

Data Carousel Protocol 을 위한 Descriptor 의 효율적 사용 기법

고광현, 이광기
삼성전자

An URI and Content Type Descriptor Usage for Data Carousel Protocol

Kwang-Hyun Koh, Kwangkee Lee
Samsung Electronics

요약

DASE(DTV Application Software Environment)는 ATSC의 지상파 방송 표준으로 한국에서는 월드컵과 아시안게임을 거쳐 현재까지 시험방송 중에 있다. 국내에서는 주요한 데이터 전송 방식으로 데이터 carousel 프로토콜을 사용하고 있는데, 지금까지는 전송되는 파일의 이름과 content type에 관한 정보가 DST(Data Service Table)에 들어가 있었으나, ATSC 표준이 변경됨에 따라 이 정보가 DII(Download Info Indication)[2]의 descriptor로 위치가 변경되었다. 이로 인해 DII에 삽입해야 하는 데이터의 양이 증가하여 하나의 DII에 삽입 가능한 모듈의 개수가 감소 되었고 전송량이 증가하게 되었는데, 본 논문은 base URI와 default content type을 통해 이러한 제약을 완화할 수 있는 방식에 대해 ATSC S13에 제안한 내용을 토대로 기술하였다.

I. 서론

디지털 TV 방송 기술의 발전과 더불어 보다 많은 부가정보를 TV를 통해 시청자에게 전달하며, 전자상거래(T-Commerce)를 이용할 수 있도록 하는 방식인 데이터 방송에 대한 규격들이 정의되어 구현되고 있다. 현재 데이터 방송을 위한 국제 표준으로 ATSC(Advanced Television Systems Committee)의 DASE, DVB(Digital Video Broadcasting)의 MHP (Multimedia Home

Platform), 그리고 OCAP등이 있는데 국내의 지상파에는 DASE가 월드컵과 아시안게임을 거쳐 현재 시험 방송중에 있다. ATSC는 2002년 9월에 DASE-1을 Standard로 확정하였는데[1], 이와 더불어 ATSC Data Broadcast Standard(A/90)와 ATSC Data Application Reference Model(A/94)에도 수정된 사항이 있어[3]-[6], 국내에서는 이를 토대로 표준에 적합하도록 이전에 구현 하였던 방식을 수정 중에 있다.

본 논문에서는 수정된 내용 중에서 DST에

들어 있던 multipart descriptor, multiple identifier structure, multiple content type structure 등[6]이 없어지고 이 정보가 DSM-CC User-to-Network의 control message인 DII로 옮겨지면서[5] 생기는 문제점과 이에 대한 개선 방안에 대해 기술하였다. 이를 위해 2절에서는 DSM-CC User-to-Network Download 방식인 Data Carousel[2]에 대해 설명하였으며, 3절에서는 ARM(Application Reference Model) binding[5]에 대해 기술하였다. 그리고 4절에서는 표준의 수정 이후 생긴 문제점과 이에 대한 해결방안을 기술하였으며, 이후 5절과 6절에서 실험과 결론으로 마무리하였다.

II. 데이터방송에서의 Data Carousel Protocol

DASE 표준은 ATSC의 다른 표준들과 그림 1과 같은 계층적 관계를 가지고 있다.

