Mux Log로 알아보는
ATSC 3.0 system(feat.ELK stack)

글.
신 호 SBS 송출기술팀

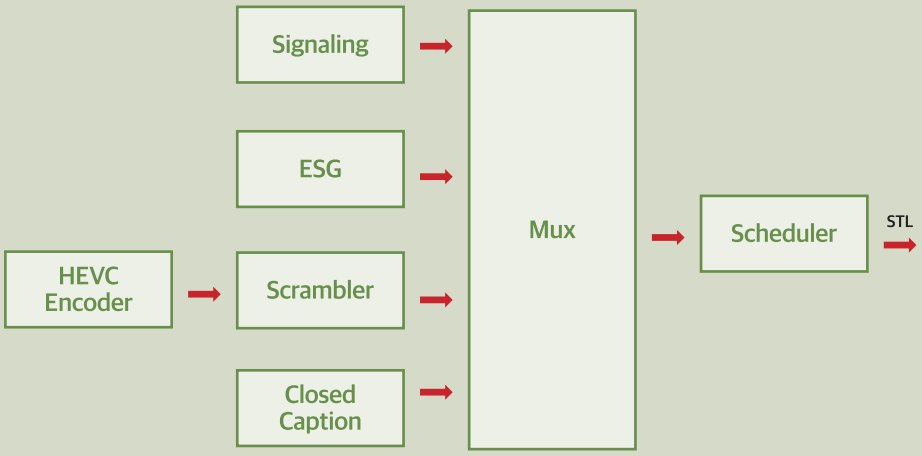
들어가기 전에

2017년 5월 31일을 기점으로 하여 세계 최초로 지상파 UHD 본 방송을 시작으로 지난 10월 UHD&HD 동시 라이브 방송까지 시간이 제법 흘렀다. 그동안 NQC(Network Quality Center, 회선관리실)에서 근무하며 UHD 개국쇼, 국회 인사청문회, 문재인 대통령 취임 100일 기념 기자회견 등 여러 이벤트들을 UHD&HD 동시 생중계로 진행하며 방송 노하우를 쌓고, 곧 있을 평창 동계올림픽 또한 준비하고 있다.

SBS NQC는 송신 관제를 담당하는 TCR(Transmission Control Room)과 함께 통합관제실에 있다. 이곳에서 UHD 중계 회선 관리와 더불어 UHD HeadEnd System도 운용 관리를 담당한다.

이 시스템은 UHD 주조에서 Quad-SDI를 받아 HEVC Encoder부터 Scrambler, Mux와 Scheduler까지 이어지는 구성으로되어 있는데, 필자는 그 중 비디오, 오디오와 폐쇄자막, ESG(Electronic Service Guide), Signaling을 각각 입력받아 처리하는 Mux를 세부적으로 살펴보고자 한다. 이를 위해 Mux의 Log를 살펴보았으며 이를 분석하는데 ELK라는 유용한 툴을 사용했음을 미리 밝힌다. 본 기고를 통하여 ATSC 3.0 시스템과 조금씩 친해지길 기원한다.

ATSC 3.0 시스템을 소개하며



ATSC 3.0 HeadEnd 구성 개념도

SBS는 방송 서비스를 위해 사용하는 MMTP와 ROUTE 프로토콜 중 ROUTE 프로토콜을 사용하여 서비스를 하고 있다. UHD 수상기를 켜면 ROUTE/UDP/IP 패킷 중 LLS(Low Level Signaling) 패킷을 먼저 찾고 그 후에 SLT(Service List Table)의 위치를 찾는다(LLS table_id=0x01). LLS의 IP 주소는 224.0.23.60/4937로 표준에 지정돼 있다. SLT는 지역정보와 주파수 값으로 이루어진 BSID(Broadcast Stream ID)와 각 서비스에 대한 정보를 가지고 있고 이 패킷을 열어야만 ROUTE인지 MMTP인지 파악이 가능하다(@slsprotocol=0x01이 ROUTE, 0x02는 MMT). 여기서 언급하는 서비스는 A/V인 Linear Service와 음성전용 Linear Service, 앱 기반 서비스, ESG 서비스, 비상경보 서비스까지 총 다섯 가지를 의미한다.

