



## HW3 Report

과 목 명	정보통신공학
담당 교수	이형준
학 과	컴퓨터공학과
학 번	2071035
이 름	이소민 (LeeSomin)
제 출 일	2023.05.22


## Part I.

## HW#3

34743-02 정보통신공학

2071035 이소민

## [Part I]

1.  4 call for 8hr workday  
 ↳ 6min each in avg  
 ↳ 10% long distance

max # of telephones an end office can support

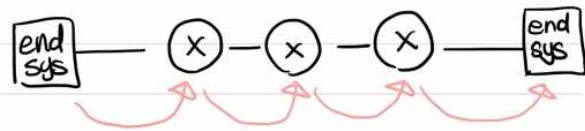
$$\text{circuit switching} \rightarrow \frac{1 \cdot 10^6 (\text{Hz})}{4 \cdot 10^3 (\text{Hz})} = \frac{1000}{4} = 250 (\text{circuits})$$

1 hr  $\rightarrow$  0.5 call, 6min each  $\rightarrow$  3min per hr60min  $\rightarrow$  3min  $\times$  20 phones can use circuit (long dist)long dist  $\rightarrow$  (10%) 20 phones, tot call  $\rightarrow$  200 phones per circuit

$$250 (\text{circuit}) \times 200 (\text{phones per circuit}) = 50000$$

A. 50000 telephones

2.  $N = 4$  (# of hops)

 $L = 3200$  bit (message length) $B = 9600$  bps (data rate) $P = 1024$  bit (fixed packet size) $H = 16$  bit (overhead per packet) $S = 0.2$  sec (call setup time) $D = 0.001$  sec (propagation delay per hop)

\* datagram packet switching

$$\# \text{ of packet} : \frac{L}{P} = 3200 \div 1024 = 3.125$$

$$\text{total packet length} : P + H = 1024 + 16 = 1040 (\text{bits})$$

$$\text{total data size} : 1040 \times 3.125 = 3250 (\text{bits})$$

$$\text{transmission delay} : \frac{3250 (\text{bits})}{9600 (\text{bps})} = 0.33854 \dots$$

$$\text{end to end delay} : (\text{transmission delay}) + (\text{propagation delay} \times \# \text{ of link (hop)}) = 0.3385 \dots + 0.004 \approx 343 (\text{ms})$$

\* virtual circuit packet switching

$$\text{end to end delay} : \left( \text{datagram end-end delay} \right) + \left( \text{call setup time} \right) = 343 (\text{ms}) + 200 (\text{ms}) = 543 (\text{ms})$$

\* Circuit switching

$$\text{transmission delay} : L/B = \frac{3200(\text{bits})}{9600(\text{bps})} = \frac{1}{3}(\text{sec})$$

$$\text{end to end delay} : \left( \text{transmission delay} \right) + \left( \text{call setup time} \right) + \left( \text{propagation delay} \right)$$

$$= 0.333... + 0.2 + 4 \cdot 0.001 \approx 0.537(\text{sec}) = 537(\text{ms})$$

A. Circuit switching: 537ms

datagram packet switching: 343ms

virtual circuit packet switching: 543ms

3.



AP/IP: 802.11b

Max data rate 11Mbps

a) 두 ISP는 각각의 MAC 주소와 SSID를 가지므로 addressing을 통해 각각 독립적 작업이 가능하다. 하지만 두 ISP가 같은 channel을 이용한다면 collision의 가능성이 있다. 따라서 두 ISP가 동시에 같은 channel을 사용할 때의 maximum aggregate transmission rate은 11Mbps이다.

A. 11Mbps

b) 두 ISP가 각각 다른 channel을 이용한다면 collision의 가능성 없이 각각 max data rate인 11Mbps를 완전히 maximum transmission rate로 가진다. 따라서 두 ISP의 maximum aggregate transmission rate은 22Mbps이다.

A. 22Mbps

4. RTS/CTS frame: 2 byte + 4 byte(CRC) = 6 byte = 48 bit

Max data rate: 11Mbps =  $11 \cdot 10^3$  bps

Data length: 1000 bytes =  $2^3 \cdot 10^3$  bits

802.11 frame: 2+2+6+6+6+2+6+payload+4 = 34+payload(bytes)

RTS, CTS, ACK 프레임 전송 시간:  $48 / (11 \cdot 10^3) = 4.36 \mu s$

전체 프레임 전송 시간:  $(1000 + 34) \times 8 / (11 \cdot 10^3) = 752 \mu s$

전송 sequence: wait DIFS  $\rightarrow$  RTS  $\rightarrow$  wait SIFS  $\rightarrow$  CTS  $\rightarrow$  wait SIFS  $\rightarrow$  Data frame  
 $\rightarrow$  wait SIFS  $\rightarrow$  ACK

$\therefore$  총 전송 시간:  $DIFS + 3SIFS + (3 \cdot 4.36 + 752) \mu s$

A.  $DIFS + 3SIFS + 766.08 (\mu s)$

>> part II 다음 쪽에 이어짐

**Part II.****1. Code Analysis**

\*이 코드는 flexible generator CRC Shift Register를 구현하고 있다.

1) `crc_shift_register(data, generator)`

input: data, generator  
return: crcstr, codewordstr

이 함수는 string 형태의 data와 generator를 받아 split함수로 쪼개어 정수의 배열로 각각 변환한다. 함수 내부에서 사용되는 변수와 배열은 CRC 레지스터의 크기인 `crclen`과 `crc register`를 의미하는 정수형 배열 `crc`, 그리고 데이터와 CRC 비트를 이어붙인 `codeword`의 배열 `codeword`가 있다. `codeword` 배열에 data를 복사한 후 data를 `crc`의 길이만큼 왼쪽으로 shift해준다.