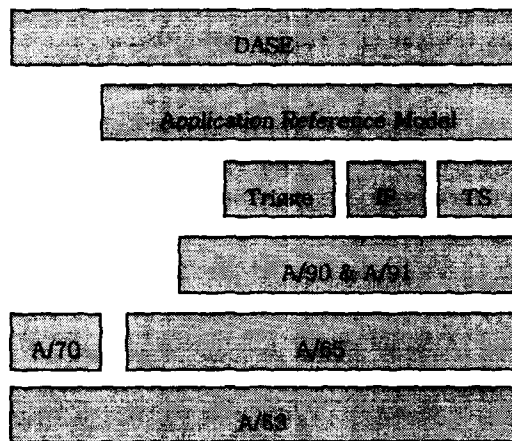


그림 1. ATSC 표준들의 계층구조

DASE 어플리케이션은 그림1의 하위 구조인 데이터 서비스에서 제공되는 여러가지 프로토콜로 전송될 수 있는데, 데이터 서비

스 테이블(DST)의 protocol_encapsulation 필드에 해당 프로토콜이 명시되어진다. 이에 대한 내용은 A/90과 A/91[3][4]에 자세히 기술되어 있다. 이 필드의 값이 0x0D이면 DSM-CC download 프로토콜 [2][3]의 asynchronous carousel 시나리오, 즉 데이터 carousel 방식을 나타내게 된다. 이 방식은 아래 그림2와 같이 한정된 양의 carousel 내 데이터들을 순환적으로 반복해서 전송해 주는 방식이다.

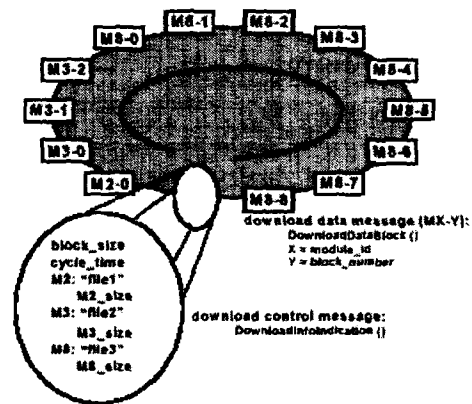


그림 2. 데이터 carousel 전송방식

이 데이터 carousel 프로토콜을 이용해 전달 되는 내용은 Declarative Applications 이거나 Procedural Applications인데, Declarative Applications은 initial entity가 application/xhtml+xml인 markup content type을 가진 경우의 어플리케이션이며, Procedural Applications은 initial entity가 application/java-tv-xlet을 가진 active object content type인 어플리케이션이다. 이 어플리케이션들의 파일들은 각각 하나의 모듈로 구성되며, 이 모듈들을 전송하기 위하여 DDB라는 단위로 모듈을 분리하게 된다.

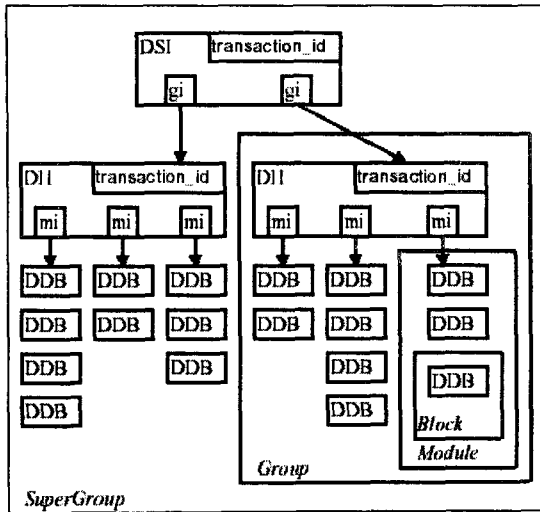


그림 3. 2-layer control 정보의 구조

DSI: DownloadServerInitiate
gi: GroupInfoBytes
DII: DownloadInfoIndication
mi: ModuleInfoBytes
DDB: DownloadDataBlock

데이터 carousel을 통해 DDB를 전송할 때 각각의 모듈들에 대한 정보를 수신단에 제공하기 위하여 DII라는 메시지가 구성되는데 이 메시지에는 각각의 모듈들과 DDB들에 대한 정보가 들어간다. 이는 그림3에서 'Group'이라는 단위로 묶인 형태로 존재하는 경우이며, 이와 같이 1-layer로도 데이터 carousel을 구성할 수 있다. 만약 모듈의 개수가 많아서 하나의 그룹에 들어갈 수가 없거나 관리상의 편의를 제공하기 위하여 여러 개의 그룹을 사용하여야 할 때는 그림3의 'SuperGroup'과 같이 여러 개의 DII 메시지를 두고 모듈들을 나누어서 구성하여 여러 개의 'Group'들을 두게 되며, 각각의 'Group'에 대한 정보를 수신단에 알려주기 위하여 DII들에 대한 정보를 담은 DSI라는 메시지를 구성하게 된다.

