- How do I monitor the activity of sockets?

**netstat**

**Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, netstat displays active TCP connection**

**a : Displays all active TCP connections and the TCP and UDP ports on which the computer is listening**

How would I put my socket in non-blocking mode?

**\* Set socket to non-blocking \*/   
  
if ((flags = fcntl(sock\_descriptor, F\_GETFL, 0)) &lt; 0)   
{   
/\* Handle error \*/   
}   
  
  
if (fcntl(socket\_descriptor, F\_SETFL, flags | O\_NONBLOCK) &lt; 0)   
{   
/\* Handle error \*/**

a [socket](http://en.wikipedia.org/wiki/Internet_socket) that allows access to [packet](http://en.wikipedia.org/wiki/Packet) [headers](http://en.wikipedia.org/wiki/Header_%28information_technology%29) on incoming and outgoing packets. Raw sockets are usually used at the [transport](http://en.wikipedia.org/wiki/Transport_layer) or [network](http://en.wikipedia.org/wiki/Network_layer) layers.

Usually raw sockets always receive packets with the header included (as opposed to non-raw sockets, which strip the header and receive just the [payload](http://en.wikipedia.org/wiki/Payload)). Whether or not a header is automatically prepended to outgoing packets is usually a configurable socket option.

How can I be sure that a UDP message is received?

design your protocol to expect a confirmation back from the destination when a message is received. Of course is the confirmation is sent by UDP, then it too is unreliable and may not make it back to the sender. If the sender does not get confirmation back by a certain time, it will have to re-transmit the message, maybe more than once. Now the receiver has a problem because it may have already received the message, so some way of dropping duplicates is required. Most protocols use a message numbering scheme so that the receiver can tell that it has already processed this message and return another confirmation

or use sliding window algo.

**Internetwork Packet Exchange** (**IPX**) is the [OSI-model](http://en.wikipedia.org/wiki/OSI_model) [Network layer](http://en.wikipedia.org/wiki/Network_layer) [protocol](http://en.wikipedia.org/wiki/Protocol_%28computing%29) in the [IPX/SPX](http://en.wikipedia.org/wiki/IPX/SPX) [protocol stack](http://en.wikipedia.org/wiki/Protocol_stack).

* Logical networks are assigned a unique 32-bit [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal) address in the range of 0x1 - 0xFFFFFFFE.
* Hosts have a 48-bit node address which by default is set to the network interface card's [MAC address](http://en.wikipedia.org/wiki/MAC_address). The node address is appended to the network address to create a unique identifier for the host on the network.
* **A** [**local-area network (LAN)**](http://www.webopedia.com/TERM/R/local_area_network_LAN.html) **whose** [**topology**](http://www.webopedia.com/TERM/R/topology.html) **is a ring. That is, all of the** [**nodes**](http://www.webopedia.com/TERM/R/node.html) **are connected in a closed loop. Messages travel around the ring, with each node** [**reading**](http://www.webopedia.com/TERM/R/read.html) **those messages addressed to it. One of the advantages of ring networks is that they can span larger distances than other types of** [**networks**](http://www.webopedia.com/TERM/R/network.html)**, such as bus networks, because each node regenerates messages as they pass through it.**

A **bus network Topology** is a [network architecture](http://en.wikipedia.org/wiki/Computer_network) in which a set of [clients](http://en.wikipedia.org/wiki/Client_%28computing%29) are connected via a shared communications line, called a [bus](http://en.wikipedia.org/wiki/Computer_bus).

Bus networks are the simplest way to connect multiple clients, but often have problems when two clients want to transmit at the same time on the same bus. Thus systems which use bus network architectures normally have some scheme of collision handling or collision avoidance for communication on the bus, quite often using [Carrier Sense Multiple Access](http://en.wikipedia.org/wiki/Carrier_Sense_Multiple_Access) or the presence of a [bus master](http://en.wikipedia.org/wiki/Bus_master) which controls access to the shared bus resource eg. 10base5 10base2

**ping" tool enables End-to-End reachability status to be investigated; its mechanism is based on Internet Control Message Protocol (ICMP) Echo Request/Reply messages. When the source sends an Echo Request message to the destination, the destination sends back an Echo Reply message to the source. In this way, the communication reachability is investigated. The "ping" tool is a good one; it is simple and can collect elementary End-to-End status information. However, it cannot be used for hop-by-hop based status investigation, and it cannot locate bottlenecks and problems in a communication path.**

**"traceroute" [**[**5**](https://www.isoc.org/inet2000/cdproceedings/1e/1e_3.htm#r5)**] tool enables a hop-by-hop based investigation to be performed; it finds IP address information of nodes along the communication path.** **"traceroute" mechanism is based on User Datagram Protocol (UDP) probe packets and ICMP reply messages. At first, the source prepares UDP probe packets whose TTL field values in the IPv4 header are controlled and sends them to the destination. The TTL values of probes are started from 1 and are increased one by one. If the TTL value is smaller than the number of hops to the destination, it expires on an intermediate node on the way to the destination and the probe does not reach the destination. From such a node, an ICMP message is issued to the sender of the probe message to notify the sender that the TTL has expired, and the ICMP message becomes a reply to the probe. The source IP addresses of the replied ICMP messages provide IP address information of nodes along the communication path. By using this mechanism, reachability to each replied intermediate node is verified.**