

홈 헬스를 위한 메디컬 센서노드의 분산보관 프로토콜

Distributed Archiving Protocol between the Medical Sensor Nodes for the Home Health Service

이영호*, 장희태**, 이병문**
가천의과학대학교 정보공학부*, 유-헬스케어연구소**

Young-Ho Lee(lyh@gachon.ac.kr)*, Hee-Tae Jang(halfmoon1990@nate.com)**,
Byung-Mun Lee(bmlee@gachon.ac.kr)**

요약

메디컬 센서노드에서 측정된 생체정보는 홈 헬스 게이트웨이가 장애중이더라도 분실되지 말아야 서비스를 정상적으로 제공할 수 있다. 장애 중에 데이터를 수신할 수 없다면 메디컬 센서노드가 임시로 로컬저장소에 저장하거나 다른 센서노드와 상호 교신하여 저장 공간이 가장 많을 것으로 예상되는 센서노드를 선택한 후 분산 보관한다면 생체정보의 분실을 최소화 할 수 있다. 뿐만 아니라 정보의 측정주기는 생체정보의 유형에 따라 다르기 때문에 분산 보관에 필요한 공간이 기기마다 다르므로 보다 효율적인 보관기법이 필요하다. 이에 본 연구에서는 각각의 다양한 측정주기를 갖는 생체정보 센서노드간의 효율적 분산보관 프로토콜(DAP)을 제안하였다. 또한 DAP의 효용성을 확인하기 위해 센서노드와 게이트웨이 간에 DAP을 설계하고 구현하였다. 구현한 프로그램을 이용하여 실험을 한 결과, 99.3%의 높은 회수율과 적중률을 구할 수 있어 일시적인 장애에는 센서노드가 그 역할을 할 수 있음을 확인할 수 있었다.

■ 중심어 : | 홈 헬스 | 생체정보 | 분산보관 프로토콜 | 센서네트워크 | 장애관리 |

Abstract

Medical data sampled through medical sensor nodes can provide services properly only when the data are not lost even during the fault of a home health gateway. The loss of medical data can be minimized if a sensor node, in which it is expected that there are the most saving spaces, is selected after medical sensor nodes tentatively conduct local save or communicate with each other during a fault when data cannot received. Furthermore, efficient saving techniques are necessary since the cycle for sampling information is different according to the type of medical data and a space for distributed saving is different for each apparatus. So, this research suggests an efficient distributed archiving protocol (DAP) for medical data sensor nodes, each of which has a diverse sampling cycle. In order to confirm the usefulness of DAP, DAP between sensor node and gateway was designed and materialized. An experiment was conducted using the materialized program and earned a high level of recovery rate (99.3%) and of accuracy rate, which confirms that sensor nodes can play their role during a temporary fault.

■ keyword : | Home Health | Medical Information | Distributed Archiving Protocol | Sensor Network | Fault Management |

* 본 연구는 2011년도 산학연 공동기술개발 사업(중소기업청)으로부터의 지원에 의한 연구결과임.

접수번호 : #111125-003

심사완료일 : 2011년 01월 04일

접수일자 : 2011년 11월 25일

교신저자 : 이병문, e-mail : bmlee@gachon.ac.kr

I. 서론

지그비나 블루투스과 같은 센서 네트워크를 이용한 홈 헬스 서비스는 메디컬 센서노드로부터 측정된 생체정보를 게이트웨이로 전송하고, 게이트웨이는 이것을 다시 홈 헬스 서버로 전송하여 건강관리를 제공한다[1]. 홈 헬스용 게이트웨이는 측정된 개인의 생체정보를 의료기관으로 전송하는 매우 중요한 역할을 하므로 생체정보를 안전하게 전송하는 것이 중요하다[2]. 그러나 이 과정에서 발생할 수 있는 여러 가지 문제 중 하나는 순간적인 트래픽증가로 게이트웨이가 정상적으로 동작하지 않거나 실수로 전원이 꺼져 일시적으로 동작이 정상적으로 되지 못하는 장애가 발생하는 경우이다[3][4].

게이트웨이의 장애로 메디컬 센서노드로부터 전송되는 생체정보를 정상적으로 수신할 수 없어 생체정보가 유실될 수 있다. 따라서 게이트웨이가 역할을 수행할 수 없는 문제를 해결하기 위한 방법으로 게이트웨이의 역할을 다른 메디컬 센서노드가 대신하도록 하는 것이다. 그러나 이것이 가능하려면 메디컬 센서노드가 게이트웨이의 상태를 주기적으로 감시하여야 하며 장애가 감지될 때 다른 모든 노드에 게이트웨이에 장애가 발생했다는 사실을 알려야 한다. 또한 게이트웨이가 정상적으로 복구될 때까지 측정된 생체정보를 각 메디컬 센서노드는 자체적으로 보관하여야 한다. 그러나 센서노드는 제한된 저장 공간과 배터리전력의 소모로 자체노드에 데이터를 다량으로 저장하기 어렵다[5][6]. 특히 심전도(ECG)와 같은 생체정보는 20ms마다 16비트 정보를 저장하여야 하기 때문에 장시간 저장하기에는 공간이 부족하다.

