

cBR-8、TSDuck、およびVLCを使用したDVB-Cラボ環境の設定

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[設定](#)

[cBR-8ビデオセッション](#)

[ストリーマ](#)

[ECMG](#)

[確認](#)

[cBR-8上](#)

[ECMGで](#)

[トラブルシューティング](#)

[関連情報](#)

概要

このドキュメントでは、TSDuckツールキット、VLC、およびcBR-8を使用して、Digital Video Broadcasting - Cable(DVB-C)ラボシナリオを設定する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- DVB-C
- Symulcrypt
- VoD
- cBR-8

使用するコンポーネント

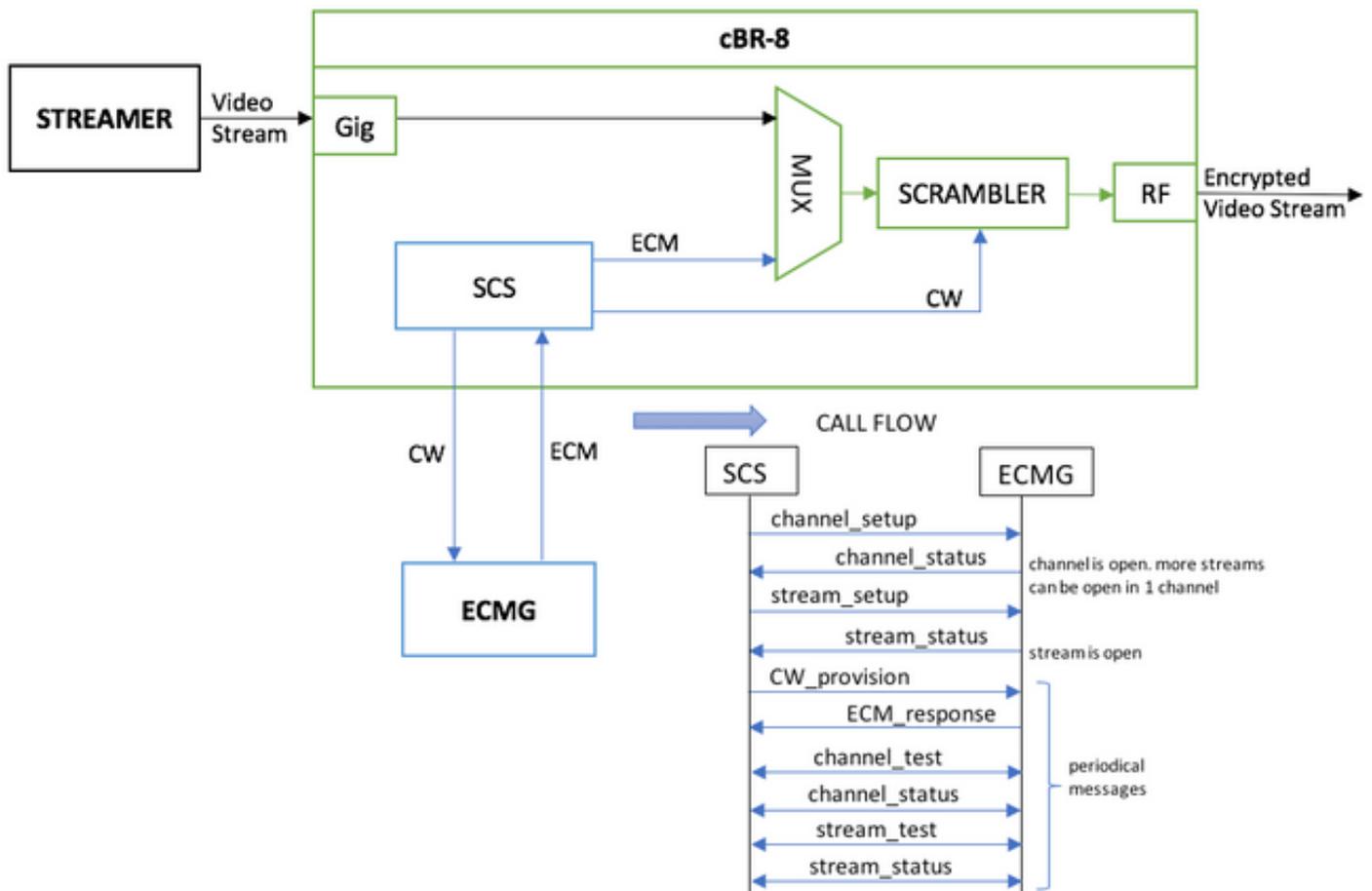
このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期(デフォルト)設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してく

