

네트워크컴퓨팅 2주차 과제 해답

[문제1] 혼잡제어에 대해 기술하라.

* 혼잡제어(congestion control)

- 인터넷의 전송 용량 한계로 인해 혼잡이 발생할 시 송신 측의 전송 속도를 낮춰 혼잡을 피한다.
- TCP에서는 혼잡제어를 하기 위해 혼잡윈도우를 사용
- 송신 측은 흐름제어 윈도우와 혼잡 윈도우 중 작은 값을 최종 윈도우 크기로 정한다.
- 정해진 윈도우 크기 범위 내에서만 송신이 가능
- 송신 측 TCP는 수신 측으로부터 정상적으로 Ack가 오면 네트워크에 여유가 있다고 판단하여 혼잡윈도우를 1씩 증가
- 네트워크에 혼잡이 발생하면 혼잡윈도우 값을 급격히 1로 줄여 전송량을 조정
- 정상적으로 송신이 이루어질 때 Ack를 받을 때마다 다시 증가시키는데 혼잡윈도우 값이 1로 되기 직전의 혼잡윈도우 값의 $1/2$ 이 될 때까지 2배씩 증가

[문제2] 통신 프로토콜이 가져야할 특징들과 그 의미를 기술하라.

* 효율적

- 주어진 통신 채널(매체, 전송속도 등)을 최대한 이용할 수 있어야 한다.
- 프로토콜의 처리순서 정의, 패킷의 최적 크기 설정, 흐름제어를 사용

* 안정적

- 비정상적인 장애(고의로 잘못된 데이터 송신, 통신망 장애 발생, 전송중 오류 발생) 발생 시에도 안정되게 동작해야 한다
- 오류제어가 필요

* 표준화

- 널리 사용되기 위해서는 미리 표준화되어야 한다
- 인터넷 관련 표준은 IETF(Internet Engineering Task Force)에서 RFC로 제공

[문제3] OSI 7계층 구조와 각 계층의 기능을 기술하라.

* 강의자료 참조

[문제4] TCP/IP 4계층 구조와 각 계층의 기능을 기술하라.

* 강의자료 참조

[문제5] 통신 프로토콜 정의 방식 중 필드 정의 방식에 대해 기술하라

* 프로토콜 데이터 단위(PDU)의 헤더나 트레일러 필드 이용

* 서비스 데이터 단위(SDU)

- 통신 프로토콜 계층에서 상위 계층으로 전달할 데이터 부분
- 사용자 데이터 또는 유료부하(payload)라고 함

* PDU는 SDU에 헤더와 트레일러가 추가된 것

헤더	서비스 데이터 단위	트레일러
----	------------	------

* 필드 정의에 의한 프로토콜 표현 방식

- 헤더나 트레일러의 지정된 필드에 프로토콜 처리 내용을 기록하는 방식
 - . 주소 필드, 순서번호, 에러처리 코드 필드 등
- 헤더나 트레일러를 크게 정의할 경우
 - . 다양한 프로토콜을 쉽게 표현
 - . 오버헤드가 증가하여 전송 효율 저하

[문제6] 응용계층프로그래밍의 특징과 장단점을 기술하라.

- * 특징
 - 네트워크 응용 패키지나 TCP/IP 응용 프로토콜을 이용
 - 데이터의 송수신을 구체적으로 다루지 않고 응용작업을 네트워크 환경에서 실행
- * 장점
 - 복잡한 네트워크 서비스를 편리하게 구현할 수 있음
- * 단점
 - 하위 계층의 구체적인 동작(연결설정, 데이터그램 송수신, 흐름제어)을 직접 제어할 수 없음
 - 아래 계층 프로그래밍에 비해 통신 효율이 떨어질 수 있음

[문제7] 2-tier 클라이언트-서버 모델 구조를 설명하고 장단점을 기술하라.

- * 클라이언트가 서버로 서비스를 요청하고 서버는 이에 응답
- * 대부분의 통신 프로그램 : 웹(http), ftp, telnet, mail
- * 장점
 - 구조가 간단하여 프로그램이 용이
- * 단점
 - 서버에서의 병목 현상
 - 클라이언트는 서버의 정보(IP주소, 포트번호)를 알아야한다.
 - 클라이언트의 증가는 서버에 트래픽 집중과 처리 용량 부족 현상 발생

[문제8] 분산객체 모델의 구조와 장점을 기술하라.

- * TCP/IP와 같은 통신 서비스를 이용한 지능화된 네트워크 서비스 제공
- * CORBA, Java RMI, .NET, SOAP등이 미들웨어 환경을 제공
- * 클라이언트, 미들웨어, 서버의 관계
 - 클라이언트 : 미들웨어를 통해서 서비스 요청을 서버로 전송
 - 서버 : 미들웨어를 통해서 클라이언트로 서비스 결과를 전송
- * 서버와 클라이언트의 역할이 미리 지정되지 않고 모두 객체로 구현
 - 구현 객체를 갖고 있으면 서버로 동작하면서 서비스를 제공
 - 클라이언트는 인터페이스만 알면 분산객체 서비스를 받을 수 있음
- * 클라이언트와 서버 프로그램이 독립적으로 구현 및 업그레이드 가능