따라서 본 연구에서는 이를 해결하기 위한 방법으로 메디컬 센서노드에 분산하여 보관하는 방법을 제안하고자 한다. 생체정보를 효율적으로 분산하여 저장하려면 생체정보 측정주기와 저장 공간의 잔량을 고려하여야 한다[7][8]. 심전도, 산소포화도, 혈압, 혈당, 체온과 같은 생체정보는 측정주기가 서로 다르다. 예를 들면 심전도는 20ms의 측정 주기를 갖고 산소포화도는 1초단위의 측정주기를 갖고, 혈압은 하루 2~3회 정도의 측정주기를 갖게 되므로 환자가 측정할 때마다 생성된

다. 메디컬 센서노드의 메모리 공간은 메디컬 센서 노드의 매우 제한적일뿐만 아니라 생체정보를 버퍼에 저장하는 속도에 따라 달라질 수 있다[9]. 따라서 게이트웨이의 장애 시 모든 메디컬 센서노드에게 동일한 저장 방법을 사용하지 않고, 각 노드마다 적합한 상황에 맞도록 선정하여 저장하는 방법이 필요하다[10]. 본 논문에서는 이를 위해 분산 보관 프로토콜(DAP, Distributed Archiving Protocol for Medical Data)을 제안하고자 한다. 또한 DAP의 효율성을 확인하기 위해 심전도, 산소포화도, 혈압/혈당의 3가지 유형의 메디컬 센서노드에 TinyOS2.1기반의 센서응용 프로그램을 설계하고 구현하여 이를 이용한 실험을 통해 유효성을 확인하고자 한다.

본 논문의 구성은 2장에서는 홈 헬스 센서네트워크에서 게이트웨이가 장애중일 때 데이터를 임시로 보관하는 방법에 대해서 살펴본다. 3장에서는 본 논문에서 제안하는 효율적 분산보관 기법을 기술하며 4장에서 제안한 모델과 프로토콜을 구현하여 실험과 평가를 하고 마지막 5장에서 결론을 맺는다.

II. 홈 헬스 생체정보의 저장방식

1. 홈 헬스와 게이트웨이 장애

2007년부터 노인과 저소득층을 포함하는 사회 취약 계층에 대하여 실시하고 있는 홈 헬스 서비스는 보건소에서 일부 시범적 사업을 시행하였다. 그 시범사업의 결과를 살펴보면 대상자의 건강지표는 두드러지게 개선되고 있었으나 대부분의 서비스 대상이 고혈압이나 당뇨와 같은 예후가 있는 161명의 65세 이상의 1인 독거노인이었으므로 일반주민에 비해 IT접근도가 낮아 홈 헬스 관련기기에 대한 사용이 원활하지 않다는 문제가 드러났다. 이 중 대부분의 문제는 노인들이 게이트웨이의 전원을 끄거나, 장애가 발생해도 즉시 처리되지 못하는 상황에 대한 문제였다[11].

[그림 1]에서 보듯이 이러한 문제는 게이트웨이가 생체정보의 전송통로로 매우 중요한 역할을 하기 때문에 장애가 발생되면 생체정보를 분실해서 환자의 건강상

태를 정확히 판단할 수 없다[12].

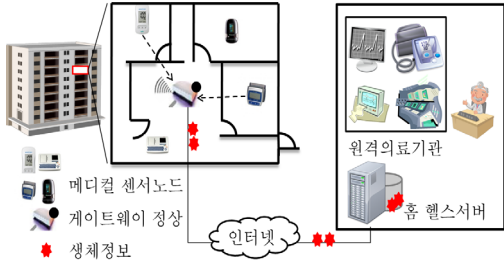


그림 1. 홈 헬스 시스템의 구성

따라서 게이트웨이에서 장애가 발생하였다면 측정된 생체정보를 메디컬 센서노드가 전송할 경우 분실될 수 있어 이를 처리할 장애처리 방법이 필요하다[13][14]. 이러한 분실 문제를 해결하기 위해 측정된 데이터를 저장하는 방법에 대해 먼저 고찰할 필요가 있다. 다음 절에서 생체정보의 특성에 따른 센서노드의 데이터 저장 방법에 대해 알아보자.

2. 생체정보의 특성과 저장방법

생체정보에는 심전도, 산소포화도, 혈압, 혈당, 체온 등 여러 종류가 있다. 생체정보는 종류에 따라 측정주기와 측정량이 다르다. 심전도의 측정주기는 20ms이며, 산소포화도는 1초의 측정주기를 가진다. 이와 달리 주기가 비교적 느린 혈압, 혈당, 체온과 같은 생체정보는 보통 하루에 2번, 혈당은 당뇨환자의 경우 3~4번의 혈당을 측정한다. 혈압은 보통 3번 정도 측정하나, 고혈압 환자는 24시간 내내 혈압을 측정하는 것이 바람직하다

일반적으로 센서 네트워크에서는 각 노드들이 측정된 데이터를 3가지 방법으로 저장한다. 즉, 외부의 저장 공간에 저장하는 방법, 노드 내부에 저장하는 방법, 다른 노드에 저장하는 방법으로 분류한다. 이 방법들은 저장할 데이터의 측정량과 주기에 따라 서로 장단점을 갖는다. 우선 홈 헬스에서는 사용하는 생체정보를 기준으로 살펴보자. 첫 번째의 경우는 센서에서 측정된 데이터를 지정된 외부장치로 보내어 저장하는 방법이다 [6][12].

[그림 2]을 보면 센서노드 N_1 에서 발생한 생체정보는

외부 서버의 저장공간에 저장하는 방법이다. 다시 말하면 센서노드 N_1 이 측정한 첫 번째 생체정보 D_1 은 서버에 N_1-D_1 형태로 저장되며, 센서노드 N_2 에서 측정된 첫 번째 생체정보 D_1 은 외부서버에 N_2-D_1 형태로 저장된다. 이와 같은 방식으로 외부 서버의 저장 공간에 연속적으로 저장된다.

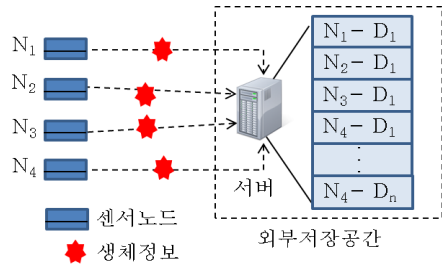


그림 2. 데이터의 외부저장

이 방법은 센서노드 자체에는 저장 공간이 필요 없지만 심전도와 같이 짧은 주기의 생체정보가 다량으로 외부 저장 공간에 저장돼야 할 경우 많은 트래픽을 발생시킬 수 있다. 둘째, 특정 센서노드에서 발생한 데이터를 센서노드 내부에 저장하는 방법이다[8][11]. [그림 3]를 보면 센서노드 N_1 에서 발생한 데이터 D_1 은 바로 센서노드 N_1 의 내부저장 공간에 저장되어 있다(N_1-D_1). 이 기법은 외부 저장 기법에서 발생하는 트래픽을 감소시킬 수 있다.

