



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2010년10월29일  
 (11) 등록번호 10-0990555  
 (24) 등록일자 2010년10월21일

- (51) Int. Cl.  
*H04L 29/06* (2006.01)
- (21) 출원번호 10-2004-7004351
- (22) 출원일자(국제출원일자) 2002년09월24일  
 심사청구일자 2007년09월11일
- (85) 번역문제출일자 2004년03월25일
- (65) 공개번호 10-2004-0039414
- (43) 공개일자 2004년05월10일
- (86) 국제출원번호 PCT/US2002/030387
- (87) 국제공개번호 WO 2003/027904  
 국제공개일자 2003년04월03일
- (30) 우선권주장  
 09/967,406 2001년09월27일 미국(US)
- (56) 선행기술조사문헌  
 US6157935 A\*  
 US6175560 B1  
 US6286015 B1  
 \*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
**칼컴 인코포레이티드**  
 미국 캘리포니아 샌디에고 모어하우스  
 드라이브5775 (우 92121-1714)
- (72) 발명자  
**맥그래스, 게리, 지.**  
 미국92009-5415캘리포니아칼스배드할싱코트521
- (74) 대리인  
**남상선**

전체 청구항 수 : 총 37 항

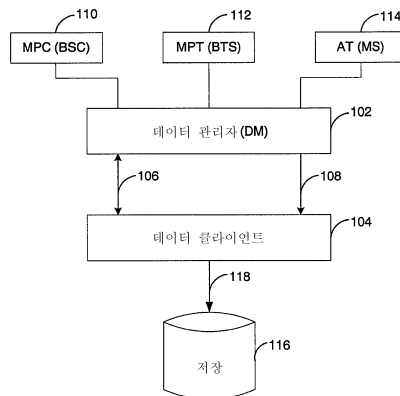
심사관 : 오제욱

**(54) 통일된 데이터 교환 및 저장 포맷을 제공하는 방법 및시스템**

**(57) 요약**

네트워크에서 데이터 교환 및 데이터 저장 양쪽 모두의 프로토콜을 위한 통일된 포맷을 제공하는 방법 및 시스템이 제공된다. 통일된 데이터 포맷은 실시간 데이터의 가변적인 크기를 수신, 분석, 디코딩, 및 저장하기 위한 효율적인 프로토콜을 제공한다.

**대표도 - 도1**



## 특허청구의 범위

### 청구항 1

네트워크에서 데이터를 교환하기 위한 장치로서,

데이터 교환 프로토콜에 따라 네트워크 접속을 통해 데이터 생성기(producer)로부터 데이터를 수신하도록 구성되는 수신기;

상기 수신기에 동작가능하게 접속되는 안테나; 및

상기 안테나에 동작가능하게 접속되며, 데이터 저장 프로토콜에 따라 상기 데이터를 저장 유닛에 전송하도록 구성되는 송신기를 포함하는데,

상기 네트워크 접속을 통해 수신된 상기 데이터 및 상기 저장 유닛에 저장된 상기 데이터는, 상기 데이터 교환 프로토콜과 상기 데이터 저장 프로토콜 모두를 따르기 위해, 다수의 필드들을 포함하는 통일된(unified) 데이터 포맷을 갖는 데이터 구조에 포함되며, 상기 통일된 데이터 포맷은 전체 포맷과 레코드 포맷을 포함하는,

데이터 교환 장치.

### 청구항 2

제 1항에 있어서, 상기 전체 포맷은 레코드 섹션을 포함하는, 데이터 교환 장치.

### 청구항 3

제 1항에 있어서, 상기 전체 포맷은 헤더 섹션, 레코드 섹션, 및 테일(tail) 섹션을 포함하는, 데이터 교환 장치.

### 청구항 4

제 3항에 있어서, 상기 레코드 섹션의 레코드는 레코드 헤더, 레코드 보디(body), 및 레코드 테일을 가지는 레코드 포맷을 포함하는, 데이터 교환 장치.

### 청구항 5

제 1항에 있어서, 상기 네트워크는 무선 네트워크를 포함하는, 데이터 교환 장치.

### 청구항 6

제 1항에 있어서, 상기 네트워크는 광 네트워크를 포함하는, 데이터 교환 장치.

### 청구항 7

네트워크에서 데이터를 포맷하기 위한 방법으로서,

전체 포맷과 레코드 포맷을 포함하는 통일된 데이터 포맷을 제공하는 단계;

상기 전체 포맷에 헤더 섹션 -상기 헤더 섹션은 데이터 구조에 포함된 하나 또는 그 이상의 레코드들의 포맷을 식별하는 하나 또는 그 이상의 필드들을 포함함- 을 제공하는 단계;

상기 전체 포맷에 레코드 섹션 -상기 레코드 섹션은 상기 하나 또는 그 이상의 레코드들을 포함함- 을 제공하는 단계; 및

상기 전체 포맷에 테일 섹션 -상기 테일 섹션은 데이터 무결성(integrity) 검사 정보를 유지하기 위한 하나 또는 그 이상의 필드들을 포함함- 을 제공하는 단계를 포함하며, 이를 통해 상기 전체 포맷은 데이터 교환 및 데이터 저장 모두를 위해 사용될 수 있는,

데이터 포맷 방법.

### 청구항 8

제 7항에 있어서,

상기 레코드 섹션에 레코드 헤더, 레코드 보디 및 레코드 테일을 포함하는 레코드를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 9**

제 7항에 있어서,

상기 전체 포맷에 상기 데이터에 대한 버전을 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 10**

제 7항에 있어서,

상기 레코드 섹션에 대한 크기를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 11**

제 10항에 있어서,

상기 레코드 섹션에 대한 크기를 식별하는 상기 정보에 할당된 비트들의 수에 대한 표시를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 12**

제 7항에 있어서,

레코드-포맷 버전을 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 13**

제 7항에 있어서,

상기 전체 포맷에 선택 필드가 사용되는지 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 14**

제 7항에 있어서,

상기 전체 포맷에 상기 데이터의 종료를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 15**

제 7항에 있어서,

상기 레코드 섹션의 첫번째 레코드를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 16**

제 7항에 있어서,

상기 레코드 섹션의 마지막 레코드를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 17**

제 7항에 있어서,

상기 헤더 섹션의 종료를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 18**

제 7항에 있어서,

순환 중복 검사(CRC) 값에 대한 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 19**

제 7항에 있어서,  
상기 레코드 섹션의 레코드의 크기를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 20**