SLT 내에 SLS(Service Layer Signaling) 획득 정보를 가지고 SLS 패킷을 보게 되면, USBD(User Service Bundle Description)을 Parsing하고 그 후에 MPD(Media Presentation Description) 그리고 S-TSID 정보들을 Parsing 할 수 있다. USBD와 MPD, S-TSID와 같은 시그널링 정보는 XML 형태로 GZIP으로 압축되어 전달된다. MPD에 기술된 각 컴포넌트들의 세그먼트 URI와 S-TSID에 기술된 정보를 함께 이용하여 컴포넌트 정보를 획득한다. S-TSID는 컴포넌트가 어떤 TSI로 오는지 알 수 있기에 중요한 정보라고 할 수 있는데, RS(Route Session)와 LS(LCT Channel) 두 파트로 나누어져 있다. 위의 Session과 Channel 개념은 RFC 3926 – FLUTE(File Delivery over Unidirectional Transport)에 나오는 개념을 준용한 것이다. RS에는 Source IP와 Destination IP/Port로 이뤄져 있다. 여기서 Destination IP는 Multicast IP다. LS에는 각 컴포넌트들의 TSI가 몇 인지, Bandwidth가 얼마인지, 시작 끝 시간 등의 정보가 들어있다.

설명하다 보니 간단하지 않은 것 같지만, UHD HeadEnd 시스템은 새로운 표준 생성보다는 방송을 IP 패킷으로 전송하기 위해 기존에 있는 표준을 방송에 맞게 재구성한 것에 가깝다고 볼 수 있다. 그렇기 때문에 3GPP나 ISO/IEC 표준을 상황에 맞게 인용하였다. 따라서 여기에 언급되는 용어들은 대부분 위와 같은 표준에서 나온 것들이 대부분이다. TSI는 FLUTE에서, ESG에서 SGDD와 SGDU는 OMA BCAST에서 준용 및 내용을 확장한 것들이다. 국내 지상파 UHD 표준이 800여 페이지에 달하고 그 내용을 이 지면에 전부 담아내기에는 많은 제약이 있어서 이제 Mux가 실제로 데이터를 받아 만드는 로그를 통해 알아보겠다.

ATSC 3.0 Mux Log와 친해지기

Mux가 실제로 생성하는 로그는 다음과 같다. (9월 21일 로그 발췌)

```
[09/21 00:00:09:636] [OK ] [0x0000] [18672] [ROUTE::SEND] SLS Signaling Send...(service_id:601, len:7453)
[09/21 00:00:10:022] [OK ] [0x0000] [18672] [ROUTE::SEND] SLS Signaling Send...(service_id:607, len:2128)
[09/21 15:00:03:233] [OK ] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, tsi:1, toi:1471, len:1049, file_name:SGDD)
[09/21 00:00:09:719] [OK ] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, tsi:2, toi:3442, len:607, file_name:SGDU_Short)
[09/21 00:00:09:748] [OK ] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, tsi:3, toi:2440, len:553, file_name:SGDU_Long)
[09/21 00:00:09:750] [OK ] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, tsi:4, toi:5295, len:27193, file_name:IMG)
[09/21 00:00:10:480] [OK ] [0x0000] [18672] [ROUTE::SEND] Video Segment Send...(service_id:601, tsi:1, toi:1991862, transfer_len:1785844, file_name:aster_stream0_1991862.m4s)
[09/21 00:00:10:019] [OK ] [0x0000] [18672] [ROUTE::SEND] Audio Segment Send...(service_id:601, tsi:3, toi:3634627, transfer_len:7116, file_name:aster_stream2_3634627.m4s)
[09/21 00:00:10:020] [OK ] [0x0000] [18672] [ROUTE::SEND] Audio Segment Send...(service_id:601, tsi:2, toi:3634627, transfer_len:7106, file_name:aster_stream1_3634627.m4s)
```