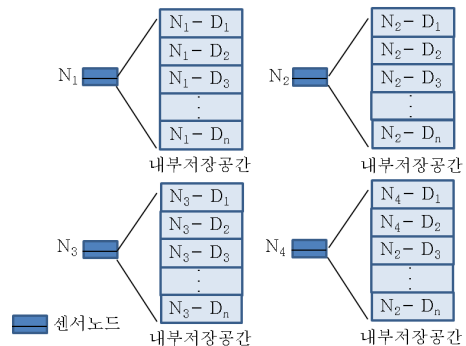


그림 3. 데이터의 내부저장

그러나 심전도와 같이 측정주기가 빨라서 저장할 데

이터가 많아질 경우에는 센서노드의 저장 공간이 빠르게 소모되기 때문에 오버플로우가 발생할 가능성이 높다. 또한 혈압과 같이 측정주기가 느리고 데이터양이 적은 경우에는 저장 공간이 상대적으로 충분히 남기 때문에 센서노드간의 상호 비효율적인 공간관리의 문제가 발생한다.

세 번째 방법으로 데이터 중심 저장(DCS, Data Centric Storage)기법이 있다[12-14]. 이 방법은 지리적인 네트워크 공간을 여러 개로 나누고 이 공간들을 식별하기 위한 식별자로 이름을 부여한다. 같은 이름을 갖는 모든 데이터는 같은 센서노드에 저장되는 방법이다[15]. 5개의 영역으로 구분한 [그림 4]의 예를 보면 센서노드 N_1 에서 생성된 데이터는 저장주소 이름이 "myhome"이므로 N_2 를 거쳐 N_5 로 이동한 후 데이터가 저장된다. 따라서 N_1 의 데이터는 센서노드 N_5 영역에 저장된다(N_1-D_5).

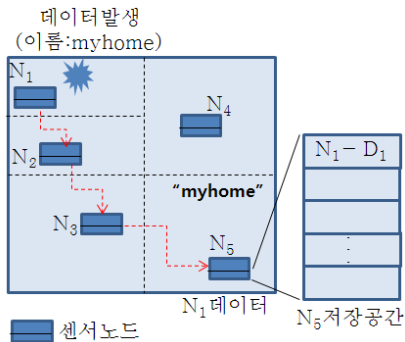


그림 4. 데이터의 중심저장

DCS방법은 데이터의 콘텍스트를 중심으로 저장하는 방식이므로 데이터의 저장가치를 높이는 방법이며, 센서노드가 이동을 하더라도 원래 저장되어야 할 곳에 저장되는 방식이다. 그러나 이 방법은 지리적으로 광범위한 영역환경에 적합하며, 데이터의 이름과 내용에 따른 별도의 주소지정 알고리즘이 필요하기 때문에 데이터 발생주기가 빠를수록 부하가 커져 홈 헬스 환경에 적용하기 어렵다. 또한 센서노드의 저장 공간을 동적으로 고려하여야 하므로 적용하기 어렵다. 따라서 본 연구에서는 생체정보마다 각각 다른 측정주기, 저장 공간을

고려하여 상황에 적합한 센서노드를 선정하고 측정된 정보를 분산하여 보관하는 분산 보관 모델(DAM, Distri-buted Archiving Model)을 제안한다.

III. 홈 헬스에서의 분산 보관 모델

1. 분산 보관 모델(DAM)

본 연구에서 제안하는 홈 헬스용 분산 보관 모델(DAM)은 게이트웨이가 장애를 일으켜서 정상적인 생체정보가 전송되지 못하는 경우 임시로 분산 보관하도록 하는 모델이다. 이 모델은 저장 공간의 효율적 관리와 불필요한 전송과정을 줄여야 하며, 장애 시에는 데이터 분실의 위험성을 줄일 수 있어야 한다. [그림 5]은 분산 보관 모델이 어떤 환경에서 적용될 수 있는지 보여준다.

[그림 5]에서처럼 게이트웨이에서 장애가 발생했다고 가정하자. 센서노드 N_4 는 게이트웨이가 정상적으로 수신할 수 없음(①)을 알게 되어 D_1 부터 D_5 까지의 5개의 데이터를 N_4 의 로컬 저장공간에 보관(②)한다. 로컬 저장공간의 여유공간이 부족하면 저장공간이 충분한 다른 센서노드인 N_3 에 D_6 과 D_7 을 보관(③)하도록 한다.

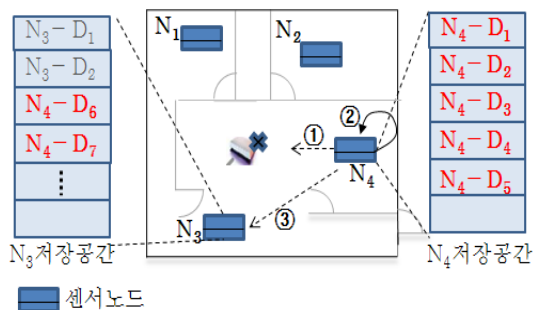


그림 5. 분산보관 모델

다음 절차는 이 모델에 대한 단계적 처리과정을 정의한다.

- Step 1) 게이트웨이로 전송할 생체정보를 로컬 센서노드 저장공간에 임시 보관한다.
- Step 2) 저장공간의 오버플로우가 예상되면 생체정

보 측정주기가 가장 긴 센서노드에게 Token 을 전송한다.

