UDP是无连接的，因此并没有TCP那样的三次握手和四次挥手过程。这使得UDP具有更高的传输效率，特别适合于在高并发场景下使用。

要使用UDP建立高并发的网络框架，可以尝试以下步骤：

1. 创建UDP套接字：使用socket函数创建一个UDP套接字。
2. 绑定地址和端口：使用bind函数将套接字绑定到一个特定的地址和端口。
3. 设置非阻塞模式：将套接字设置为非阻塞模式，这样在没有数据可读或者无法立即发送数据时，套接字的操作不会阻塞线程。
4. 使用epoll或者select进行IO多路复用：使用epoll或者select可以让一个线程同时处理多个套接字的IO事件，从而提高并发性。
5. 使用connect函数"连接"远程地址：当从一个地址接收到数据时，可以使用connect函数将套接字"连接"到这个地址，这样以后就可以直接用这个套接字向这个地址发送数据，而不必每次都指定地址。
6. 接收和发送数据：使用recvfrom和sendto函数接收和发送数据。

在这个过程中，"连接"的含义并不是真正的网络连接，而是将套接字和一个特定的远程地址关联起来，使得以后通过这个套接字发送数据时可以不必指定远程地址。

这样，就可以使用UDP建立一个高并发的网络框架了。

不过，请注意，虽然UDP的效率高，但是它不提供可靠性保证，数据包可能会丢失、重复或者乱序。如果需要这些特性，就需要自己在应用层实现。