	80/443	TCP		X
FTP	20/21	TCP/UDP		X
TELNET	23	TCP/UDP		X
POP3	110	TCP		X
SMTP	25	TCP		X
IRCU	6665-6669/7000	TCP/UDP		X
IDENT	113	TCP	X	
Private File Service	59	TCP/UDP	X	X
NNTP	119	TCP/UDP		X
NTP	123	TCP/UDP		X
Remote Desktop	3398	TCP/UDP	X	X

A complete list of ports registered at the IANA can be found here:
http://www.iana.org/assignments/port-numbers

The ports for “incoming traffic” should ONLY then be opened if YOUR computer has to provide this particular service.

For example: If your using your Web-Browser to connect to the internet and to download files using FTP, you should allow outgoing traffic, TCP/UDP on port 20,21 and TCP on port 80 and 443 for this particular application only.
For your E-Mail client you should allow outgoing traffic, TCP only on port 25 and 110, allowing

you to send and receive E-Mails.
Last but not least, here are three useful links for people having problems with DCC and mIRC:

http://www.ircbeginner.com/ircinfo/dcc-trouble.html
http://www.mirc.org/dcchelp.html
http://www.mircscripts.org/showdoc.php?type=tutorial&id=2355