제 7항에 있어서,  
상기 레코드 섹션의 상기 레코드의 크기를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 21**

제 7항에 있어서,  
상기 레코드 섹션의 상기 레코드에 대한 포맷을 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 22**

제 7항에 있어서,  
상기 레코드 섹션의 레코드 타입의 크기를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 23**

제 22항에서,  
상기 레코드 섹션의 상기 레코드의 레코드 타입을 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 24**

제 7항에 있어서,  
상기 전체 포맷의 상기 데이터에 대한 포맷을 식별하는 정보를 상기 헤더 섹션에 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 25**

제 7항에 있어서,  
상기 전체 포맷의 상기 데이터에 대한 타임 스탬프(time stamp)를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 26**

제 7항에 있어서,  
상기 네트워크의 데이터 서브시스템의 소스를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 27**

제 7항에 있어서,  
상기 네트워크의 데이터 액세스 단말에 대한 목표(target)를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 28**

제 7항에 있어서,  
상기 네트워크의 서브시스템의 섹터를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 29**

제 7항에 있어서,

상기 레코드 섹션의 상기 레코드에 대한 순환 중복 검사(CRC) 값을 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 30**

제 7항에 있어서,

상기 레코드 섹션의 상기 레코드의 종료를 식별하는 정보를 제공하는 단계를 더 포함하는, 데이터 포맷 방법.

**청구항 31**

네트워크에서 데이터를 포맷하기 위한 장치로서,

전체 포맷과 레코드 포맷을 포함하는 통일된 데이터 포맷을 제공하기 위한 수단;

상기 전체 포맷에 헤더 섹션 -상기 헤더 섹션은 데이터 구조에 포함된 하나 또는 그 이상의 레코드들의 포맷을 식별하는 하나 또는 그 이상의 필드들을 포함함- 을 제공하기 위한 수단;

상기 전체 포맷에 레코드 섹션 -상기 레코드 섹션은 상기 하나 또는 그 이상의 레코드들을 포함함- 을 제공하기 위한 수단; 및

상기 전체 포맷에 테일 섹션 -상기 테일 섹션은 데이터 무결성 검사 정보를 유지하기 위한 하나 또는 그 이상의 필드들을 포함함- 을 제공하기 위한 수단을 포함하며, 이를 통해 상기 전체 포맷은 데이터 교환 및 데이터 저장 모두를 위해 사용될 수 있는,

데이터 포맷 장치.

**청구항 32**

네트워크에서 데이터 포맷 방법을 수행하기 위한 명령들을 포함하는 컴퓨터 판독가능 매체로서, 상기 방법은,

전체 포맷과 레코드 포맷을 포함하는 통일된 데이터 포맷을 제공하는 단계;

상기 전체 포맷에 헤더 섹션 -상기 헤더 섹션은 데이터 구조에 포함된 하나 또는 그 이상의 레코드들의 포맷을 식별하는 하나 또는 그 이상의 필드들을 포함함- 을 제공하는 단계;

상기 전체 포맷에 레코드 섹션 -상기 레코드 섹션은 상기 하나 또는 그 이상의 레코드들을 포함함- 을 제공하는 단계; 및

상기 전체 포맷에 테일 섹션 -상기 테일 섹션은 데이터 무결성 검사 정보를 유지하기 위한 하나 또는 그 이상의 필드들을 포함함- 을 제공하는 단계를 포함하며, 이를 통해 상기 전체 포맷은 데이터 교환 및 데이터 저장 모두를 위해 사용될 수 있는,

컴퓨터 판독가능 매체.

**청구항 33**

네트워크에서 데이터를 포맷하기 위한 장치로서,

메모리 유닛; 및

상기 메모리 유닛에 통신가능하게 접속된 디지털 신호 프로세서(DSP) 유닛을 포함하고, 상기 DSP 유닛은,

전체 포맷과 레코드 포맷을 포함하는 통일된 데이터 포맷을 제공할 수 있고,

상기 전체 포맷에 헤더 섹션 -상기 헤더 섹션은 데이터 구조 내에 포함된 하나 또는 그 이상의 레코드들의 포맷을 식별하는 하나 또는 그 이상의 필드들을 포함함- 을 제공할 수 있고,

상기 전체 포맷에 레코드 섹션 -상기 레코드 섹션은 상기 하나 또는 그 이상의 레코드들을 포함함- 을 제공할 수 있고, 그리고

상기 전체 포맷에 테일 섹션 -상기 테일 섹션은 데이터 무결성 검사 정보를 유지하기 위한 하나 또는 그 이상의 필드들을 포함함- 을 제공할 수 있고, 이를 통해 상기 전체 포맷은 데이터 교환 및 데이터 저장 모 두를 위해 사용될 수 있는,

데이터 포맷 장치.

**청구항 34**

데이터 클라이언트 및 데이터 관리자를 포함하는 네트워크에서 데이터를 처리하기 위한 방법으로서,

상기 데이터 클라이언트에서, 데이터 교환 프로토콜에 따라 네트워크 접속을 통해 상기 데이터 관리자로부터 데이터를 수신하는 단계; 및

데이터 저장 프로토콜에 따라 상기 데이터를 저장 유닛에 저장하는 단계를 포함하며,

상기 네트워크 접속을 통해 수신된 상기 데이터 및 상기 저장 유닛에 저장된 상기 데이터는, 상기 데이터 교환 프로토콜과 상기 데이터 저장 프로토콜 모두를 따르기 위해, 다수의 필드들을 포함하는 통일된 데이터 포맷을 갖는 데이터 구조에 포함되며, 상기 통일된 데이터 포맷은 전체 포맷과 레코드 포맷을 포함하는,

데이터 처리 방법.

**청구항 35**

데이터 클라이언트 및 데이터 관리자를 포함하는 네트워크에서 데이터 처리 방법을 수행하기 위한 명령들을 포함하는 컴퓨터 판독가능 매체로서, 상기 방법은,

상기 데이터 클라이언트에서, 데이터 교환 프로토콜에 따라 네트워크 접속을 통해 상기 데이터 관리자로부터 데이터를 수신하는 단계; 및

데이터 저장 프로토콜에 따라 상기 데이터를 저장 유닛에 저장하는 단계를 포함하며,

상기 네트워크 접속을 통해 수신된 상기 데이터 및 상기 저장 유닛에 저장된 상기 데이터는, 상기 데이터 교환 프로토콜과 상기 데이터 저장 프로토콜 모두를 따르기 위해, 다수의 필드들을 포함하는 통일된 데이터 포맷을 갖는 데이터 구조에 포함되며, 상기 통일된 데이터 포맷은 전체 포맷과 레코드 포맷을 포함하는,

컴퓨터 판독가능 매체.