신 호
SBS 송출기술팀

개발자가 로그를 생성한 것이기에 정해진 형식은 없지만 그 내용 자체는 매우 유의미하다. 그리고 자세히 들여다보면 ATSC 3.0 시스템에 좀 더 가까워질 수 있다고 믿는다. 먼저 로그의 형태를 보면 시간, 상태, 16진수, 숫자, 메시지로 이뤄져 있다. 16진수는 에러코드고 에러발생 시 0x0000이 아닌 값이 찍힌다. 메시지를 자세히 보면 각 로그들이 조금씩 다른 것을 알 수 있다. 주제가 SLS Signaling, Audio, Video, ESG로 이뤄져 있으며, 그 뒤에는 Service ID, tsi, toi, 파일크기, 파일명으로 끝난다. (각 서비스들은 일정한 Duration을 가진 파일을 출력하기 때문.) Service ID는 601과 607 두 가지다. 601은 Linear 서비스들인 Video, Audio, 폐쇄자막과 이들의 시그널링 정보인 SLS로 구성된다. 607은 ESG 정보와 이 정보의 시그널링 정보인 SLS다. 그러면 Service ID가 601인 로그를 더 자세히 보겠다.

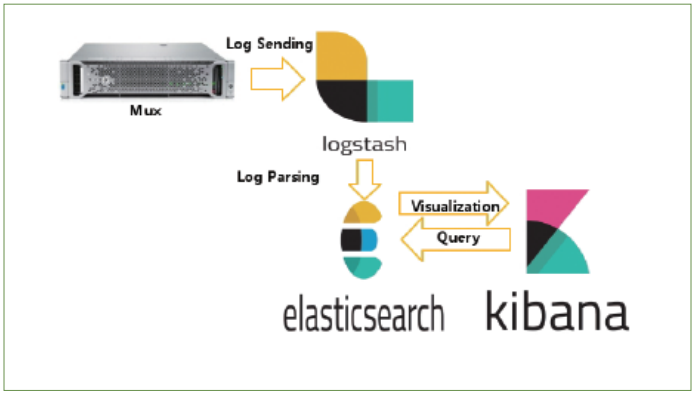
[09/21 00:00:10:480] [OK] [0x0000] [18672] [ROUTE::SEND] Video Segment Send...(service_id:601, **tsi:1,** toi:**1991862,** transfer_len:1785844, file_name:aster_stream0_1991862.**m4s**)
[09/21 00:00:10:019] [OK] [0x0000] [18672] [ROUTE::SEND] Audio Segment Send...(service_id:601, **tsi:3,** toi:**3634627,** transfer_len:7116, file_name:aster_stream2_3634627.**m4s**)
[09/21 00:00:10:020] [OK] [0x0000] [18672] [ROUTE::SEND] Audio Segment Send...(service_id:601, **tsi:2,** toi:**3634627,** transfer_len:7106, file_name:aster_stream1_3634627.**m4s**)

Bold 표시한 부분을 자세히 살펴보면 tsi는 Transport Session Identifier이며 1이면 Video를, 2면 오디오 L채널을, 3이면 오디오 R채널을 의미한다. toi는 Transport Object Identifier이며 표준에서는 동일한 DASH Segment를 전달하는 하나 이상의 ROUTE 패킷들은 동일한 toi값을 헤더에 포함해야 한다고 나와 있다. 위의 오디오의 toi가 같은 이유는 동일한 Segment이기 때문이다. 마지막으로 ROUTE Segment는 ISOBMFF 형식을 따르므로 Encoder의 출력은 파일로 출력되는데 이때 MPEG-4 Segment의 확장자를 .m4s라고 한다. 다음으로 607인 로그를 보겠다.

[09/21 15:00:03:233] [OK] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, **tsi:1,** toi:1471, len:1049, file_name:**SGDD**)
[09/21 00:00:09:719] [OK] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, **tsi:2,** toi:3442, len:607, file_name:**SGDU_Short**)
[09/21 00:00:09:748] [OK] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, **tsi:3,** toi:2440, len:553, file_name:**SGDU_Long**)
[09/21 00:00:09:750] [OK] [0x0000] [18672] [ROUTE::SEND] ESG Data Send...(service_id:607, **tsi:4,** toi:5295, len:27193, file_name:**IMG**)