III. ARM Binding

ARM(Application Reference Model)은 그림2에서 보는 것과 같이 DASE표준의 하위단에 존재하는 ATSC 표준으로 어플리케이션의 환경에 대한 정보를 수신단에 알려주는 역할을 한다. 여기에는 어플리케이션 모델, identification과 naming, 데이터 encapsulation, 디스크립터, announcement, signaling등에 관한 내용이 들어 있다. 이 중 표준의 변경으로 인해 현재 시험 방송 스트림에서 수정하여야 하는 부분은 디스크립터에 관한 내용이다. 파일(3절에서의 모듈)에 대한 경로와 이름을 나타내기 위하여 identifier와 content type을 수신단에 제공하게 하여야 하는데, 이전의 ARM 표준에서는 이를 multipart descriptor, multiple identifier structure, multiple content type structure를 사용하여 DST에 삽입, 전송하였다. 그러나, 현재 정의된 ARM 표준에서는 DST내에 위에서 언급한 여러 descriptor들에 대한 정의가 사라져 DST를 통하여 이 내용들이 전송할 수 없으며, DII의 그룹에 대한 정보 내에 URI decriptor와 content type descriptor를 삽입하여, 파일의 경로와 이름에 대한 내용, 그리고 콘텐츠 타입에 대한 정보를 전달하도록 수정되었다.

IV. Descriptor 사용 방식

3절에서 기술한 대로 ARM binding 정보의 수정에 의해서 DST의 크기는 줄어든 대신 DII의 크기가 커지게 되었는데 DII는 하나의 section으로 구성되어야 하므로[2], 하

나의 DII에 들어 갈 수 있는 모듈의 개수가 감소하게 되었다. 이러한 제한은 하나의 그룹을 Group Link Descriptor[3][4]를 사용하여 여러 개의 DII로 분리해 구성해야 하는 등 DASE system에서 데이터 carousel 프로토콜을 사용하여 파일을 전송하기에 구성 상의 어려움을 가져올 수 있다.

늘어난 용량에 관한 문제와 디스크립터의 크기제한(253Byte)에 대한 문제, 그리고 하나의 DII 메시지 안에 보다 많은 수의 모듈을 실어 보내기 위한 방식의 해결 방안의 파일의 경로 및 이름에 관한 정보와 content type에 관한 정보를 줄이는 것인데, 이를 위해서 본 실험에서는 DSI 내에 그룹에 관련된 정보를 넣는 groupInfoByte(그룹 4)에 URI 디스크립터와 Content Type 디스크립터를 사용하여 공통된 정보를 담고 DII 메시지 내의 공통된 정보는 제거하는 방식을 사용하였다.

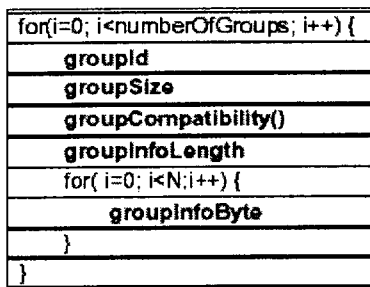


그림 4. Group Indication Information

Group Indication Information(GII) 내의 groupInfoByte에는 각 그룹에 대한 정보가 디스크립터의 형태로 삽입될 수 있는데, 이곳에 URI descriptor와 content type descriptor 등이 들어갈 수 있다. 본 제안에서는 URI descriptor에는 동일한 그룹 내에 있는 모듈들의 공통된 base URI 정보를 담

아서 DII내의 URI descriptor에 불필요하게 반복되는 내용을 넣지 않도록 하였으며, DSI내의 Content type descriptor에는 해당 그룹에 가장 많이 들어 있는 콘텐츠 타입(Default Content Type)을 입력하도록 하였다. 동일한 그룹내에는 이미지 파일 등과 같이 동일한 콘텐츠를 가진 파일들이 많이 있으므로, DII내의 콘텐츠 타입 디스크립터에는 Default Content Type과 동일한 콘텐츠 타입일 경우에는 이를 기술하지 않고, 다른 경우에만 콘텐츠를 기술하도록 하였다.