**청구항 36**

데이터 클라이언트 및 데이터 관리자를 포함하는 데이터 처리 시스템들의 네트워크에서 데이터를 처리하기 위한 장치로서,

상기 데이터 클라이언트에서, 데이터 교환 프로토콜에 따라 네트워크 접속을 통해 상기 데이터 관리자로부터 데이터를 수신하기 위한 수단; 및

데이터 저장 프로토콜에 따라 상기 데이터를 저장 유닛에 저장하기 위한 수단을 포함하며,

상기 네트워크 접속을 통해 수신된 상기 데이터 및 상기 저장 유닛에 저장된 상기 데이터는, 상기 데이터 교환 프로토콜과 상기 데이터 저장 프로토콜 모두를 따르기 위해, 다수의 필드들을 포함하는 통일된 데이터 포맷을 갖는 데이터 구조에 포함되며, 상기 통일된 데이터 포맷은 전체 포맷과 레코드 포맷을 포함하는,

데이터 처리 장치.

**청구항 37**

데이터 클라이언트 및 데이터 관리자를 포함하는 네트워크에서 데이터를 처리하기 위한 장치로서,

메모리 유닛; 및

상기 메모리 유닛에 통신가능하게 접속된 디지털 신호 프로세서(DSP)를 포함하고, 상기 DSP 유닛은,

상기 데이터 클라이언트에서, 데이터 교환 프로토콜에 따라 네트워크 접속을 통해 상기 데이터 관리자로부터 데이터를 수신할 수 있고, 그리고

데이터 저장 프로토콜에 따라 상기 데이터를 저장 유닛에 저장할 수 있으며, 상기 네트워크 접속을 통해 수신된 상기 데이터 및 상기 저장 유닛에 저장된 상기 데이터는, 상기 데이터 교환 프로토콜과 상기 데이터 저장 프로토콜 모두를 따르기 위해, 다수의 필드들을 포함하는 통일된 데이터 포맷을 갖는 데이터 구조에 포함되며, 상기 통일된 데이터 포맷은 전체 포맷과 레코드 포맷을 포함하는, 데이터 처리 장치.

**청구항 38**

삭제

**청구항 39**

삭제

**청구항 40**

삭제

**명세서**

**기술 분야**

[0001] 본 명세서에서 개시되는 실시예들은 일반적으로 통신 시스템에 관한 것이며, 더 상세히 말하자면, 데이터 교환 및 데이터 저장 프로토콜들에 대하여 통일된 포맷을 제공하는 방법 및 시스템에 관한 것이다.

**배경 기술**

[0002] 네트워크에서 데이터 교환을 위한 현존하는 포맷들은 데이터 저장을 위한 현존한 포맷과 상당히 다르며, 그로 인해, 데이터 클라이언트에서 수신된 데이터는 저장되기에 앞서 재포맷되어야 한다. 현존하는 데이터 교환 포맷들 및 디스크 저장 포맷들은 작거나 큰 데이터 크기 중 어느 한쪽을 위해 최적화되었을 뿐 둘 모두를 위해 최적화되지는 않았다. 현재의 데이터 저장 포맷들은 네트워크 접속을 통해서 실시간으로 크기가 변하는 데이터를 견고하게(robustly) 교환하는데 상기 포맷이 유용하도록 하는데 필요한 특징들이 부족하고, 현재의 네트워크 프로토콜들은 데이터를 견고하게 저장하는데 필요한 특징들이 부족하다.

[0003] 그러므로, 통일된 포맷이 비용-효과적인 구현, 다른 틀들과의 호환성, 및 코드-재사용 가능성을 유도하도록 하기 위해 데이터 교환 및 저장 모두에 적합한 통일된 포맷이 해당 분야에 필요하다. 또한, 하드웨어 및/또는 소프트웨어 운용 비용을 줄이기 위해서 효율적인 데이터 압축/압축해제를 위한 통일된 데이터 교환 및 저장 포맷이 필요하다.

**발명의 상세한 설명**

[0004] 본 명세서에서 개시된 실시예들은 데이터 클라이언트 및 데이터 관리자를 포함하는 네트워크에서 데이터를 포맷하기 위한 방법 및 시스템을 제공함으로써 위에서 설명된 필요요건을 해결한다. 상기 방법 및 시스템은 헤더 섹션, 레코드 섹션, 및 테일 섹션을 구비하는 통일된 데이터 포맷을 제공함으로써, 데이터 포맷이 데이터 교환 및 데이터 저장 모두를 위해 적용가능하도록 한다.

[0005] 본 발명의 또 다른 양상에서는, 데이터 클라이언트 장치가 네트워크 접속을 통해 데이터 관리자로부터 데이터를 수신하도록 구성된 수신기와, 그 데이터를 저장 유닛에 전송하도록 구성된 송신기를 구비함으로써, 네트워크 접속을 통해 수신되는 데이터와 저장 유닛에 전송되는 데이터가 동일한 포맷을 갖도록 한다.

[0006] 본 발명의 또 다른 양상에서는, 네트워크에서 데이터를 포맷하기 위한 데이터 구조가 헤더 섹션, 레코드 섹션, 및 테일 섹션을 구비함으로써, 상기 데이터 구조가 데이터 교환 및 데이터 저장 프로토콜 모두에 적용가능하도록 한다.

**실시예**

[0014] 도 1은 접속부(106 및 108)를 통한, 데이터 관리자(102)와 데이터 클라이언트(104) 사이에서의 예시적인 네트워크 인터페이스를 나타내고 있다. 데이터 관리자(102)는 웹 서버나 액세스 네트워크(AN)와 같은 네트워크 엘리

먼트를 구비할 수 있다. 데이터 관리자(102)는 모뎀 풀 제어기(MPC : modem pool controller:110), 모뎀 풀 트랜시버(MPT:112), 및/또는 액세스 단말(AT:114)와 같은 데이터 소스들과 통신 중일 수 있다. 데이터 클라이언트(104)는 개인용 컴퓨터, HDR 분석 툴(HAT), 또는 데이터 관리자(102)로부터의 데이터를 고유의 데이터에 통합하는 또 다른 데이터 서버와 같은 클라이언트 단말을 포함할 수 있다.