Step 3) Token을 받은 센서노드는 Token을 2번 이상 받았다면 Token을 폐기하고 그렇지 않다면 저장 공간의 여유 량을 확인하여 승인한다. Step 5로 이동한다.

Step 4) 만약 저장 공간이 부족하거나 승인할 수 없는 경우라면 [표 1]을 기준으로 생체정보 측정주기 시간이 가장 긴 센서노드에게 Token을 전송한다. 그리고 Step 3을 반복한다.

Step 5) 만약 승인이 가능하면 모든 이웃노드에게 자신이 Token을 보유하고 있으니 자신에게 생체정보를 보관하라고 송신한다.

Step 6) 차후 게이트웨이가 장애에서 복구되면 분산보관된 정보를 게이트웨이로 전송하고 저장 공간을 재설정한다.

표 1. 생체정보의 측정주기

메디컬 센서노드	생체정보 유형	측정주기 시간	data-rate (ms)
N ₁	심전도(ECG)	20ms	20
N ₂	산소포화도(SpO ₂)	1sec	1,000
N ₃	혈압(Blood Pressure)	3회/1일	28,800,000
N ₄	혈당(Blood Glucose)	2~3회/1일	28,800,000

이와 같은 처리절차는 [표 1]에 있는 생체정보의 유형에 기반을 둔다. 예를 들면, 측정주기가 20ms로 가장 빠른 심전도측정 센서노드 N₁가 게이트웨이의 장애를 탐지했다고 가정하자. N₁은 로컬 저장 공간에 데이터를 저장하게 된다. 빠르게 저장되기 때문에 여유 공간의 소진도 빠르다. 분산 보관할 다른 센서노드를 선정하는데 있어서 가장 많은 여유 공간이 예상되는 혈압 N₃이나 혈당 N₄이 유력하다. 따라서 N₃이나 N₄의 여유 공간의 최대 잔여시간(A)을 식(1)과 같이 구하여 선정하게 된다. Data_{rate}는 측정주기 시간을 말하며 Row_{available}은 최대 여유공간의 길이를 말한다.

$$A = Data_{rate} \times Row_{available} \quad (1)$$

예를 들면, 혈당센서는 여유 공간이 500개의 entry 만큼 있다고 가정하고, 혈압센서는 100개의 entry 만큼 있다고 가정한다면, 혈당센서는 28,800,000ms x 500 row = 14,400x10⁶ 이지만 혈압센서는 28,800,000ms x 100 row = 288x10⁷의 값으로써 실제 혈당센서노드의 잔여 시간이 훨씬 많다. 따라서 혈당센서는 혈압센서에 비해 상대적으로 남아 있는 여유시간이 많다. 그래서 노드선정알고리즘은 혈당센서를 선정하게 된다. Data_{rate}는 생체측정 유형에 따라 고정되지만 row는 상황에 따라 가변적이기 때문에 A의 결과도 가변적이다. 그래서 가변적인 상황을 고려하여 시간에 따라 A의 값도 변화될 것이고 이 부분을 Token으로 반영하게 되는 것이다. 이와 같이 게이트웨이에 장애가 발생하더라도 효율적인 분산보관을 통해 극복할 수 있다. 그렇다면 분산보관을 처리하기 위한 센서노드간의 분산보관 프로토콜(DAP)과 전송메시지에 대해 다음절에서 정의한다.

2. DAP 정의 및 설계

DAP은 2가지의 경우를 고려한 프로토콜이다. 첫 번째는 게이트웨이가 정상적으로 동작하는 형태에서의 전송 프로토콜이고, 두 번째는 게이트웨이가 장애로 인해 동작하지 못하는 경우의 전송프로토콜이다.

먼저 [그림 6]의 (a)에서는 게이트웨이(GW)가 정상적으로 동작하는 경우를 보여주고 있다. 메디컬 센서노드(N₁)가 GW로 생체정보 데이터를 전송하기 위한 연결요청(Con.req)을 하게 된다. 연결요청을 받은 게이트웨이는 연결이 응답(Con.res)을 보내게 된다.

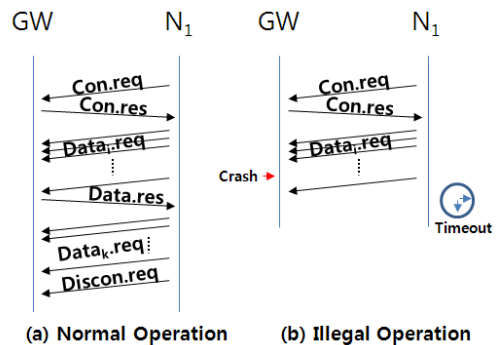


그림 6. 분산보관 프로토콜

이때 N_1 의 메시지를 모두 받을 수 있는 전송범위에 있는 다른 노드들도 이 메시지를 받으며, [표 2]와 같은 형태의 이웃노드테이블에 노드주소와 생체유형에 맞는 data-rate, 그리고 생체정보를 보관하는 저장 공간 테이블의 여유 공간 크기 row-available을 받아 등록한다.

표 2. 이웃노드테이블(NextNodeTable, NNT)

node-address	data-rate	row-available
N_1	20	1,000
N_2	1,000	1,000
N_3	28,800,000	1,000

연결이 설정된 이후부터 Data.req 메시지에 생체정보를 저장하여 전송한다. 이때 전송효율을 위해서 슬라이딩 윈도우를 적용하여 응답(Data.res)이 즉시 오지 않더라도 다량의 생체정보를 수신할 수 있다. 그러나 만약 [그림 6]의 (b)에서처럼 일정 시간 내에 응답(Data.res)이 오지 않게 되면 타임아웃이 처리되어 N_1 은 게이트웨이 장애로 인지한다. 장애가 인지되면 [그림 7]과 같은 노드의 로컬 저장 공간 테이블에 측정된 생체정보를 임시로 보관하기 시작한다. 이때 전체 저장 공간에서 row-available의 크기를 갱신하고 관리한다. row-available의 크기가 전체크기의 90%에 이르면 분실을 막기 위해 [그림 8]의 노드선정 알고리즘을 이용하여 대체노드 주소를 획득하고 Token.req를 생성하여 대체노드 요청을 전송한다. [그림 9]의 (b)의 경우를 보면 N_1 은 이웃한 N_2 와 N_3 중에서 선정된 대체노드가 N_2 이므로 N_2 로 Token.req를 전송한다.

