

# Protocol Design

## Introduction

A client and server need to exchange information via messages. TCP and UDP provide the transport mechanisms to do this. The two processes also have to have a protocol in place so that message exchange can take place meaningfully.

## Protocol Design

Some parameters are

- Broadcast vs point to point.  
Broadcast must be UDP or the more experimental MBONE. Point to point could be either TCP or UDP.
- Stateful vs stateless.  
Is it reasonable for one side to maintain state about the other side? It is often simpler to do so, but what happens if something crashes?
- Reliable vs unreliable.
- Replies needed.  
If a reply is needed, how do you handle a lost reply? Timeouts may be used.
- Data format  
MIME or byte level.
- Bursty or steady stream.  
Ethernet and the Internet are best at bursty traffic. Steady stream is needed for video streams. Quality of Service (QoS) may be needed, such as from ATM
- Synchronisation required. Does the data need to be synchronised with anything? e.g. video and voice.
- Standalone or library.

## Version control