将主机间交付扩展到进程间交付被称为 运输层的多路复用和多路分解

multiplexing at sender:

handle data from multiple sockets, add transport header (later used for demultiplexing)

demultiplexing at receiver:

use header info to deliver received segments to correct socket

**多路分解是怎样工作的？**

host主机从网络层接受IP数据报

每个数据报有目的和源头IP地址

每个数据报携带一个传输层的报文段(segment)

每个报文段有目的和源头的端口号

host使用IP地址和端口号将报文段(segment)引领到适当的套接字。

**无连接的多路分解：**

when host receives UDP segment:

checks destination port # in segment

directs UDP segment to socket with that port #

UDP套接字是由二元组标识的（目的IP和目的端口）

如果两个UDP报文段有不同的源IP地址和、或源端口号，但具有相同的目的IP地址和端口号，则这两个报文段将通过相同的目的套接字被定向到相同的目的进程。

源端口号用作“返回地址”的一部分

**面向连接的多路分解：**

TCP套接字是由一个四元组（源IP地址，源端口号，目的IP地址，目的端口号）来标识的。

服务器主机可以支持很多并行的TCP套接字，每个套接字与一个进程相联系，并由其四元组来标识每个套接字。当一个TCP报文段（segment）到达主机时，四元组被用来将报文段定向（分解）到相应的套接字。