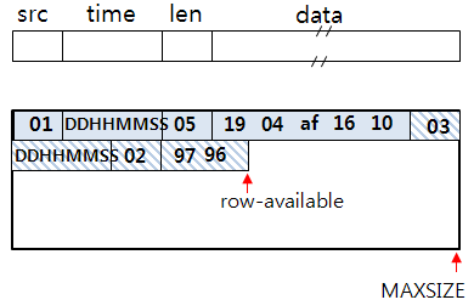


그림 7. 노드의 저장공간 테이블구조

N_2 는 자신이 대체노드가 될 수 있는지 확인하여 수락한다. 이때 다른 메디컬 센서노드에게 생체정보 데이터를 보내라는 의미로 Archive.req를 브로드캐스트 하게 된다.

Node Select Algorithm

```

1 find-entry ← -1
2 current-free-size ← -1
3 for (row←0; row < MAXSIZE; row++) begin
4   A ← NNT[row].data-rate
5     * NNT[row]. row-available
6   if (A > current-free-size) then
7     find-entry ← row
8     current-free-size ← A
9   end if
10 end for
11 next-node ← NNT[find-entry].node-address
12 Token←Make-Message(TokenID, next-node, myrow)
    
```

그림 8. 노드선정 알고리즘

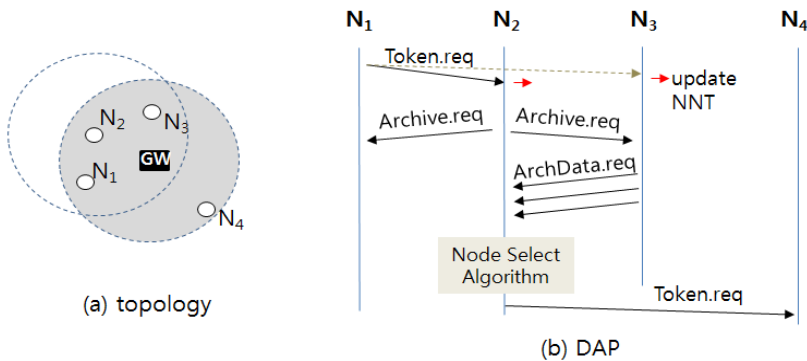


그림 9. 분산 보관 프로토콜(DAP)

모든 노드가 Archive.req를 받으면 생체정보를 게이트웨이 대신 대체노드로 전송하게 된다. [그림 9]에서 보면 N_2 가 대체노드이기 때문에 Archive.req를 받은 N_3 는 N_2 로 데이터를 전송한다. 이때 일반적인 메시지와 구별하기 위해서 Data.req 가 아닌 ArchData.req로 보낸다.

그러나 이 과정에서 N_2 의 로컬 저장공간에 지속적으로 생체정보를 저장한다면 row-available의 값이 전체 크기의 90%에 이르게 된다. 이와 같은 경우에는 대체노드 선정알고리즘을 다시 이용해서 다음의 대체노드 N_4 를 선정하고 Token.req를 생성해 전송한다. 다음은 이러한 송·수신 과정을 센서노드와 게이트웨이간의 처리흐름을 상태로 표현한 것이다.

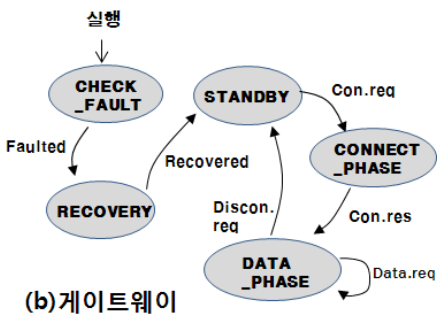
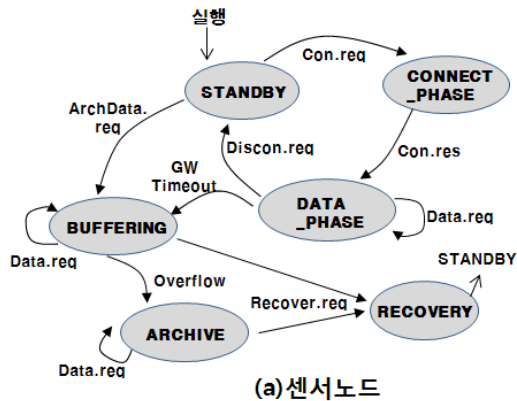


그림 10. 센서노드의 상태도(state diagram)

[그림 10]의 (a)에서 살펴보면 센서노드는 6개의 상태로 동작한다. STANDBY상태에서 전송할 생체정보가 있다면 Con.req를 게이트웨이로 보낸 뒤에 CONNECT_PHASE로 변경되고, 다른 센서노드로부터

ArchData.req를 받게 되면 게이트웨이가 장애중이라는 사실을 알 수 있으므로 BUFFERING 상태로 천이한다. 만약 저장 공간의 여유 공간이 부족하게 되면 ARCHIVE 상태로 천이되어 다른 센서노드의 저장 공간으로 전송하여 분산 보관하도록 한다. (b)의 게이트웨이의 상태를 보면 실행된 직후에 장애가 발생되었는지를 확인하는 상태(CHECK_FAULT)를 거치고 장애가 발생되었었다면 RECOVERY상태로 천이하여 처리하게 된다. 이후에 STANDBY상태로 천이한다. 다음 장에서는 이와 같은 상태를 이용하여 본 논문에서 정의한 DAP가 효과적으로 분산보관을 하는지 실험을 통하여 확인하였다.