V. 실험 및 고찰

본 실험에서는 스트림의 수신 및 테스트를 위해서 삼성전자에서 제작한 데이터 방송용 STB(K-150i)를 사용하였으며, 테스트 스트림으로는 아시안 게임 시험방송 중에 KBS와 MBS, SBS가 송출한 시험 방송용 스트림을 사용하였다. 스트림1과 스트림3는 방송 중인 프로그램과 관계가 없는 날씨, 교통, 증권 등을 제공하는 독립정보 서비스이며, 스트림2,4,5은 방송되고 있는 아시안게임과 관련된 경기 내용 및 결과 등 프로그램 관련정보를 제공하기 위한 연동형 서비스이다.

	방송사 및 이름	G	M
스트림1	KBS M13	5	252
스트림2	MBC Asian	1	88
스트림3	SBS v	1	167
스트림4	KBS M2 (폐막식)	1	136
스트림5	KBS M2 (경기)	1	70

표 1. 테스트 스트림

표 1에는 각 테스트 스트림에 관한 내용이 기술되어 있는데 괄호안에 있는 G는 그룹의 개수를, M은 모듈의 개수를 나타내는 것이다.

본 논문에서 제안한 방식의 성능을 테스트 하기 위하여 적용 시 줄어드는 데이터 양과 하나의 DII 안에 들어 갈 수 있는 모듈의 개수를 측정하여 비교하였다.

아래 표1은 각 스트림에 대해서 어플리케이션 전체에 속해 있는 모든 URI 디스크립터와 콘텐츠 타입 디스크립터의 크기를 합산하여 나타낸 표이다.

(단위 : Byte)

	적용전	적용후	적용후/ 적용전
스트림1	15009	4965	0.33
스트림2	5155	2350	0.46
스트림3	7684	4628	0.60
스트림4	5807	1914	0.33
스트림5	3278	1299	0.40

표 2. 전체 디스크립터 용량

적용 전에 비해 적용 후에 용량이 이전 크기의 약 41% 정도로 줄어든 것을 알 수 있다. 스트림3의 경우에는 감소율이 다른 스트림에 비해 적은 것을 볼 수 있는데 이는 스트림3에 있는 리소스의 경로명이 짧아서 공통되는 부분이 적으며, 파일의 이름이 다른 스트림에 비해 조금 더 길었기 때문이다.

아래 표3은 URI 디스크립터와 Content type 디스크립터의 평균 크기를 나타낸 것이다.

(단위 : Byte)

	URI		Content type	
	적용전	적용후	적용전	적용전
스트림1	47.0	18.2	12.8	1.6
스트림2	46.1	24.1	12.5	2.6
스트림3	33.2	23.3	12.7	4.4
스트림4	30.7	11.7	12.0	2.4
스트림5	34.4	15.4	12.4	3.1

표 3. 각 디스크립터의 평균 크기 비교

URI 디스크립터의 경우에는 대략 38 Byte 에서 18 Byte 정도로 약 20 Byte 정도의 크기절감의 효과가 있으며, 콘텐츠 타입 디스크립터의 경우에는 적용 후 대략 3 Byte 정도의 크기로, 효과적으로 그룹을 만들 경우에 거의 DII 메시지의 크기에 거의 영향을 미치지 않음을 알 수 있다.