ださい。

背景説明

次の図に示す、このドキュメントで説明するシナリオには、cBR-8 as iCMTS、VLCのビデオストリーマとして使用されるLinux仮想マシン(VM)、およびTSDuckのLinux VMが含まれます。DVB-Symulcrypt暗号化システムが再作成され、cBR8はSimulcrypt Synchronizer(SCS)として機能し、TSDuck VMはNagraサーバと同様にEntitlement Control Message Generator(ECMG)の役割を果たします。



ストリーマとして機能するVMは、ローカルに保存されたビデオクリップを送信するだけで、連続ストリームをシミュレートするためにループします。cBR-8には、このシミュレーション用に1つのテーブルベース(スタティック)セッションが設定されており、VoDストリームを要求するセットトップボックス(STB)やモデムは存在せず、ストリーマで手動で開始されます。

ストリームを受信すると、cBR-8はビデオストリームを暗号化するために、設定されたECMGサーバとの通信を試み、上の図のコールフローに示されているメッセージを交換します。これらのメッセージはTSDuckと明確に交換されるため、メッセージとデバッグの内容を分析するのに役立ちます。また、TSDuckは、送信されたすべての要求に応答します。パラメータの正確性をca-system-id、access-criteriaなどにチェックしません。

cBR-8がECMGとの通信に失敗した場合、命令のフェールツークリアが原因で、ストリームがクリアテキストで送信されます。

実際のシナリオでは、受信者に特定のコントロールワード(CW)の復号化を許可する権限付与管理メッセージ(EMM)をSTBに送信する必要があります。EMMは、cBR-8を介して、またはレシーバに別のチャンネルで送信できます。また、TSDuckには、EMM Generator(EMMG)をシミュレートす

る機能もあります

設定

cBR-8ビデオセッション

cBR-8でDVBビデオセッションを設定する方法の例を次に示します。通常、アクセス基準はConditional Access System(CAS)によって提供されます。このシミュレーションケースでは、ca-system-idだけでなく、ランダムな16進数を生成できます。

virtual-edge-input-ipは、ストリームのIP宛先です。この場合、実際の宛先ではありませんが、ストリームからビデオストリームを送信するために使用されるIPと同じである必要があります。

```
cable video
  encryption
    linecard 1/0 ca-system dvb scrambler dvb-csa
  dvb
    ecmg NAGRA_ELK id 1
      mode tier-based
      type nagra
      ca-system-id 2775 3
      auto-channel-id
      ecm-pid-source auto 48 8190
      connection id 1 priority 1 10.48.88.12 3337
      overrule
        min-cp-duration 300000
    tier-based
      ecmg name NAGRA_ELK access-criteria c972bfd7701e6d28069ae85f5d701d63ac1aec4a
      fail-to-clear
      enable
  service-distribution-group SDG-ACDC-LAB-TEST1 id 1
    onid 100
    rf-port integrated-cable 1/0/3
  virtual-carrier-group VCG-ACDC-LAB-TEST1 id 1
    encrypt
    service-type narrowcast
    rf-channel 32-35 tsid 42496-42499 output-port-number 1-4
  bind-vcg
    vcg VCG-ACDC-LAB-TEST1 sdg SDG-ACDC-LAB-TEST1
  logical-edge-device LED-ACDC-LAB-TEST1 id 1
    protocol table-based
      virtual-edge-input-ip 10.10.10.10 input-port-number 1
      vcg VCG-ACDC-LAB-TEST1
      active
    table-based
      vcg VCG-ACDC-LAB-TEST1
      rf-channel 32
        session vod1 input-port 1 start-udp-port 65 num-sessions-per-qam 1 processing-type remap
  start-program 1
!
controller Integrated-Cable 1/0/3
  max-carrier 44
  base-channel-power 40
  rf-chan 32 35
  type VIDEO
  frequency 850000000
  rf-output NORMAL
  power-adjust 0.0
```