IV. 실험 및 평가

1. 시스템 구성 및 구현

게이트웨이가 정상적으로 동작할 때부터 장애가 발생하여 DAP가 수행될 때까지의 효과성을 확인하기 위해 한백전자의 센서노드에 TinyOS2.1 센서 애플리케이션을 구현하였다. 또한 게이트웨이는 노트북 컴퓨터를 이용하여 Cygwin 리눅스 환경에서 개발하였다. 다음 [그림 11]은 구현한 센서노드와 게이트웨이이다.



그림 11. 실험환경(test-bed)

[그림 11]의 센서노드는 한백전자의 ZigbeXII 모뎀에

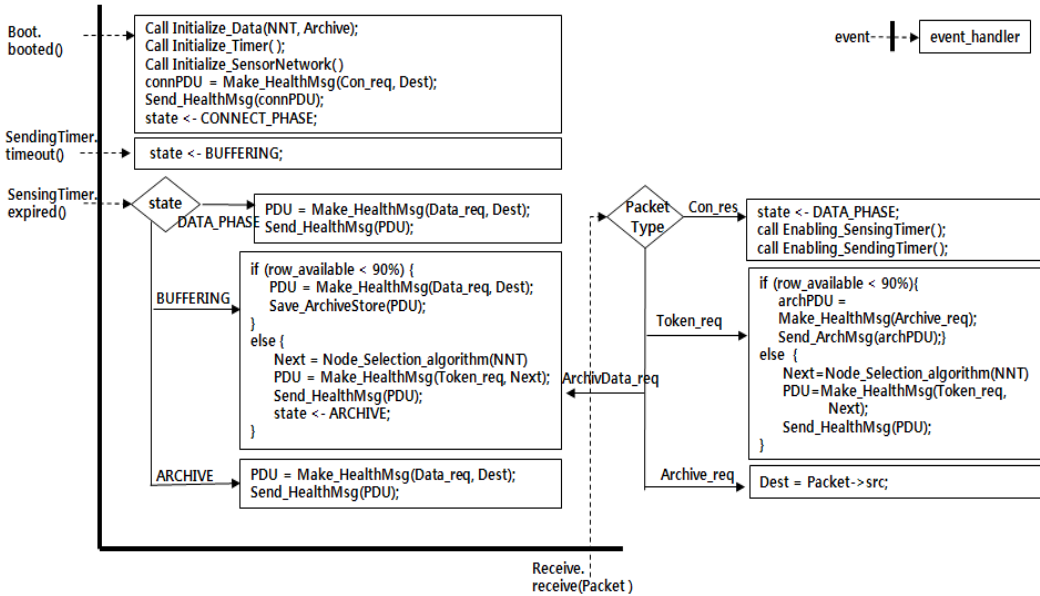


그림 12. 센서노드의 분산보관 처리 흐름도

산소포화도(SpO₂)센서노드(①), 심전도(ECG)센서노드(②), 혈압센서노드(③), 체온센서노드(④)을 이용하여 각 생체정보를 측정하도록 하였다. 센서노드의 DAP는 이클립스 개발 툴을 이용하여 TinyOS 2.1 환경에서 동작될 수 있도록 nesc로 구현하였다. nesc 프로그램은 이벤트 처리방식으로 구동되기 때문에 처리 알고리즘을 이벤트 핸들러 형태로 [그림 12]에서처럼 처리 흐름도로 표현하였다. 또 게이트웨이(⑤)는 리눅스 플랫폼인 Cygwin에서 장애와 장애복구 기능을 제공하는 프로그램으로 구현하였으며, 시리얼로 연결된 베이스노드를 통해 지그비 센서 네트워크와 연결되어 있다.

센서노드의 처리 흐름도를 살펴보면 [표 2]와 [그림 7]에서 정의한 NNT 테이블과 Archive를 사용하고 있다. 센서노드가 부팅할 때 booted() 이벤트를 처리하기 위해 NNT, Archive를 초기화시키고 Timer와 센서네트워크를 담당하는 모듈을 초기화한다. 또한 게이트웨이로 Con_req 패킷을 전송하고 state를 CONNECT_PHASE 상태로 설정한다.

센서노드는 2개의 Timer를 가지고 운영한다. SensingTimer는 생체정보를 주기적으로 측정하여 전송하는데 사용하며, SendingTimer는 게이트웨이의 장

애를 탐지하기 위해 사용한다. 게이트웨이로부터 데이터가 수신되면 Receive.receive() 이벤트가 발생되며 각 상태와 패킷타입에 따라 동작을 달리한다.

2. 노드선정 적응률의 실험 및 평가

구현된 시스템을 이용하여 노드 선정 알고리즘의 적응률을 살펴보았다. 이 실험에서는 [그림 13]에서처럼 GW의 네트워크 기능을 비활성화시켜 발생될 두 가지 경우의 실험을 하였다. 첫 번째 case 1은 N₁이 장애를 탐지하고 노드선정 알고리즘을 통해 N₂노드로 정확히 선정되는가를 확인하는 실험이다.