[0015] 일실시예에서, 데이터 관리자(102)와 데이터 클라이언트(104) 사이의 데이터 통신은 두 가지 프로토콜, 즉, 데이터 제어를 위한 데이터 관리자 프로토콜(DMP) 및 데이터 전달 및 저장을 위한 이진 데이터 교환 포맷(BDEF)에 의해 제어될 수 있다. DMP 프로토콜은, 공동-계류 중인 특허 출원에 상세히 설명되고 있는 바와 같이, 데이터 관리자 제어 프로토콜(DMCP) 및 데이터 관리자 복구 프로토콜(DMDP)을 포함할 수 있다.

[0016] **데이터 관리자 프로토콜(Data Manager Protocol)**

[0017] DMP 프로토콜은 쉬운 구현 및 디버깅(debugging)을 위해 설계된다. 이 설계의 목적은 명령 오토믹(commands atomic) 및 서버 접속 스테이트리스(sever connections stateless)를 구비하는 것을 포함한다. 지시는 모든 것들이 즉시 이루어질 때 몇 가지 일들을 "오토미컬하게(atomically)" 수행하고, 그 지시가 절반이 완료되거나, 산재되거나, 인터럽트될 가능성은 없다. 스테이트리스 서버는 임의의 이전 요청과는 관련이 없는 독립적인 트랜잭션과 같은 각각의 요청을 처리한다. 이는 서버의 설계가 간단해지도록 하는데, 그 이유는 진행 중인 대화를 처리하기 위해 저장을 할당하거나 클라이언트가 트랜잭션 중간에 없어지는 경우에 저장을 제거하는 것에 대한 걱정을 할 필요가 없기 때문이다. 명령들 및 응답들은 정보 교환을 위한 미국 표준 코드(American standard code for information interchange:ASCII)의 형태일 수 있고, 그것의 포맷은 머신 분석(machine parsing)이 구현되기 쉽도록 선택된다.

[0018] **데이터 관리자 제어 프로토콜(DMCP)**

[0019] 데이터 관리자(102)는, 예컨대, 양방향 접속부(106)를 통해 DMCP 명령들을 전송하고 수신하기 위한 DMCP 프로토콜을 지원할 수 있다. 상기 양방향 접속부(106)는 TCP 또는 사용자 다이어그램 프로토콜(UDP) 접속을 포함할 수 있다. 데이터 관리자(102)는 포트 1880을 통한 DMCP 명령들을 위해 TCP 접속을 사용할 수 있다. 그러나, 만약 데이터 관리자(102)가 DMDP 프로토콜을 지원한다면, 아래에 정의된 바와 같이, 데이터 관리자(102)는 다른 포트들에서 DMCP 프로토콜을 지원할 수도 있다.

[0020] **이진 데이터 교환 포맷(BDEF)**

[0021] 통일된 데이터 교환 및 저장 포맷은 유리하게도 효율적인 구현들, 다른 네트워크 구성성분들에 대한 호환성, 및 코드 재사용 가능성을 유도한다. 그러한 데이터 포맷은 실시간으로 입력될 수 있는 가변 크기를 갖는 큰 볼륨의 데이터에 적합하다. 도 1은 네트워크 접속부(108)를 통해 교환되는 데이터와 저장 유닛(116)에 저장되는 데이터가 예컨대 BDEF와 같은 통일된 포맷을 갖는 예시적인 네트워크를 도시하고 있다. 데이터 클라이언트(104)는 네트워크 접속부(108)를 통해 데이터 관리자(102)로부터 데이터를 수신할 수 있으며, 접속부(118)를 통해 저장 유닛(116)에 데이터를 전송할 수 있다. 일실시예에서는, 네트워크 접속부(108)를 통한 데이터 교환 포맷과 저장 유닛(116)의 데이터 저장 포맷이 통일된다.

[0022] 일실시예에서, 이러한 통일된 포맷은 전체 포맷 및 레코드 포맷을 포함할 수 있다. 전체 포맷은 데이터 저장 포맷을 정할 수 있는 파일 포맷과 네트워크를 통해 전달되는 데이터 교환 포맷을 정할 수 있는 네트워크 포맷으로서 확인될 수 있다.

[0023] 도 2는 일실시예에 따른 통일된 데이터 포맷에 대한 예시적인 구조를 나타낸다. 파일(202)의 전체 포맷은 헤더 섹션(204), 레코드 섹션(206), 및 테일 섹션(208)을 포함할 수 있다. 일실시예에서, 전체 포맷(202)은 레코드 섹션만을 구비할 수 있다. 레코드 섹션(206)내에서 레코드의 레코드 포맷은 레코드 헤더(210), 레코드 보디(212), 및 레코드 테일(214)을 또한 포함할 수 있다.

[0024] 도 3은 BDEF 포맷에 대하여 상세히 설명된 예시적인 전체 포맷을 나타낸다. 행(302 내지 322)은 헤더 섹션(204)에 상응하고, 행(324 내지 332)은 레코드 섹션(206)에 상응하며, 행(334 내지 336)은 테일 섹션(208)에 상응한다. 그러한 행들 중 일부가 아래에서 설명된다:

[0025] OVERALL\_FORMAT\_ID(302)는 기본적인 전체 포맷을 식별한다. 일실시예에서, 기본적인 전체 포맷은 일반적인 이진 레코드 포맷(GBRF)이다. 임의의 파일 뷰어(viewer)나 에디터(editor)가 예컨대 GBRF와 같은 기본적인 전체 포맷을 결정하기 위해서 파일의 첫번째 4비트, 즉 OVERALL\_FORMAT\_ID(302)를 검사하기 위해서 사용될 수 있다. 도 3에 도시된 바와 같이, 만약 이 파일은 GBRF-포맷된 파일이라면, 상응하는 ID 값은 GBRF에 대한 ASCII 문자



들로 설정되어야 한다.