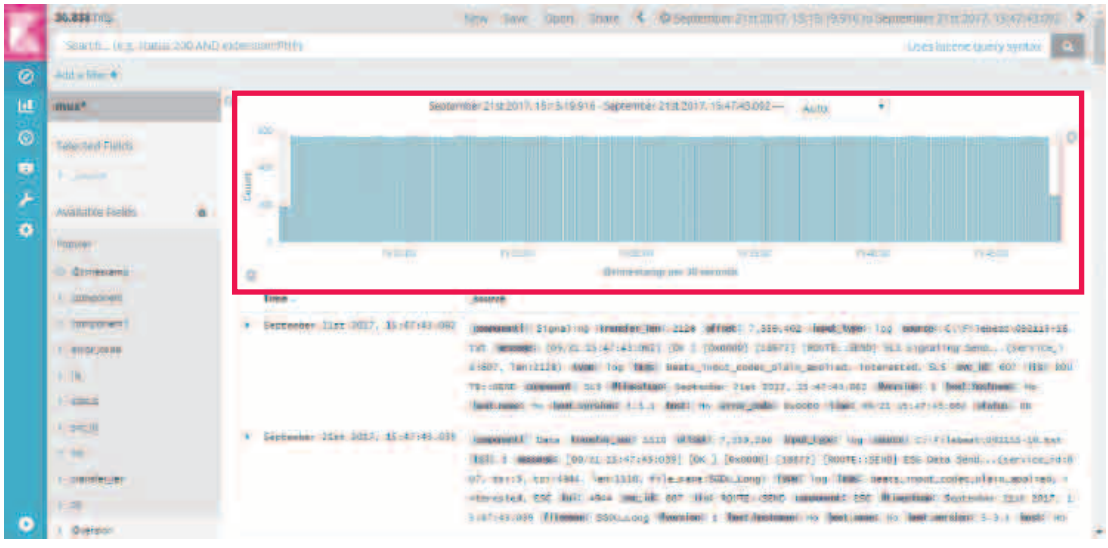
tsi가 1이면 SGDD로서 Descriptor이고, 2면 짧은 기간의 ESG 정보, 3이면 긴 기간의 ESG 정보, 4면 이미지를 뜻한다. SGDD에 기술된 정보를 가지고 ESG 데이터의 전송범위와 SGDU의 위치정보를 제공한다. 이처럼 Mux에서는 Component들이 언제 어떤 파일 이름으로 어떤 크기와 식별자를 가지고 입출력되는지 알 수 있다. 따라서 Mux의 로그들만 가지고 어느 정도의 시스템 모니터링이 가능하며 나아가 다른 서버들이 로그들과 연계한다면, 더욱 세밀한 모니터링이 가능해진다. 이를 토대로 데이터를 유의미하게 정형화하여 ELK에서 분석하고 모니터할 수 있게 해놓았다.

