**Data Communications Take-Home Exam**

**Dec. 15th, 2020**

**WARNING**: These are mostly “open-ended” questions that have no fixed answers. If your approach or the solution is partially or totally similar to other students’, it will be considered cheating, and all involved parties will be given “F” grade for the course. Likewise for copying the resources on the Internet.

***경고: 아래 문제들은 대부분 정해진 답이 없는 “열린” 문제들입니다. 만일 풀이 과정과 답이 일부 혹은 전부 다른 학생(들)의 답안과 비슷한 경우 부정행위가 있었던 것으로 간주하고 모든 관련자를 F 학점 처리합니다. 인터넷에 있는 자료를 그대로 복사한 경우에도 같은 조치가 취해집니다.***

[**Checking criteria**: (1) application of the course materials discussed after midterm, (2) concreteness, (3) originality, (4) [no] jumps in the protocol procedure, (5) possibility of performance calculation of the designed scheme, (6) meeting the design requirements]

[**채점기준**: (1) 중간고사 이후 학습한 내용의 활용도, (2) 구체성, (3) 독창성, (4) 프로토콜 작동상 절차적 허점 유무, (5) 설계된 시스템의 성능 계산 가능성 (6) 설계 요구조건 만족도]

**Student ID/학번: 2018380603**

**Name/이름: 정이든**

**Submit to BLACKBAORD \*\*\*Assignments \*\*\* (\*\*\*과제 및 시험 \*\*\*).**

**Submission deadline is 23:59 PM tonight – All late submissions will be given 0 points.**

**블랙보드 \*\*\*과제 및 시험\*\*\* 에 제출.**

**제출기한 오늘 밤 23:59 PM – 늦게 제출시 0점 처리**

**채점자 사용 공간/Grader Only**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **Total** |
|  |  |  |  |  |  |

1. In order to help detect close contacts that enable Covid-19 infections, you have decided to implement a communication-based smartphone app that records such contacts. The distance between two smartphones that defines the close contact is 1 meter, and the infection between the smartphone holders is likely in the close contacts that last over 5 minutes. You do not want to miss any true close contact events, or record any “false” contacts that do not satisfy the distance requirement. Design a wireless protocol that will be used by the app. Among others, elaborate on the three DLC functionalities in your protocol.

당신은 밀접 접촉으로 인한 코로나 감염을 탐지하는데 도움을 주기 위해, 통신을 기반으로 하여 그러한 접촉들을 기록하는 스마트폰 앱을 개발하기로 하였다. 밀접 접촉이란 1 m 거리 이내의 접촉을 말하며, 감염은 밀접 접촉이 5분 이상 지속되면 일어날 가능성이 높다. 당신은 설계한 시스템이 밀접 접촉을 하나라도 놓치기를 원치 않으나, 또한 거리 기준을 맞추지 못하는 “가짜” 밀접 접촉을 기록하고 싶지도 않다. 이 앱이 사용할 무선 통신 프로토콜을 설계하라. 무엇보다도 당신의 프로토콜의 DLC 기능 3가지가 잘 드러나도록 자세히 설명하라.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| --- Insert your answer here. Max 2 pages. You can use appropriate figures. ---  ------- 여기에 답안 삽입. 최대 2 페이지 이내. 적절한 그림들 사용 가능. -------  **먼저 스마트폰 앱의 이름은 “코로나존”이며 감염자와 비감염자는 항상 앱에 접속하고 있다고 가정하겠다.**  **“코로나존”은 사용자가 앱에 접속한 상태라면 1초에 한번씩 사용자의 위치값을 3개의 위성을 통해 얻어낸 후,**  **그 위치 데이터를 신호로 바꾸어,**  **가장 근처에 있는 무선 기지국으로 보낸다.**  **그리고 이는 고속 유선망을 통해 서버로 보내진다.**  **서버 개발자들은 얻어낸 감염자의 위치값과 비감염자들의 위치 값을 실시간으로 확인하여 그 사이 거리가 10m 이내면 “밀접 접촉”이라고 인식하여 비감염자의 휴대폰에 “밀접 접촉!”이라는 데이터를 보내도록 한다.**  **여기까지가 앱의 기능이다.**  **다음은 DLC 기능이다.**  **먼저 우리는 밀접 접촉을 놓치거나, “가짜” 밀접 접촉을 기록하고 싶지 않기 때문에 사용자의 휴대폰에서 기지국으로 보낸 패킷 payload에 담겨져 있는 위치 정보가**  **기지국에 도착하였을 때의 패킷과 다르지 않음을 보장해야한다. 그렇기 위해서는 data-link layer에서 error, flow control을 보장해야하고 그것의 프로토콜로서 “Stop-and-Wait”을 사용한다. 물론 당연히 단위화를 해야하므로 framing도 한다.**  **Frame에는 반드시 사용자의 위치 정보를 담아서 보내야하고, Error detection을 위해 훌륭한 에러 감지율을 자랑하는 CRC를 사용한다.**  **그리고 Receiver로부터 ACK를 받아야 에러 없이 정확한 데이터가 서버로 전송된 것이므로 Sender는 ACK를 기다린 후 정상적으로 들어왔으면 다음 패킷을 보내도록 flow control을 구성한다.**  **그리고 timer의 expire time을 10ms로 설정하여, ACK를 못받았을 시 빠르게 재전송함으로써, 앱이 1초 마다 사용자의 새로운 위치를 패킷에 담아 보내기 전에, 전송에 실패한 데이터를 재전송할 수 있도록 함으로써 유저에게 감염 소식을 빠르게 알린다.** | | | | | | |
| A | C | O | L | P | R | total |
|  |  |  |  |  |  |  |