- [0026] OVERALL\_FORMAT\_VERSION(304)은 파일이 기초하는 기본적인 전체 포맷, 예컨대 GBRF의 버전이 무엇인지를 명시한다.
- [0027] RECORD\_FORMAT\_TYPE(306) 및 RECORD\_FORMAT\_VERSION(308)은 BDEF와 같은 레코드 포맷 타입, 및 2와 같은 그것의 버전을 각각 명시한다. "0x0002"에서 앞에 있는 "0x"는 2를 16진수로 나타낸 것이다.
- [0028] OVERALL\_FORMAT\_OPTIONS(310)은 포맷 옵션들을 명시한다. 일실시예에서, 16 비트 중 2 비트는 선택적인 필드를 명시할 수 있고, 다른 비트들은 장래의 포맷 옵션을 위해 예비될 수 있는데, 즉, 0으로 설정될 수 있다.
- [0029] EOR\_SENTINEL(312) 및 EOS\_SENTINEL(314)는 어떠한 센티널(sentinel) 값들이 레코드의 마지막과 스트림의 마지막을 위해서 각각 사용되는지를 명시한다. EOR\_SENTINEL(312)는 레코드의 마지막을 나타내고, EOS\_SENTINEL(314)는 전체 포맷의 마지막을 나타낸다. 이러한 값들은 고정될 수 없고, 따라서 GBRF-포맷된 파일들이 다른 GBRF-포맷된 파일들의 레코드에 쉽게 삽입될 수 있으며, 그로 인해 순환적인 파일 구조를 제공할 수 있다.
- [0030] TOTAL\_RECORDS\_SIZE(316), FIRST\_RECORD\_BYTE(318), 및 LAST\_RECORD\_BYTE(320) 필드들은 레코드들이 바이트순이지 않지만 래핑되는(wrapped) 스트림들을 위해 사용될 수 있다. 레코드 래핑은 레코드들이 고정-크기의 버퍼에 국부적으로 버퍼링되고 버퍼 크기 보다 더 많은 양의 데이터가 있을 수 있는 상황에서 유용하다. 그러므로, 래핑된 레코드들은 단지 FIFO 큐(queue)와 같은 파일에 적합하게 될 가장 최근의 레코드들을 유지한다.
- [0031] END\_OF\_HEADER(322)는 분석기가 헤더 정보를 분석한 이후에 간단한 데이터 무결성 검사를 위해 사용될 수 있다.
- [0032] 스트림의 마지막에 있는 CRC 필드(334)는 애플리케이션-레벨 데이터 무결성 검사(application-level data integrity check)를 제공한다. 만약이 파일이 래핑된다면, CRC는 시간순으로 계산된다. 데이터 분석기들은 그것의 특징을 제공해야만 할 수도 있지만, 애플리케이션 기록기(writer)는 그것을 구현하지 않는 옵션을 가질 수 있다.
- [0033] END\_OF\_STREAM(336)은 또 다른 데이터 무결성 검사를 제공한다. 그것은 비신뢰적인 전송을 검사하고 실시간적인 전송의 끝을 결정하며 GBRF 순환 포맷의 데이터 무결성 검사를 위해 사용될 수 있다.
- [0034] **일예 1.**
- [0035] 다음의 일예는 비옵션을 갖는 헤더를 나타내는데, 이는 예컨대 "47 42 52 46 00 42 44 45 46 00 01 02 00 00 FA CE DE AD CO DE"를 구현하기 위해서 19개의 16진수 바이트를 필요로 한다.
- [0036] 도 5를 참조하면, 구성요소인 바이트들이 다음과 같이 정의된다:
- [0037] 바이트 0 내지 3은 GBRF와 같은 OVERALL\_FORMAT\_ID(302)를 정의한다.
- [0038] 바이트 4는 OVERALL\_FORMAT\_VERSION(304)을 정의한다.
- [0039] 바이트 5 내지 8은 BDEF와 같은 RECORD\_FORMAT\_TYPE(306)을 정의한다.
- [0040] 바이트 9 및 10은 RECORD\_FORMAT\_VERSION(308)을 정의한다.
- [0041] 바이트 11 및 12는 비옵션(no option)과 같은 OVERALL\_FORMAT\_OPTIONS(310)을 정의한다.
- [0042] 바이트 13 및 14와 15 및 16은 EOR\_SENTINEL(312)와 EOS\_SENTINEL(314)를 각각 정의한다.
- [0043] 바이트 17 및 18은 END\_OF\_HEATHER(322)를 정의한다.
- [0044] 동일하지만 가능한 CRC 옵션을 갖는 헤더는 "47 42 52 46 00 42 44 45 46 00 01 02 00 01 FA CE DE AD CO DE"와 같은 19개의 16진수 바이트들을 또한 필요로 하고, 여기서 바이트 11 및 12는 예컨대 CRC가 존재한다는 것을 의미하는 B0=1과 같은 OVERALL\_FORMAT\_OPTION(310)을 정의한다. 비 CRC 옵션을 갖는 트레일러(trailer)는 2개의 16진수 바이트들 "DE AD"를 필요로 할 수 있다.
- [0045] 도 4A 및 4B는 일실시예에 따른 예시적인 BDEF 레코드 포맷을 나타내고 있다. 행(402 내지 434)은 레코드 헤더(210)에 상응하고, 행(436 내지 440)은 레코드 보디(212)에 상응하며, 행(444 내지 446)은 레코드 테일(214)에 상응한다. 이러한 필드 중 일부가 아래에서 설명된다.
- [0046] 구현을 상당히 간단하게 만드는 REC\_SIZE\_SIZE(402)는 REC\_SIZE(404)에 할당되는 비트들의 수를 제공한다. 데

이터 분석기는 REC\_SIZE(404)를 완료하기 위해서 얼마나 많은 바이트들이 필요한가를 결정하기 위해서 신속하게 1 바이트를 판독할 수 있다. 일단 분석기가 전체 REC\_SIZE(404)를 판독하고 그것을 디코딩하면, 상기 분석기는 효율적으로 메모리를 할당한다.

[0047] REC\_SIZE\_SIZE(402)를 사용함으로써 레코드를 생략(skip)하는 것이 용이하다. 레코드들을 기록하는 애플리케이션은 미리 데이터 크기를 알고 있을 수 있고, 따라서 레코드들을 기록하는 것이 또한 간단하게 이루어진다. 그러나, 애플리케이션 기록기는, 상기 레코드 기록기가 32-비트 워드의 상위 2 비트를 하드-코딩하고 심지어 작은 값들을 위해 30-비트 워드 크기를 사용하는 경우에는, 가변적인 REC\_SIZE 데이터를 기록할 때 REC\_SIZE\_SIZE 필드(402)를 걱정할 필요가 없을 수 있다. 그러므로, 본 명세서에 개시된 레코드 포맷은 최적으로 유동적이며, 부수적인 구현 비용을 필요로 하지 않을 수 있다.

[0048] END\_OF\_RECORD(446)가 또한 레코드 기록기가 사용할 수 있는 전체 포맷 헤더 필드 EOR\_SENTINEL(312)(도 3)에서 지정될 수 있다.