ELK를 통한 ATSC 3.0 Mux Log 분석



ELK(Elasticsearch Logstash Kibana) 구성도

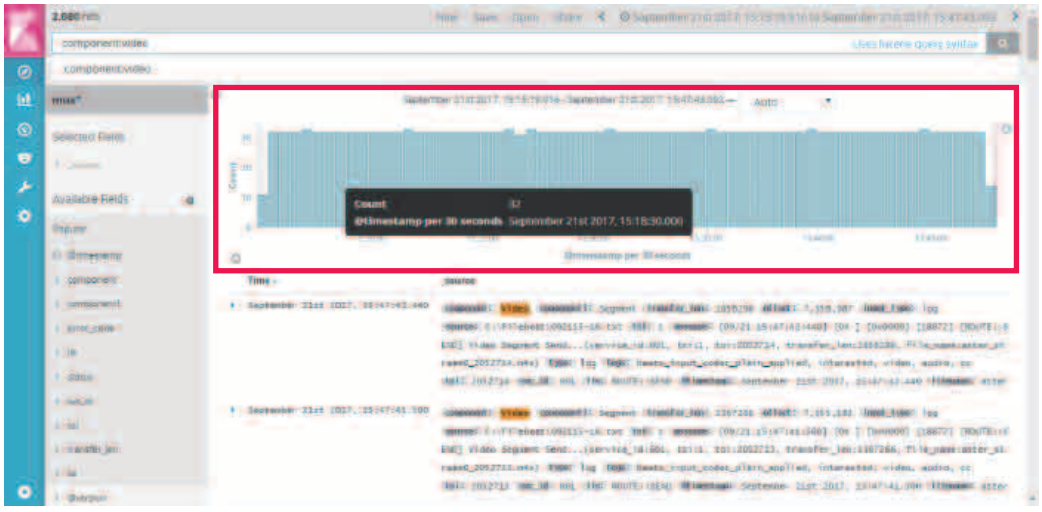
어떤 로그를 통합해서 저장할 수 있고 비정형 데이터를 원하는 데이터로 적절히 다듬어 보기 쉽게 볼 수 있는 오픈 툴 중 하나가 ELK Stack이다. 이 Stack을 구성하는 방법은 다음과 같다. Logstash에서 Raw 데이터인 로그를 정형화하여 Elasticsearch 서버에 데이터를 저장하고 Kibana에서 데이터를 시각화한다. Logstash에서 로그를 정형화할 때 사용자 임의 지정이 가능하며, 그럴 경우에 데이터 타입을 지정해줘야 한다. 따로 지정해주지 않으면 String으로 자동 지정된다. 자료형을 쪼개면 쪼갤수록 Elasticsearch에서 세밀한 검색과 Kibana에서 데이터 분석이 가능해진다. 위의 언급한대로 어떤 컴포넌트가 언제, 어떤 tsi와 toi로 어떤 파일 이름으로 들어오는지 분석하기 위해 미리 데이터 타입을 지정해야 한다는 의미다. Kibana는 GUI 형식으로 Elasticsearch에 검색 쿼리(Lucene Query Syntax)를 날려 그 결과를 화면에 띄우는 것을 가능하게 해준다. 이 ELK를 통해 Mux 데이터의 흐름을 알아보겠다..



Kibana 구성 화면

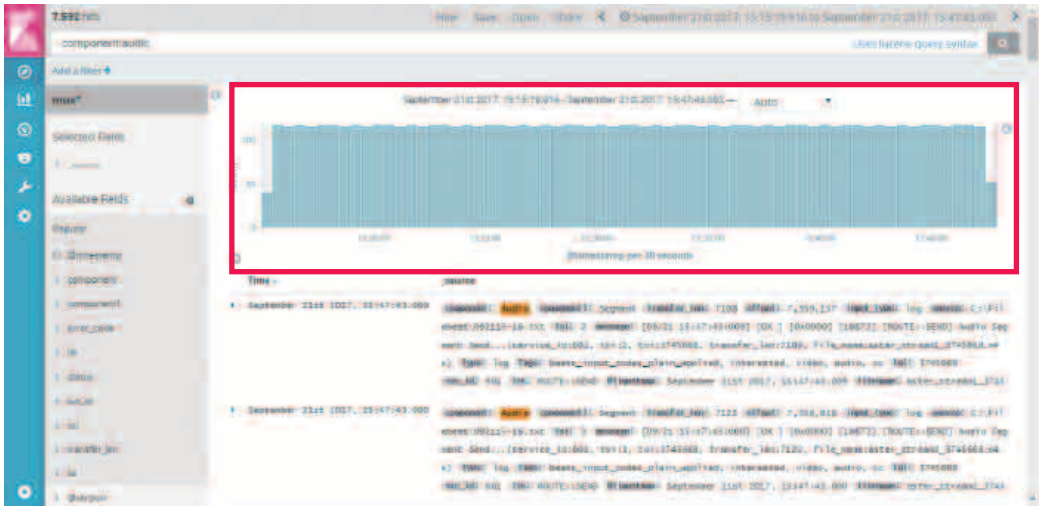
신 호
SBS 송출기술팀

표시 사항을 보면 각 컴포넌트 데이터와 시그널링 데이터가 리니어하게 들어오는 것을 알 수 있다. Mux로부터 입력 받은 Log를 Logstash에서 Parsing 한대로 tsi, component, error_code, service id 등으로 구분 지었다. 이를 통해 알아낸 사실은 다음과 같다.