1. Suppose a communication channel induces the following type of errors. Once an error occurs at a bit position, the next bit will also suffer an error with probability . On the other hand, if a bit does not have an error, the next bit is also clean with probability . Design your CRC tailored for a communication mechanism using this channel. Mathematically analyze your proposed CRC mechanism in terms of its error detecting capability.

어떤 통신 채널은 다음과 같은 에러 특성이 있다. 한 비트 위치에서 에러가 나면, 다음 비트는 확률 0.9로 에러가 난다. 반면에, 한 비트가 오류가 없다면 다음 비트도 확률 0.9로 에러가 없다. 이런 채널에서 통신해야 하는 기술을 위해 특화된 CRC를 설계하라. 당신이 설계한 CRC의 예상되는 에러 발견 능력을 수학적으로 분석하라.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| --- Insert your answer here. Max 2 pages. You can use appropriate figures. ---  ------- 여기에 답안 삽입. 최대 2 페이지 이내. 적절한 그림들 사용 가능. -------  **이 채널은 Codeword의 MSB가 bit flip이 됐느냐 안됐느냐가 전체 Error를 결정한다. 예를 들어 4bit인 Codeword가 채널을 통과하면서 첫 번째 비트인 MSB가 flip이 되었다면, 나머지 3bit는 0.9^3 확률로 flip될 것이다.**  **=> E(x) = x^3 + x^2 + x^1 + 1 (1)**  **반대로 채널 통과시 첫 번째 비트에 에러가 안끼어서 bit flip이 되지 않았다면, 나머지 3bit도 0.9^3 확률로 flip되지 않을 것이다.**   * **E(x) = 0 (2)**   **\* Received codeword = C(x) + E(x)**  **그렇다면 확률상 이 채널을 통과하여 Receiver에게 전달될 Codeword의 E(x)는 매우 높은 확률로 위의 두 경우의 수로 떨어진다.**  **(2)번째 경우는 e(x) = 0. 즉, 에러가 없으므로 고려할 필요가 없다. 우리는 (1)번째 경우에서 e(x)를 g(x)로 나눌 때의 나머지가 0인 상황만 피해서 CRC를 설계하면 된다.**  **위의 경우의 수는 아래의 두가지 경우의 수로 또 다시 나뉜다. 각각의 경우에 e(x) / g(x) != 0 을 항상 만족시키는 g(x)는 많지만 나는 g(x)가 10이라고 가정하고 풀어보겠다.**   1. **Codeword의 비트수가 짝수일 때 (4bit라 가정)** 2. **Codeword의 비트수가 홀수일 때 (3bit라 가정)**   **두 경우의 수 모두 g(x)인 10으로 나누어도 Syndrome이 non-zero이므로 0.9^n 확률로 생겨나는 1가지 Error에 대하여 정확하게 error detection을 할 수 있다.** | | | | | | |
| A | C | O | L | P | R | total |
|  |  |  |  |  |  |  |