[0049] 데이터 관리자(생성기들) 및 데이터 클라이언트들(소비자들) 양쪽 모두는 선형적으로 상이한 레코드 BODY 필드들에 대한 정의를 알 수 있는데, 그 이유는 그것들이 정의될 수 있으며 특정 REC\_TYPE 값에 매핑될 수 있기 때문이다. BODY 정의는 일부 프로토콜에 의해서 또는 저장된 특정 레코드로부터 또한 결정될 수 있다. 그러나, 데이터 소비자에게 명확히 알려지지 않은 RECORD\_TYPE 값은 마치 BODY 정의가 BODY 필드에 이용가능한 많은 비트들을 갖는 정수 워드였던 것처럼, 처리될 수 있다. 알려지지 않은 RECORD\_TYPE 값이 임의적인 크기의 워드들로서 처리됨으로써, 데이터 생성기는 새로운 데이터 타입들을 신속하게 정하여 구현할 수 있으며, 그로 인해 데이터 생성기는 현존하는 데이터 소비자들과 자동적으로 작업한다.

[0050] 일실시예에서, BDEF 스트림들은 제 1 레코드로서 데이터 타입 요약 레코드를 포함할 수 있고, 따라서, 시간에 따라 변할 수 있는 이름들 및 정의들을 알아내기 위해 REC\_TYPE 값들을 분석하는 시간을 절감시킨다.

[0051] **일예 2.**

[0052] 예시적인 실시예에서는, TIME=0x123456789AB, 데이터 소스를 정의하는 CARD\_IP=0xABCDEF01, 데이터 목표를 정의하는 AT\_IP=0x12345678, REC\_TYPE=0x2, 및 문자 0, 1, 2, 및 3으로 구성되는 4-바이트 BODY에 대해서, 레코드 포맷은 "15 02 11 23 45 67 89 AB AB CD EF 01 12 34 56 78 30 31 32 33 FA CE"와 같은 22개의 16진수 바이트들을 필요로 할 수 있다.

[0053] 도 6을 참조하면, 구성요소인 바이트들 및/또는 비트들이 아래에서 설명된다:

[0054] 바이트 0의 첫번째 두 비트들은 예컨대 0과 같은 REC-REC-SIZE(402)를 정의하고,

[0055] 바이트 0의 마지막 여섯 비트들은 21개의 바이트들이 나머지 레코드 포맷을 정의하기 위해 필요하다는 것을 나타내는 예컨대 6과 같은 REC-SIZE(404)를 정의하고,

[0056] 바이트 1의 첫번째 비트는 예컨대 6과 같은 REC-FORMAT(406)을 정의하고,

[0057] 바이트 1의 두번째 비트는 6개의 바이트들이 REC\_TYPE(410)을 정의하기 위해 필요하다는 것을 나타내는 예컨대 0과 같은 REC\_TYPE\_SIZE(408)를 정의하고,

[0058] 바이트 1의 마지막 여섯 비트들은 예컨대 2와 같은 REC-TYPE(410)을 정의하고,

[0059] 바이트 2 내지 7은 DATA\_TIME\_STAMP(414)를 정의하고,

[0060] 바이트 8 내지 11은 데이터 소스 IP 어드레스를 정의하는 DATA\_CARD\_IP(416)을 정의하고,

[0061] 바이트 12 내지 15는 데이터 목표 IP 어드레스를 정의하는 DATA\_AT\_IP(418)을 정의하고,

[0062] 바이트 16 내지 19는 데이터 문자들을 정의하며,

[0063] 바이트 20 및 21은 END\_OF\_RECORD(446)을 정의한다.

[0064] 도 1의 시스템에서 동작하는 셀 전화기나 PDA(personal digital assistant)와 같은 데이터 클라이언트(104), 또는 데이터 관리자(102)가 도 7에 도시되어 있다. 도 7의 시스템은 데이터는 전송 및 수신하기 위한 안테나(702)를 구비한다. 안테나(702)는 송신기 경로로부터 수신기 경로를 분리하기 위한 듀플렉서(704)에 연결된다. 듀플렉서(704)는 수신기 경로를 형성하는 수신기 회로(706)에 연결되며, 송신기 경로를 형성하는 증폭기(708) 및 전송 회로(710)에 연결된다. 증폭기(708)는 상기 증폭기(708)를 제어하는 전력 제어 조정 유닛(712)에 또한

연결된다. 증폭기(708)는 전송 회로(710)로부터 전송 신호들을 수신한다. 안테나(702)를 통해서 수신되는 신호들은 전력 제어 유닛(712)에 제공되는데, 상기 전력 제어 유닛(712)은 페루프 전력 제어 방식을 구현할 수 있다. 전력 제어 유닛(712)은 통신 버스(714)에 연결된다. 통신 버스(714)는 도 7에서 다른 모듈들 간의 공통 접속을 제공한다. 통신 버스(714)는 메모리 유닛(716)에 또한 연결된다. 메모리(716)는 데이터 클라이언트(104)나 데이터 관리자(102)에 적용가능한 여러 동작들 및 기능들을 위한 컴퓨터 판독가능 지령들을 저장한다. 프로세서(718)는 메모리(714)에 저장된 상기 지령들을 수행한다.

[0065] 본 명세서에서 "예시적인"이란 용어는 "일예, 경우, 또는 예시로서 제공하는"을 의미하기 위해서만 사용된다. "예시적인"으로서 본 명세서에 설명된 임의의 실시예는 다른 실시예들에 비해 바람직하거나 유리한 것으로서 해석될 필요는 없다.

[0066] 당업자라면 정보 및 신호들이 임의의 다양한 다른 공학 및 기술을 사용하여 표현될 수 있다는 것을 이해할 수 있을 것이다. 예컨대, 위의 설명 전반에 걸쳐 참조될 수 있는 데이터, 지령들, 명령들, 정보, 신호들, 비트들, 심볼들, 및 칩들은 전압, 전류, 전자기파, 자계 또는 입자, 광자계 또는 입자, 또는 그것들의 임의의 결합으로 표현될 수 있다.