CRC shift register를 구현하기 위해 원형 배열을 사용한다. 정수형 변수 `idx`는 `crc` 배열의 첫 원소의 인덱스를 가리키고 `idx`를 1 증가시키는것은 각 비트를 왼쪽으로 하나씩 shift시키는 것과 같다. `generator`의 MSB가 1이면 `crc`의 MSB가 XOR연산의 항으로 사용된다. `out` 변수는 CRC레지스터의 XOR 연산에서 한쪽 항으로 들어가게 되는 값이다. `generator`의 LSB가 1이면 `out`과 다음 data bit를 XOR하고, 0이면 연산없이 data bit를 가져와 CRC의 가장 왼쪽에 저장한다. 나머지 CRC 레지스터의 값들은 대응되는 `generator`의 값이 1인 경우 `out`의 값과 XOR연산이 수행된다. 이 단계의 모든 연산이 끝나면 `idx`의 값이 1 증가하여 CRC의 가장 왼쪽 원소가 가장 오른쪽으로 가고 레지스터 내부의 값이 왼쪽으로 한 칸씩 shift 된다.

data의 모든 bit가 `crc` 연산을 완료하면 함수는 `codeword`에 계산된 CRC bit를 이어붙이고 `make_string()`을 불러 `crc`와 `codeword`를 string형으로 변환하여 리턴한다.

2) `print_crc(idx, crc)`

input: idx, crc  
return: crcprt

이 함수는 원형 배열 `crc`와 그 시작인덱스 `idx`를 받아 순서대로 새 배열 `crcprt`에 복사하여 원형배열을 일반 배열로 변환하여 리턴한다.

3) `make_string(arr)`

input: arr  
return: temp

이 함수는 입력받은 배열을 for문을 통해 구분자가 공백문자인 string형태로 변환하여 리턴한다.

4) `crc_check(codeword, generator)`

input: codeword, generator  
return: -

이 함수는 data 대신 `codeword`를 받아 `crc_shift_register`가 `crc` bits를 생성하는것과 같은 작업을 수행한다. 다만 `crc`가 `codeword`에서 더 이상 읽어들이 bit가 없을 때 `crc`에 남은 bit가 모두 0이 되지 않으면 오류메시지를 출력하고, `crc`에 남은 bit가 모두 0이면 오류가 없음을 출력한 뒤 리턴한다.

5) `hw3_part2()`

CRC shift register를 실행하기 위한 함수로 최초입력을 1로 하면 TX의 기능을, 2로 하면 RX의 기능을 수행하며 2, 3번째 입력은 공백문자로 구분된 data/codeword와 generator로 받는다.



## 2. Code Result (flexible generator 포함)

```

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
CRC bits calculated by CCITT-16: 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0
The complete codeword: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 0 0 1 0 0 0 0 0 0 1 0
0 0 0 1 0
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
An error is not detected (according to CCITT-16)!
Done...
>>>

```

```

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0 1
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
CRC bits calculated by CCITT-16: 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1
The complete codeword: 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 1 0 1 0 1 0 0 0 0 1
0 1 0 0 1 0 1
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
An error is not detected (according to CCITT-16)!
Done...
>>>

```

```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part 11] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0 1 1
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
CRC bits calculated by CCITT-16: 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1
The complete codeword: 1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 1
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
>>> [HW #3 Part 11] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 1 1 1 0 1 1 0 0 0 1 0
1 1 0 1 0 1 1
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
An error is not detected (according to CCITT-16)!
Done...
>>>
```

```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part 11] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0 1 1 1
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
CRC bits calculated by CCITT-16: 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0
The complete codeword: 1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 1 0
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part 11] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 1 1 1 0 1 1 0 0 0 1 0
1 1 0 1 0 1 1 0
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
An error is not detected (according to CCITT-16)!
Done...
>>>
```



```

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0 1 1 1 0
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
CRC bits calculated by CCITT-16: 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 0
The complete codeword: 1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 0
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 1 1 1 0 1 1 0 0 0 1
0 1 1 0 1 0 1 1 0 0
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
An error is not detected (according to CCITT-16)!
Done...
>>>

```

\*flexible generator:

```

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0 1
Type generator bits: 1 1 0 0 1 1 0 0

[Result]
CRC bits calculated: 0 1 0 1 0 0 0
The complete codeword: 1 0 1 0 1 0 1 0 0 0
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 1 0 1 0 1 0 0 0
Type generator bits: 1 1 0 0 1 1 0 0

[Result]
An error is not detected!
Done...
>>>

```



```

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0
Type generator bits: 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 0
0 1 1

[Result]
CRC bits calculated: 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1
0 1
The complete codeword: 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0
1 0 1 0 1
Done...
>>>
===== RESTART: C:\Users\somin\Desktop\hw3part2.py =====
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1
Type generator bits: 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 0
1 1

[Result]
An error is not detected!
Done...
>>>

```

\* 오류 검출 sample

```

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\소민\Desktop\hw3part2.py =====
==
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 1
Type information bits that you want to send ex) 1 0 0 1 1 0 1: 1 0
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
CRC bits calculated by CCITT-16: 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0
The complete codeword: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0
Done...
>>>
===== RESTART: C:\Users\소민\Desktop\hw3part2.py =====
==
[HW #3 Part II] Student ID: 2071035 Name: Somin Lee
Select the mode between TX and RX (TX:1, RX:2): 2
Type the codeword that RX recieved: ex) x x x ... x x: 1 0 0 0 1 0 0 0 0 0 0 1 0
0 0 0 1 1
Type generator bits: 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1

[Result]
An error is detected (according to CCITT-16)!
Done...
>>>

```