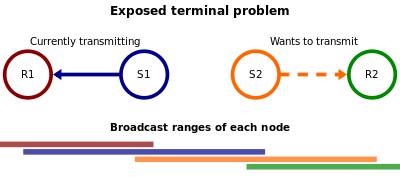
1. For Bluetooth, you are designing a protocol to maximize the throughput when there are many contending Bluetooth devices. (Imagine you are the chief designer that your algorithm will be used by all Bluetooth devices.) Now, you want to increase or decrease the number of packets to transmit per unit time , as a function of the number of contending devices , which you cannot directly observe. Fortunately, you can observe the average number of packet transmission attempts. Design the algorithm and mathematically analyze its performance. Assume that is equal to 10 packet transmission times.

당신은 여러 대의 경쟁하는 Bluetooth 기기가 있을 때, 그 throughput을 최대화하는 프로토콜을 설계하고 있다. (당신은 최고 설계자이기 때문에 모든 Bluetooth 기기는 당신이 설계한 알고리듬을 사용하게 된다고 가정하자.) 구체적으로 당신은 경쟁하는 기기의 수 N의 함수로 단위시간 T 당 전송 패킷 수를 증가하거나 감소하는 알고리듬을 설계해야 한다. 문제는 N이 직접 관찰할 수 없는 값이라는 것이다. 다만, 단위 시간당 패킷 전송 시도가 평균 몇 번 있었는지는 관찰할 수 있다. 알고리듬을 설계하고 수학적으로 분석하라. T를 10 \* 패킷 전송시간으로 가정하라.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| --- Insert your answer here. Max 2 pages. You can use appropriate figures. ---  ------- 여기에 답안 삽입. 최대 2 페이지 이내. 적절한 그림들 사용 가능. ------- | | | | | | |
| A | C | O | L | P | R | total |
|  |  |  |  |  |  |  |

1. We have discussed the hidden terminal problem, and how it can be solved through the use of RTS/CTS. But then, there is also a problem called the exposed node problem. As shown in the following Wikipedia illustration, S1 and S2 cannot transmit simultaneously because they worry that their transmissions collide. The problem is that even if they transmit simultaneously, the colliding transmission will have no problem reaching their respective destinations R1 and R2. It is because R1 is beyond the communication range of S2, and R2 is beyond the communication range of S1. Now, design a heuristic that solves the exposed node problem. Make sure that your solution does not worsen the hidden terminal problem. [Hint: RTS/CTS may come to the rescue again. In particular, CTS can be sent to the transmitter itself.]

우리는 hidden terminal problem이 무엇인지와, 그 문제가 어떻게 RTS/CTS를 통해 해결될 수 있는지를 학습했다. 그런데 exposed node problem이라는 것도 있다. 위의 Wikipedia 설명에서처럼, S1과 S2는 패킷 충돌을 우려하여 동시에 전송하지 않는다. 하지만 동시에 전송한다고 해도, 사실 문제는 없다. 왜냐 하면 R1은 S2의 전송 범위 밖에 있으며 R2는 S1의 전송 범위 바깥이기 때문이다. Exposed node problem을 해결할 방법을 설계하라. 단, 그 방법이 hidden terminal problem을 악화시켜서는 안 된다. [힌트: RTS/CTS가 도움이 될 수도 있다. 특히 CTS는 자기 자신에게 보낼 수도 있다.]