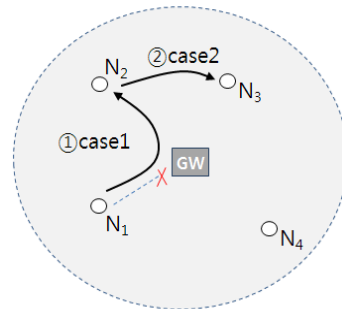


그림 13. 적응률 실험 시나리오

두 번째 case 2는 N₁에서 N₂로 송신되어 오는 데이터를 N₂가 보관하다가 여유공간의 부족으로 노드선정 알고리즘에 의해 다음 대체노드를 선정하는가를 확인하는 실험이다. 이 실험은 데이터의 측정주기가 제일 빠른 순서로 ECG를 N₁으로 정하고 SpO₂를 N₂로 정했으며 혈압은 N₃로 정하였다. 그렇게 하는 이유는 GW가 장에서 로컬저장소의 오버플로우를 쉽게 유발시킬 수 있기 때문에 실험을 효과적으로 할 수 있기 때문이다. N₁에서는 400건의 ECG 데이터를 측정하여 전송하였고 SpO₂는 7건의 데이터를 측정하도록 하였다. 또한 혈압은 측정주기가 매우 늦기 때문에 측정을 하지 않는다고 가정하여 실험하였다. 그 결과 [그림 14]와 같은 결과를 얻을 수 있었다. 각 노드에 보관된 데이터를 직접 볼 수 없기 때문에 GW를 통해 수집한 내용을 확인한 화면이며, 해당 노드가 보관하고 있는 내용을 확인할 수 있다. 따라서 [그림 14]의 일부를 설명하면 SN1으로부터 온 수집내용은 N₁(01)은 자체보관중인 데이터이며, 23일 21시 14분 58초에 50개의 ECG데이터를 보관하였고 그 내용은 “94 92 ...” 이다.

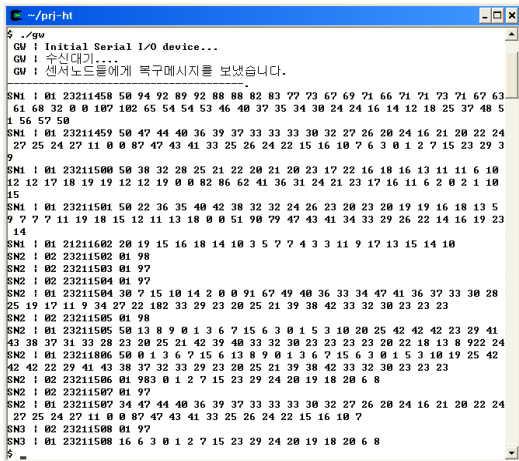


그림 14. 실험결과 화면

[그림 14]에서처럼 N₁에서 측정된 데이터는 N₂를 거쳐 N₃에 각각 분산 보관되어 있는 것을 알 수 있었다. 이와 같은 방법으로 총 10회에 걸쳐 총 4000건의 데이터를 생성하여 실험한 결과 다음 [표 3]의 결과를 얻었

다. 즉 의도한대로 N₁에서 측정된 데이터가 각각 N₁, N₂, N₃에 분산 보관됨을 알 수 있어 노드선정 알고리즘이 의도한대로 작동하고 있음을 확인 할 수 있었다. 물론 3회와 8회의 경우는 중간에 데이터를 일부 분실하여 회수하지 못하는 경우가 있는 것도 알 수 있었다.

표 3. 적중률 실험결과

실험	데이터 건수				총계
	N ₁ 데이터생성	N ₁ 로컬저장	N ₂ 저장 (case1)	N ₃ 저장 (case2)	
1	400	220	164	16	400
2	400	220	163	17	400
3	400	220	164	7	391
4	400	220	164	16	400
5	400	220	164	16	400
6	400	220	164	16	400
7	400	220	164	16	400
8	400	220	161	17	398
9	400	220	164	16	400
10	400	220	164	16	400
총계	4,000	-	-	-	3,989

3. 분산 보관과정에서의 회수율 실험 및 평가

다음 실험은 제한한 분산보관 프로토콜이 실행되는 과정에서 분실되는 데이터가 어느 정도인지를 측정하는 실험이며, 분산보관 과정의 신뢰도를 측정하기 위함이다.

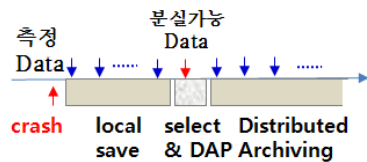


그림 15. 분실가능 시나리오

이 실험은 [그림 15]에서처럼 데이터가 빠르게 측정되는 상황에서 로컬저장소(local save)의 공간부족으로 인해 대체노드를 선정(select)하고 선정된 대체노드로 DAP를 이용하여 데이터를 전송할 때 분실 가능한 Data가 있을 수 있다. 만약에 분실이 발생된다면 100% 회수가 어렵다. 실험은 각 노드가 저장한 데이터를 게이트웨이가 수집하여 분석하면 분실 가능한 Data가 있는지 확인할 수 있다. 이러한 실험결과를 [그림 14]에서

볼 수 있다. 이와 같은 방법으로 실험하여 [그림 16]의 회수율 결과를 얻었다.

회수율 측정실험은 총 10회 하였는데 1회, 4회, 6회, 9회는 100%회수를 하였고, 2회, 3회, 7회는 99%, 5회와 10회는 98%의 결과를 얻었다. 전체적으로 평균 99.3%의 회수결과를 얻을 수 있어 분산보관 프로토콜과정에서 분실되는 분실율은 0.7%로 매우 적음을 알 수 있었다. 이로써 DAP는 게이트웨이의 장애에 대해 효과적인 임시 대체수단이 될 수 있음을 확인할 수 있었다.

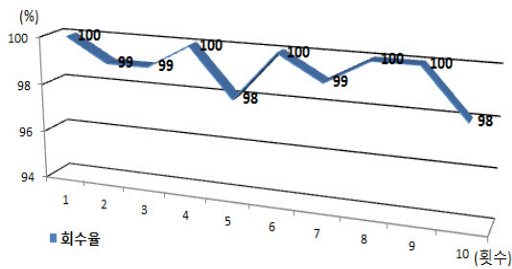


그림 16. 회수율 실험결과

V. 결론

메디컬 센서노드에서 발생하는 생체정보 데이터는 매우 중요한 개인정보이다. 그래서 안전하게 전송되는 것이 중요하다. 만약 게이트웨이에 장애가 발생되어 데이터 수신에 불가능하게 되면, 이 사실을 알지 못하는 각 메디컬 센서노드들은 계속해서 개인의 생체정보 데이터를 전송하기 때문에 중간에 분실될 수 있다. 본 연구에서는 이러한 문제를 해결하기 위해 메디컬 센서노드간의 생체유형별 특징을 이용해 분산하여 보관하는 기법을 제안하였다. 또한 제안된 기법의 유효성을 확인하기 위해 TinyOS2.1 환경에서 nesc로 구현하였다.