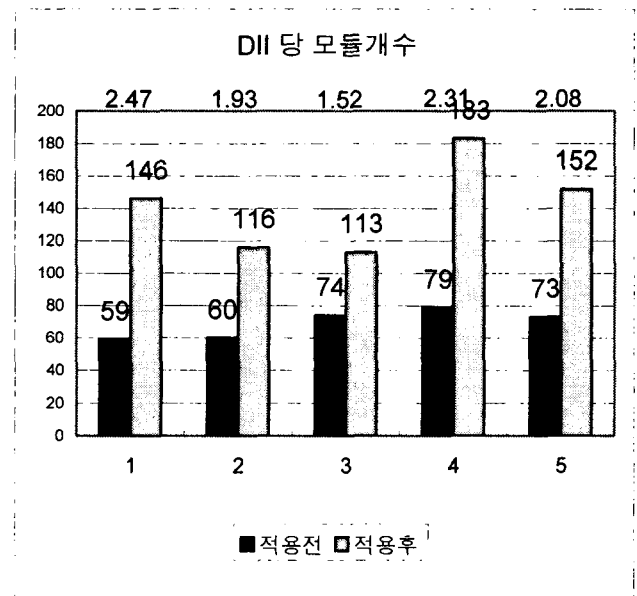


그림 5. DII 당 삽입 가능한 모듈의 개수

그림5는 실험측정치를 이용하여 하나의 DII 메시지에 삽입 가능한 모듈의 최대 개수를 도식화한 것이다. 지금까지 DII 메시지 내에

들어가는 내용의 크기와 표준의 적용으로 인해 들어가야 할 URI, Content type 디스크립터들의 평균 크기를 이용하여, DII 메시지가 하나의 섹션(4096 Byte)으로 구성된다[3] 라는 정의를 만족시키는 하나의 DII 당 모듈의 최대 개수를 구하였다. 그래프의 X축은 스트림을, Y축은 모듈의 개수를 나타내고 있으며, 각 막대 그래프 상단에 표시된 수치는 제안 방식을 적용함으로써 증가되는 모듈의 비(적용후 모듈개수/적용전 모듈개수)를 나타낸 것이다. 적용 이후에 대략 2배의 모듈을 삽입할 수 있음을 알 수 있다. 삽입 가능한 모듈의 개수와 실제 스트림의 그룹당 모듈개수를 살펴보면 본 방식을 적용하기 이전에는 스트림5를 제외한 모든 스트림에서 현재의 구조로 구성이 불가능함을 알 수 있는데, 적용이후에는 스트림3의 경우에만 하나의 DII에 삽입이 불가능하며, 1-layer의 구조를 2-layer로 수정하여 두개의 그룹으로 구성하여야 한다.

VI. 결론

DASE 방식의 데이터방송에 있어 데이터 carousel 프로토콜을 이용하여 데이터를 전송하는 경우, ATSC의 ARM 표준의 변경으로 인하여 리소스에 관한 정보를 저장하는 위치가 변경되었다. 이로 인해 DII 메시지 크기의 증가, 파일 이름의 길이에 대한 제약, 하나의 DII내에 삽입할 수 있는 모듈 개수의 제약 등 데이터 carousel을 통한 전송 방식의 효율성이 저하되었는데, 본 논문에서는 base URI와 Default Content Type을 이용하여 위의 문제들을 개선하였으며, 이로 인해 데이터 carousel 프로토콜이 보다 효율적으로 사용되도록 하였다.

본 논문의 제안 사항을 적용한 이후에도 파일의 이름이 253Byte보다 긴 모듈의 경우 디스크립터 길이 제한이 문제가 발생 할 수 있으며, 그룹이 많아져서 보다 효율적으로 파일 시스템을 구축할 필요가 발생할 수 있다. 이러한 경우에는 TSFS (Transport Stream File System)을 사용하는 것이 해결 방안이 될 수 있다.

참고문헌

- [1] ATSC T3/S17, DASE-1 Proposed Standard - Final, Aug 2002
- [2] ISO/IEC 13818-6, Part6: Extensions for Digital Storage Media Command and Control, July 1998
- [3] ATSC, Doc. A/90, ATSC Data Broadcast Standard, July 2000
- [4] ATSC, Doc. A/91, Implementation Guidelines for the ATSC Data Broadcast Standard (Doc. A/90), June 2001
- [5] ATSC, Doc. T3-558h, ATSC Data Application Reference Model, May 2002
- [6] ATSC, Doc. T3/S17-103, DTV Application Software Environment (DASE) Standard Part x: Application Reference Model, Mar 2001
- [7] 월드컵 데이터방송 기술협의회, kw-kbs-003, 지상파 데이터방송 월드컵서비스 정합 가이드라인, Mar 2002