**서버 프로그래밍**

**류은석 교수님**

[수 5, 6, 7, 8]

201233381 컴퓨터공학과 3학년 류원석

201233453 컴퓨터공학과 3학년 강수민

201233504 컴퓨터공학과 3학년 심준우

201233546 컴퓨터공학과 3학년 정우주

1. 기본 구조

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Transmission  Module | | | | | |  | Socket (Include Option) | |  |  |  | Network | | | | | | |  |  |  | Socket (Include Option) | |  | Transmission  Module | | | | | |  |  |
|  |  |  | | | | | |  |  | |  |  |  |  | | | | | | |  |  |  |  | |  |  | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  | |  |  |  |  | | | | | | |  |  |  |  | |  |  |  |  |  |  |  |  |  |
|  |  | Compress  Module | | | | | |  |  | |  |  |  |  | | | | | | |  |  |  |  | |  | Decompress  Module | | | | | |  |  |
|  |  |  | | | | | |  |  | |  |  |  |  | | | | | | |  |  |  |  | |  |  | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Encoding Module | | | | | |  | Multi Thread (Scheduling) | |  |  |  |  | | | | | | |  |  |  | Request Module | |  | Decoding Module | | | | | |  |  |
|  |  |  | | | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |
|  |  | File Read  Module | | | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | File Write  Module | | | | | |  |  |
|  |  |  | | | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

파일 전송을 위한 서버 ~ 클라이언트의 기본 구조를 간단한 블록도로 나타내면 위와 같이 나타난다. 서버에서는 프로그램이 파일을 처리할 수 있도록 파일을 읽는다. 읽은 파일을 네트워크 바이트과 같이 적절히 인코딩 후 압축을 진행한다. 인코딩과 압축을 거친 데이터는 소켓을 통해 네트워크를 거쳐 클라이언트에게 전송된다. 이때 서버는 n개의 클라이언트의 요청을 수용하고, 각 모듈들을 효과적으로 사용하기 위해 멀티쓰레드를 지원한다. 클라이언트는 전송된 데이터를 압축을 해제하고 디코딩을 한 후 클라이언트의 저장 공간에 데이터를 저장한다.

1. Parallel TCP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Transmission  Module | | | | | |  | Socket 1¸2¸ **˙˙˙** n (Include Option) | |  |  |  | Network | | | | | | |  |  |  | Socket 1¸2¸ **˙˙˙** n (Include Option) | |  | Transmission  Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | **˙˙˙** | **˙˙˙** |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Compress  Module | | | | | |  |  |  |  | Decompress  Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Encoding Module | | | | | |  | Multi Thread (Scheduling) | |  |  |  |  |  |  | Request Module | |  | Decoding Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | File Read  Module | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | File Write  Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

n개의 소켓을 이용해 서버 ~ 클라이언트의 connect를 설정한다. 연결된 n개의 소켓을 통해 데이터를 주고받는다. 이 방법은 기존의 1개의 connect에서 RTT의 지연이나 Packet loss 상황에서 fast recovery를 통해 진행할 경우(Threshold \* 1/2)보다 효율적인 속도에 빨리 다다를 수 있다는 장점(Threshold \* (1/2)^n)이 있다.

1. Reliable UDP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Transmission  Module | | | | | |  | Socket (Include Option) | |  |  |  | Network | | | | | | |  |  |  | Socket (Include Option) | |  | Transmission  Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Compress  Module | | | | | |  |  |  |  |  |  |  |  | Decompress  Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Encoding Module | | | | | |  | Multi Thread (Scheduling) | |  |  |  |  |  |  | Request Module | |  | Decoding Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | File Read  Module | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | File Write  Module | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

UDP의 장점은 TCP와 달리 connect 과정이 없으며, 파일 재전송을 하지 않아 속도가 빠르다는 점이다. 하지만 전송되는 파일의 신뢰성은 보장할 수 없다. 따라서 UDP에서 신뢰성을 보장할 수 있는 최소의 기능을 덧붙여준다. 보낸 데이터의 순서, 도착, 무결성을 보장한다면, 속도와 신뢰성을 갖춘 파일 전송을 할 수 있을 것이다.

1. 팀원 역할

팀장의 역할은 류원석 학생이 맡고, 강수민, 심준우 그리고 정우주 학생은 팀장을 도와 개발 및 연구를 진행한다. 본 팀은 2번 Parallel TCP와 3번 Reliable UDP 구현을 모두하고 속도를 비교하여 우수한 구조를 채택할 예정이며, 구조는 구현가능 여부나, 테스트를 통해 바뀔 수 있다.