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| --- Insert your answer here. Max 2 pages. You can use appropriate figures. ---  ------- 여기에 답안 삽입. 최대 2 페이지 이내. 적절한 그림들 사용 가능. -------  **현재 위 그림에서의 상황은 S1이 R1에게 패킷을 전송하고, S2는 R2에게 패킷을 전송하면 아무 문제가 없는 것을 기대한다.**  **하지만, 실제로는 S1과 S2는 Carrier sensing을 지속적으로 하고 있기 때문에 본인에게 패킷을 보내는 것이 아님에도 불구하고 채널이 busy하다고 판단하여서 데이터를 전송하지 않고있는 비효율적인 상황이다.**  **Hidden terminal problem을 약화시키지 않고 이러한 상황을 해결하기 위해서 다음과 같은 가정이 필요하다.**  **먼저 첫번째 가정은 RTS(request to send)를 내가 보내고자 하는 station 뿐만 아니라 보낼 수 있는 모든 방향으로 보내는 것이다.**  **두번째는 그 RTS 패킷의 헤더에 내가 보내고자 하는 노드의 destination address를 적는것이다.**  **이 두가지 가정 외에는 변경된 것이 없다. (CTS도 똑같이 진행한다.)**  **위와 같은 가정에 의해서, 노드 S1에서 RTS 패킷을 양쪽으로 보냈을 때 R1과 S2에 동시에 도착한다고 하더라도(최악의 상황) S2는 그 RTS 패킷의 헤더를 보고 ‘아 나에게 데이터를 보내는 것이 아니구나’라고 생각하고 R2와 통신하면 된다.**  **위의 예시는 S1을 기준으로 생각했는데 S2를 기준으로 생각해도 동일하다.**  **이렇게 채널이 busy 상태여서 collision이 나더라도**  **RTS 패킷을 확인하고 나에게 보낼 데이터인지 아닌지를 판단하는 processing을 추가하면**  **위와 같은 문제를 해결할 수 있다.** | | | | | | |
| A | C | O | L | P | R | total |
|  |  |  |  |  |  |  |

1. It is early 20th century. There is a large city with population where the citizens are very willing to subscribe to the telephone service if the telephone has no call blocking possibility. As a telephone engineer, you are given crosspoints () by the company to design a switching hardware for the city. You could make either a single huge crossbar switch or a multi-stage switch. What is the maximum number of telephone subscribers that you can get by making a wise design decision?

때는 20세기 초. 어떤 큰 도시의 시민들은 call blocking 가능성이 없다면 전화 서비스에 가입할 의사가 있다. 전화 엔지니어인 당신은 회사로부터 X개의 crosspoints를 (X << P) 받아서 이 도시를 위한 스위치 하드웨어를 제작하려고 한다. 당신은 한 개의 큰 crossbar 스위치를 만들 수도 있고, multi-stage switch를 만들 수도 있다. 당신이 현명한 설계 결정을 내림으로써 최대로 확보할 수 있는 가입자의 수는 몇인가?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| --- Insert your answer here. Max 2 pages. You can use appropriate figures. ---  ------- 여기에 답안 삽입. 최대 2 페이지 이내. 적절한 그림들 사용 가능. -------  **도시의 전체 인구가 P명일 때 non-blocking 조건을 만족해야한다. 만약 crossbar switch를 사용할 때 위 조건을 만족시키기 위해서는 P \* P개의 crosspoints가 필요하지만 실제로 한번에 사용되는 것은 그것의 25%도 되지 않으므로 매우 비효율적이다.**  **그렇다고 multistage switch를 그냥 사용하자니 같은 output으로 나가지 않더라도 길이 모자라서 blocking이 되는 경우가 발생하여 위 상황에 적합하지 않다.**  **그러므로 Clos criteria를 사용하여 Non-blocking한 condition을 만족하면서, 전화 서비스를 사용하는 인구 P의 최대값을 구해보자.** | | | | | | |
| A | C | O | L | P | R | total |
|  |  |  |  |  |  |  |