[0067] 당업자라면, 여기서 개시된 실시예와 관련하여 설명되어진 여러 도시적인 논리 블록, 모듈, 회로, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 그것들의 결합으로 구현될 수 있다는 것을 또한 알 것이다. 하드웨어 및 소프트웨어의 그러한 상호변경가능성을 명확히 도시하기 위해서, 여러 도시적인 구성성분, 블록, 모듈, 회로 및 단계가 그것들의 기능을 통해 일반적으로 위에서 설명되었다. 그러한 기능이 하드웨어로 구현되는지 또는 소프트웨어로 구현되는지 여부는 전체적인 시스템에 부가되는 특정 애플리케이션 및 설계의 제한 사항에 의존한다. 숙련된 기술자라면 각각의 특정 애플리케이션에 대해 가변적인 방법으로 상기 설명된 기능을 구현할 수 있지만, 그러한 구현 결정은 본 발명의 범위로부터 벗어나는 것으로서 이해되지 않아야 한다.

[0068] 본 명세서에 개시된 실시예와 관련하여 설명된 여러 예시적인 논리 블록, 모듈, 및 회로는 본 명세서에서 설명된 기능을 수행하도록 설계된 범용 프로세서, DSP(digital signal processor), ASIC(application specific integrated circuit), FPGA(field programmable gate array) 또는 다른 프로그램가능 논리 장치, 이산 게이트 또는 트랜지스터 논리 장치, 이산 하드웨어 성분, 또는 그것들의 임의의 결합으로 구현되거나 수행될 수 있다. 범용 프로세서는 마이크로프로세서일 수 있지만, 대안적으로는, 그 프로세서는 임의의 종래 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수 있다. 프로세서는 또한 컴퓨팅 장치의 결합, 예컨대, DSP 및 마이크로프로세서의 결합, 복수의 마이크로프로세서, DSP 코어와 관련한 하나 이상의 마이크로프로세서, 또는 임의의 다른 그러한 구성으로 구현될 수 있다.

[0069] 여기서 설명된 실시예와 관련하여 설명된 방법이나 알고리즘의 단계는 하드웨어, 프로세서에 의해 실행되는 소프트웨어 모듈, 또는 그것들의 결합으로 직접 구현될 수 있다. 소프트웨어 모듈은 RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터, 하드 디스크, 제거가능 디스크, CD-ROM, 또는 기술 분야에 알려져 있는 임의의 다른 형태의 저장 매체에 상주할 수 있다. 예시적인 저장 매체의 프로세서에 연결됨으로써, 그 프로세서는 저장 매체로부터 정보를 판독하거나 그 저장 매체에 정보를 기록할 수 있다. 대안적으로, 저장 매체는 프로세서에 통합될 수도 있다. 프로세서 및 저장 매체는 ASIC에 상주할 수 있다. ASIC는 사용자 단말에 상주할 수 있다. 대안적으로, 프로세서와 저장 매체는 사용자 단말 내에 개별적인 성분으로서 상주할 수 있다.

[0070] 개시된 실시예의 앞선 설명은 당업자가 본 발명을 제작하고 사용할 수 있게 하기 위해서 제공된다. 그러한 실시예의 여러 변경은 당업자에게는 쉽게 자명해질 것이고, 본 명세서에서 정의된 일반적인 원리는 본 발명의 사상이나 범위로부터 벗어나지 않고 다른 실시예에 적용될 수 있다. 따라서, 본 발명은 본 명세서에 도시된 실시예로 제한되도록 의도되지 않고 본 명세서에 개시된 원리 및 신규한 특징에 부합하는 가장 넓은 범위가 제공될 것이다.

**도면의 간단한 설명**

[0007] 도 1은 데이터 교환 프로토콜들을 구현하기 위한 예시적인 네트워크 인터페이스를 나타내는 도면.

[0008] 도 2는 예시적인 데이터 포맷을 나타내는 도면.

[0009] 도 3은 예시적인 전체 데이터 포맷을 나타내는 도면.

[0010] 도 4A 및 4B는 예시적인 레코드 포맷을 나타내는 도면.

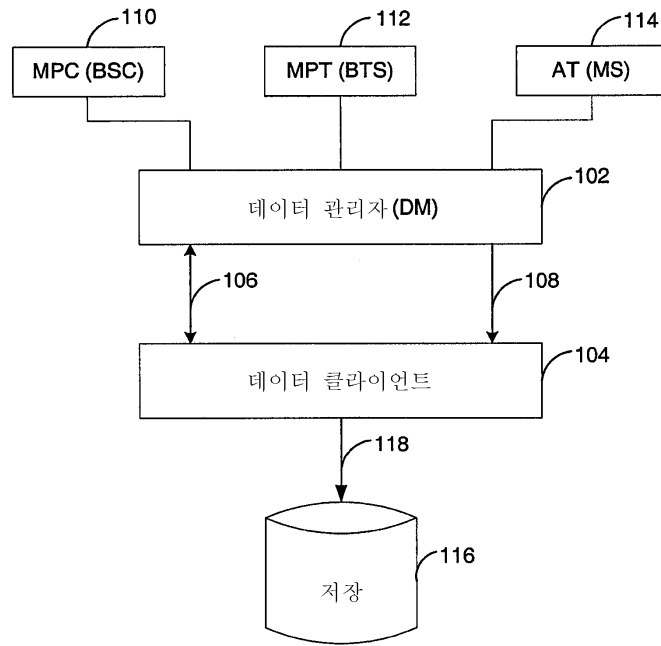
[0011] 도 5는 전체 데이터 포맷에 대한 예시적인 헤더를 나타내는 도면.

[0012] 도 6은 예시적인 레코드 포맷을 나타내는 도면.

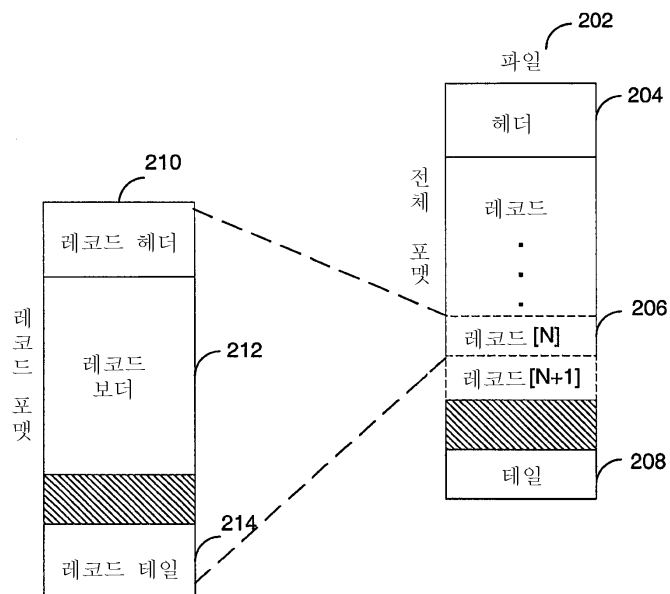
[0013] 도 7은 도 1에서 운용하는 데이터 클라이언트 및 데이터 관리자에 대한 예시적인 실시예의 블록도.