구현한 시스템을 이용하여 실시한 적중률 실험에서는 10회 모두 대체노드 선정 알고리즘이 정확히 동작함을 알 수 있어 대체노드 선정 알고리즘과 전송프로토콜의 유효성을 확인하였으며, 회수율 실험에서는 각각 총 10번의 실험을 실시하여 평균 99.3%의 회수율을 얻어 분산보관 프로토콜 과정에서의 분실이 매우 적음을

알 수 있었다.

참고 문헌

- [1] 이희정, 강신재, 장형근, 정창원, 주수중, “u-Home 환경에서 멀티센서기반 u-Care 시스템구현”, 한국인터넷정보학회논문지, 제12권, 제2호, pp.135-147, 2011.
- [2] 이병문, 임현철, 강운구, “유헬스에서 안전한 생체 정보전송을 위한 동적인 유효세션기반의 상호인증 프로토콜”, 한국콘텐츠학회논문지, 제11권, 제2호, pp.142-151, 2011.
- [3] W. Su William and S. J. Lee, “An Adaptive and Fault-Tolerant Gateway Assignment in Sensor Networks,” Proc. of Military Communications Conference, Vol.2, pp.642-648, 2004.
- [4] G. Gupta and M. Younis, “Fault-Tolerant Clustering of Wireless Sensor Networks,” Proc. of IEEE Wireless Communications & Networking Conference - WCNC, pp.1579-1584, 2003.
- [5] D. Baghyalakshmi, J. Ebenezer, and S. A. V. Satyamurthy, “LOW LATENCY AND ENERGY EFFICIENT ROUTING PROTOCOLS FOR WIRELESS SENSOR NETWORKS,” Proc. of International Conference Wireless Communication and Sensor Computing, pp.1-6, 2010.
- [6] S. Ho and G. Yu, “An Energy Efficient Data Storage Policy for Object Tracking Wireless Sensor Network,” Proc. of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing(SUTC 2006), pp.20-25, 2006.
- [7] 박용훈, 윤종현, 서동민, 김준, 유재수, “센서 네트워크에서 저장 공간의 활용성과 에너지 효율성을 위한 시간 매개변수 기반의 데이터 중심 저장 기법”, 정보과학회논문지, 제36권, 제2호, pp.99-111,

2009.

[8] K. Park and R. Elmasri, "Effects of Storage Architecture on Performance of Sensor Network Queries," Information Networking. Advanced in Data Communications Wireless Networks Lecture Notes in Computer Science, Vol.3961, pp.247-256, 2006.

[9] N. Gollan and Jens B. Schmitt, "Energy-Efficient TDMA Design Under Real-Time Constraints in Wireless Sensor Networks," International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp.80-87, 2007.

[10] 장희태, 이병문, "홈 헬스용 메디컬 센서노드의 생체유형별 효율적 분산저장", 2011년도 한국인터넷정보학회 추계학술발표대회, 제12권, 제2호, pp.119-120, 2011.

[11] 박민진, 정문희, "유헬스케어(u-Health Care)도 입진후 방문건강관리 건강지표의 관찰", 한국생활환경학회지, 제15권, 제1호, pp.42-50, 2011.

[12] 이효준, *센서 네트워크 저장 시스템을 위한 소실 데이터 근사 복구 기법*, 충북대학교 석사학위논문, pp.40-48, 2011.

[12] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-Centric Storage in Sensornets," Computer communication review, Vol.33, No.1, pp.137-142, 2003.

[13] M. Aly, P. K. Chrysanthis, and K. Pruthis, "Decomposing data-centric storage query hot-spots in sensor networks," Mobile and Ubiquitous Systems - Workshops, pp.1-9 2006.

[14] S. Rantnasamy, B. Karp, L. Yin, F. Uu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage," Mobile Computing and Networking - MOBICOM, pp.78-87, 2002.

저 자 소 개

이 영 호(Young-Ho Lee)

정희원



- 2005년 8월 : 아주대학교 의과대학 의료정보학과(이학박사)
- 1999년 ~ 2002년 : IBM Korea BI & CRM
- 현재 : ISO/TC215 전문위원
- 현재 : 스마트의료정보표준포럼

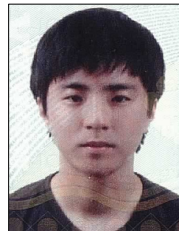
사무총장

- 2002년 3월 ~ 현재 : 가천의과학대학교 정보공학부 부교수

<관심분야> : 데이터마이닝, 유헬스, 의료정보

장 희 태(Hee-Tae Jang)

준희원



- 2009년 2월 : 가천의과학대학교 의료공학부 IT학과
- 2010년 11월 : 가천의과학대학교 유-헬스케어연구소 연구원

<관심분야> : 유헬스, 센서네트워크

이 병 문(Byung-Mun Lee)

정희원



- 1988년 2월 : 동국대학교 전자계산학과(공학사)
- 1990년 2월 : 서강대학교 전자계산학과(공학석사)
- 2007년 2월 : 인천대학교 컴퓨터공학과(공학박사)

- 1990년 ~ 1997년 : LG전자(구 LG정보통신) 선임연구원

- 1998년 3월 ~ 현재 : 가천의과학대학교 정보공학부 교수

<관심분야> : 유헬스, 센서네트워크, 센서운영체제