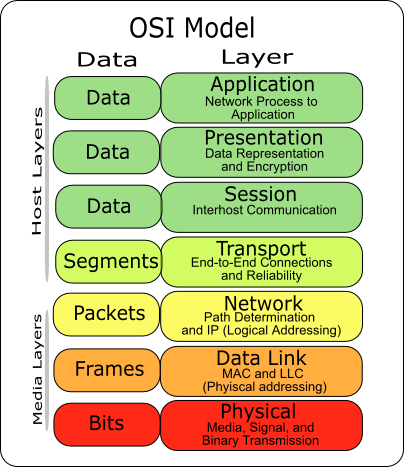
For more technical details I suggest you read up on the OSI model and the TCP/IP model. Nat operates at later 3 of the OSI model (the network layer, IP in this case) and proxies generally operate at layer 7 (the Application layer, HTTP or whatever you're proxying).

NAT and proxies are also trying to solve slightly different problems. NAT is hiding a number of private IPs behind a smaller number of public IPs to alleviate the shortage of IP addresses, while proxies are creating a network "choke point" for security, auditing or performance reasons

**Difference between NAT vs PROXY vs ROUTER**

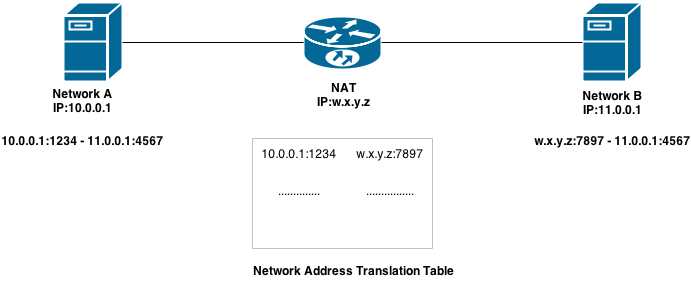
To understand the subtle and not so subtle differences between the three, one needs to know the OSI model and where in that model these operate.  
  
Here is a brief overview of the OSI model.

[](http://1.bp.blogspot.com/-yPHtWF43D8c/TY-PxT-KqNI/AAAAAAAAAAo/1OaBHHrsEKE/s1600/osi-model-7-layers.png)

The left column defines the data model for each layer. For example the smallest data block at the Network layer is a packet and the smallest data block at the transport layer is a segment.  
  
Protocols like TCP and UDP function at the Transport layer, whereas IP and routing works at the Network layer.  
  
Now coming to what operates in which layer:  
  
**Router**  
  
See the diagram below:

[](http://2.bp.blogspot.com/-lQdYMOcsknc/U7Ut1JPOPOI/AAAAAAAADzQ/hKvb94Rr8oo/s1600/NATvsPROXYvsROUTER_1.png)

 There are two machines, each on a different network connected through a Router. If machine A wants to communicate to machine B it needs to create a TCP connection. When the connection is created, both the machines A and B are unaware of the presence of a router in between. What this means is, in both the machines if we look up the IP:PORT combination it would be 10.0.0.1:1234 - 11.0.0.1:4567 i.e router is transparent.  
  
In this case router does not participate at the Transport (TCP) layer and just acts as a relay for datagram packets. All it knows is where to send the packet. It does not modify the packets and does not require the response to come back through it. In fact, if there is another router somewhere in between these networks which often is the case if we assume two machines connecting over the internet, then the response may come back through some other route. Router is Network layer (TCP) protocol unaware.  
  
NAT (Network Address Translator)  
  
  
Now NAT is nothing but an intelligent router. Since we all know that IPv4 ips are in short supply and these days even something as dumb as a fridge has an IP. It is also a fact that most of the devices are consumers and not producers i.e they do not need a publicly resolvable IP. For eg : all the work stations inside a building need not have a public IP assigned to them.  
  
This is where NAT comes into play. What NAT does is hide the machines behind it from the internet. All the machines go through NAT to access the outside world.  
  
 See the image below:

[](http://2.bp.blogspot.com/-d-Cu1_8kRW8/U7VnluG9ppI/AAAAAAAADzg/yrQRiww56Eg/s1600/Copy+of+NATvsPROXYvsROUTER_2.png)

 The machine A still thinks that it is talking to machine B directly (10.0.0.1:1234 - 11.0.0.1:4567). What the NAT does is replace the Source:Port Header of the packet it received from A with its own IP (w.x.y.z) and its own random port (7897). When the packet reaches machine B it thinks it came from the NAT IP.  
  
Since a response is always to the source, it tries to send the response packet back to the NAT at the same port as was overridden in the header(7897). When the NAT had received the packet from A it had assigned it a random port (7897) and kept that in a table called NAT table. So when the response comes back from Machine B it just does a reverse look up in the same table and forwards it to the desired recipient (Machine A). This way more than one machines can access the internet through NAT.  
  
One important point to note here is that at each layer of OSI model there is a checksum to determine if the packet/segment is valid. The same applies here. If the NAT changes the source IP:PORT combination, it need to recalculate the checksum again both at the Transport as well as Network layer. This leads to some additional work for the NAT.  
  
**Proxy:**    
  
A proxy works at the Transport layer and is aware of the protocol. Its not transparent in nature. It actually creates two connections one each with source and destination. Machine A does not even know about machine B. For machine A Proxy is the only thing it talks to and does not care how and where the proxy gets its data.  
  
See the image below:

[](http://4.bp.blogspot.com/-35rvZeHbN-k/U7Y_mN0zP_I/AAAAAAAADz0/YEHVyzxgA0Y/s1600/Copy+of+NATvsPROXYvsROUTER_3.png)

In the above picture Server A does not even know the IP of Server B. All it knows is the IP and port of the proxy server. The proxy server creates two connections, one with Server A and one with Server B. This happens at the Transport layer. Similarly server B does not even know the IP of server A

Examples of proxy servers are load balancers like HAProxy, Nginx, Apache, AWS ELB, F5 BIG-IP. They hide the backend from the outside world and do lot of nifty stuff like load balancing. optimization, etc.