Kibana Video 검색 결과

먼저 위의 그림을 보면 주황색으로 표현된 부분의 Component가 Video이고, Video의 개수가 60여 개로 일정한 것을 알 수 있다. HEVC Encoder는 1초에 1개씩 비디오 파일을 출력하고 있다.



Kibana Audio 검색 결과

Audio는 1초에 4개 혹은 2개씩 출력을 내고 있다. 마찬가지로 폐쇄 자막은 1초에 1개씩, ESG는 1초에 9개씩 출력을 내고 있다. 이와 같은 사실을 토대로 각 컴포넌트가 정해진 기간 내에 정해진 수량만큼 출력되지 않으면 시스템이 비정상임을 파악할 수 있다. 이런 점을 토대로 시스템 정상 동작을 직관적 파악이 가능하게 Kibana에 시각화했다. 그중 일부를 보겠다.

Kibana Dashboard

입력 데이터를 기반으로 한 분석 Dashboard 중 상단 부분을 보면 어떤 데이터가 몇 개 들어오는지 알 수 있고(좌측) 이를 클라우드로(우측) 표시했다. 입력 데이터 중 ESG가 제일 많고 그다음은 SLS와 Audio가 비슷하고 Video와 폐쇄자막이 그다음으로 개수가 비슷하다. 하단 부분을 보면 좌측에는 Mux의 status를 표시했고 우측에는 Mux가 입력 받은 파일명을 원형 그래프로 표현했다. 향후 각 시스템들의 로그를 분석하여 통합 Dashboard를 꾸민다면 시스템 비정상 동작에 빠른 대처를 할 수 있을 것으로 기대된다.

사족 및 참고사항

UHDTV 3.0은 ROUTE/UDP/IP라고 하였다. 실제 IP 패킷을 분석 목적으로 Wireshark를 통하여 STL 최종 출력을 연결해봤다. 결과를 살펴보면, 패킷당 1414바이트로, 이 중에 헤더가 42바이트이고 나머지 1372바이트가 데이터다. 1초에 약 1821패킷을 보냈기 때문에, 1821×1414×8=20,599,152다. 약 20Mbps로 브로드캐스팅함을 알 수 있다. 헤더에는 Ethernet, UDP, IP 헤더가 붙고 그 뒤에는 페이로드가 붙는다. 페이로드에는 ROUTE 패킷이 붙어 있고 그 내용은 Wireshark에서는 확인할 수 없지만 다들 아실 거라 생각한다.

마지막으로 미처 설명해드리지 못한 부분을 해결할 수 있는 방법을 소개하고자 한다. ATSC 3.0 Candidate Standard(2017년 10월 20일 기준) 중 'A/331: ATSC Proposed Standard: Signaling, Delivery, Synchronization, and Error Protection'에 자세히 나와 있다. 이 표준을 기반으로 국내 표준도 제정되었다. 국내 표준은 TTAK.KO-07.0127/R1 '지상파 UHDTV 방송 송수신 정합'이며 TTA 홈페이지에서 다운 가능하다. 표준 내에 Part 3. 시스템즈를 보면 자세한 내용이 나와 있으니, 표준을 보면서 ISOBMFF에서 Box 구조가 어떻게 되어있는지, Signaling 데이터를 전달하는 XML의 스키마는 어떻게 이뤄져 있는지, FLUTE에서 File Delivery Session은 어떻게 구성되어 있는지 등을 조사하다 보면 어느새 송출시스템 전문가가 되어 있을 것이다.

맺으며

UHD는 앞으로 여러 가지 서비스들이 추가되길 기다리고 있다. HDR(High Dynamic Range), HFR(High Frame Rate), MPEG-H, Advanced ESG 등을 통한 다채로운 시청각 만족 서비스가 그것이다. 지금도 제작에서 송출까지 각 분야에서 연구하는 사람들이 있으며 그런 데이터를 기반으로 시스템 도입 시 UHD HeadEnd 또한 새롭게 꾸며질 것이다. 그에 맞게 모니터링 시스템 또한 확장 구축하며 고품질의 서비스를 시청자께 제공하기 위해 노력하겠다.