도면

도면1



도면2



도면3

	필드	비트	R/O	설명
302	OVERALL_FORMAT_ID	4 X 8	req	"GBRF"
304	OVERALL_FORMAT_VERSION	8	req	0x00
306	RECORD_FORMAT_TYPE	4 X 8	req	'BDEF'
308	FORMAT_VERSION	16	req	0x0002
310	OVERALL_FORMAT_OPTIONS	16	req	포맷 옵션을 나타냄 (B0는 LSB, 1=TRUE): (B0) CRC 존재 (B1) 레코드 래핑 (B2-B15) 예비
312	EOR_SENTINEL	16	req	0xFACE
314	EOS_SENTINEL	16	req	0xDEAD
316	TOTAL_RECORDS_SIZE	32	opt	만약 레코드 래핑 플래그가 FORMAT으로 설정된다면, 이는 래핑에 의해 야기되는 홀들을 포함하는 레코드 데이터의 총 크기(바이트)이다.
318	FIRST_RECORD_BYTE	32	opt	만약 레코드 래핑 플래그가 FORMAT로 설정된다면, 이는 스트림의 처음부터 제 1 RECORD의 제 1 바이트로의 오프셋이다
320	LAST_RECORD_BYTE	32	opt	만약 레코드 래핑 플래그가 FORMAT로 설정된다면, 이는 스트림의 처음부터 마지막 RECORD의 마지막 바이트로의 오프셋이고, 이는 END_OF_RECORD 센터널이다
322	END_OF_HEADER	16	req	0xCODE
324	RECORD[0]	가변	opt	
326	RECORD[1]	가변	opt	단일 레코드
328	...			
330	RECORD[N-1]	가변	opt	
332	0x00 ... 0x00	가변	opt	래핑에 의해 야기되는 선택적인 데드 공간
334	CRC	16	opt	시간순으로(래핑되는 경우) RECORD[0]-RECORD[N-Q]에 적용되고 0xFFFF가 시드되는(see) 16-비트 CITT CRC.
336	END_OF_STREAM	16	req	0xDEAD

도면4A

	필드	비트	R/O	설명
402	REC_SIZE_SIZE	2	req	REC_SIZE의 크기(즉 0=6, 1=14, 2=22, 3=30).
404	REC_SIZE	6/14/22/30	req	REC_SIZE_SIZE 및 REC_SIZE 자체를 제외한 모든 필요한 선택적인 필드(예컨대, FORMAT, TYPE, CRC 등)을 포함하는 레코드 크기 (바이트).
406	REC_FORMAT	1	req	CRC 존재
408	REC_TYPE_SIZE	1	req	REC_TYPE의 크기(즉 0=6, 1=14).
410	REC_TYPE	6/14	req	0x00-0x3FFE - 도 13 참조 0x3FFF - GBRF에 의해 예비됨
412	DATA_OPTIONS	5	req	포맷 옵션 설명(B0는 LSB, 1=TRUE): (B0) 레코드의 동기적인 어레이 (수 및 주기). B2==0 필요 (B1) DATA_AT_IP 존재 (B2) 레코드의 비동기적인 어레이 (수 및 델타 시간). B0==0 필요 (B3) DATA_PN 존재 (B4) 예비(0으로 설정)
414	DATA_TIME_STAMP	43	req	5/3 milliseconds의 HDR 시간
416	DATA_CARD_IP	32	req	
418	DATA_AT_IP	32	opt	AT 어드레스는 AT 자체를 제외하고 하나의 AT에 특정적인 모든 데이터에 대해 로깅(logged)되어야 하고, AT 자체는 DATA_CARD_IP 필드에 단지 기록될 수 있다
420	DATA_PN	16	opt	PN 어드레스는 한 섹터에 특정적인 모든 데이터에 대해 로깅되어야 한다

도면4B

	필드	비트	R/O	설명
422	DATA_ARRAY_NUMBER	16	opt	이는 ((DATA_OPTIONS.B0 == 1)    (DATA_OPTIONS.B2 == 1)) 하다면 필요하다
424	DATA_ARRAY_PERIOD	32	opt	5/3 밀리초의 HDR 시간. 이는 required if ((DATA_OPTIONS.B0 == 1) && (DATA_OPTIONS.B2 != 1)) 하다면 필요하다
426	DATA_TIME_DELTA_SIZE [0]	2	opt	후속하는 DATA_TIME_DELTA (0=6, 1=14, 2=22, 3=30)의 크기
428	DATA_TIME_DELTA[0]	6/14/22/30	opt	이 값의 DATA_TIME_STAMP 사이의 시간차이
430	...			
432	DATA_TIME_DELTA_SIZE [N-1]	2	opt	후속하는 DATA_TIME_DELTA의 크기 (0=6, 1=14, 2=22, 3=30).
434	DATA_TIME_DELTA[N-1]	6/14/22/30	opt	이 값과 이전 값의 시간 차이
436	DATA_BODY[0]	가변	opt	컨텐츠 레코드
438	...			
440	DATA_BODY[N-1]	가변	opt	
442	PADDING	가변	opt	DATA_BODY가 BITD 값이 경우, 이러한 패딩은 바이트의 끝까지 채울것이다.
444	REC_CRC	16	opt	BODY에 적용되고 0xFFFF가 사용되는 16-비트 CITT
446	END_OF_RECORD	16	req	0xFACE

도면5

비이트 번호	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
웨이트 비이트	47	42	52	46	00	42	44	45	46	00	02	00	00	FA	CE	DE	AD	CO	DE	
설명	G	B	R	F	0	B	D	E	F	0	2	0		FACE	FACE	DEAD	DEAD	CODE	CODE	
도3 참조	302				304	306				308	310	312	314	322						

바이트 번호	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21			
헤더 바이트	15	02	11	23	45	67	89	AB	AB	CD	EF	01	12	34	56	78	30	31	32	33	FA	CE			
설명																						FACE			
도4 참조			414											416					418						444

바이트 번호	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	
설명	0						21										2	
도4 참조	402		404										406	408	410			

도면6



도면7