ストリーマ

このデバイスでは、コマンドラインからVLCをインストールするだけで、ローカルに保存されたビデオファイルのストリームを開始できます。

公式ドキュメントを参照してください。

VLCをインストールした後、次のコマンドラインは、cisco-tac-lab.movという名前のファイルのストリームを開始し、宛先IPとポート、cBR-8のtsidとポートを指定し、ビデオをループさせて連続的なフロー(—repeat)をシミュレートします。

```
cvlc cisco-tac-lab.mov —sout
'#duplicate{dst=udp{mux=ts,dst=10.10.10.10:65,tsid=42496,port=65}}' —repeat &
```

ECMG

公式WebサイトからTSDuckをダウンロードします。[TSDuck](#)をインストールして、機能の情報を検索するには、ユーザガイドのドキュメントを参照してください。

TSDuckがインストールされている場合、冗長オプション(-v)と必要なレベルのデバッグ(-d#)を使用して、特定のポート(-p)でECMG機能を実行できます。

例：

```
sudo tsecmg -p 3337 -v -d7
```

確認

cBR-8上

cBR-8でビデオセッションを設定した後は、セッションが作成されたことを確認できます。これは、セッションが常に存在するテーブルベースの設定であり、入力ストリームが表示されないためです。

```
acdc-cbr8-2#show cable video session all
```

Session	Output	Frequency	Streaming	Sess	Session	Source	UDP	Output		
Input	Output	Input	Output	Encrypt	Encrypt	Low PMV	Session			
Id	Port	Hz	Type	Type	Ucast	Dest IP/Mcast	IP (S,G)	Port	Program	
State	State	Bitrate	Bitrate	Type	Status	Lat	NUM	Name		
1048576	1	850000000	Remap	UDP	10.10.10.10			65	1	OFF
ON	0	0	DVB	Pending	N	-	vod1.1.0.1.32.65			

ビデオストリームを開始すると、ECMGがまだアップしていない場合は、cBR-8のfail-to-clear命令に従って、ストリームがクリアで送信されていることがわかります。

```
acdc-cbr8-2#show cable video sess logical-edge-device id 1
```

Session	Output	Frequency	Streaming	Sess	Session	Source	UDP	Output
---------	--------	-----------	-----------	------	---------	--------	-----	--------

Input Id	Output Port	Input Hz	Output Type	Encrypt Type	Encrypt Status	Low PMV IP/Mcast	Session IP (S,G)	Port	Program
1048576	1	850000000	Remap	UDP	10.10.10.10			65	1
ACTIVE-PSI ON		15403951	15164562 DVB	Clear		N -	vod1.1.0.1.32.65		

ECMGも起動すると、ビデオセッションが暗号化されていることがわかります。

```
acdc-cbr8-2#sh cable video sess logical-edge-device id 1
```

Session Id	Output Port	Frequency Hz	Streaming Type	Sess Encrypt Type	Session Encrypt Status	Source IP/Mcast	Low PMV IP (S,G)	UDP Port	Output Program
1048576	1	850000000	Remap	UDP	10.10.10.10			65	1
ACTIVE-PSI ON		15353613	15476997 DVB	Encrypted		N -	vod1.1.0.1.32.65		

暗号化セッションの詳細：

```
acdc-cbr8-2#sh cable video sess logical-edge-device id 1 session-id 1048576
```

```
Session Name      : vod1.1.0.1.32.65
Session Id       : 1048576
Creation Time    : Thu Dec 6 14:12:54 2018
```

```
Output Port      : 1
TSID             : 42496
ONID             : 100
Number of Sources : 1
  Destination IP : 10.10.10.10
  UDP Port       : 65
Config Bitrate   : not specified
Jitter           : 100 ms
Processing Type   : Remap
Stream Rate      : VBR
Program Number   : 1
Idle Timeout     : 2000 msec
Init Timeout     : 2000 msec
Off Timeout      : 60 sec
Encryption Type  : DVB
Encryption Status : Encrypted
```

Input Session Stats:

```
=====  
State: ACTIVE-PSI, Uptime: 0 days 00:31:33  
IP Packets: In 899927, RTP 0, Drop 0  
TP Packets: In 6299489, PCR 6408, PSI 4424, Null 0  
             Unreference 2212, Discontinuity 0  
Errors: Sync loss 0, CC error 795, PCR Jump 7,  
             Underflow 215, Overflow 4, Block 0  
Bitrate: Measured 16483732 bps, PCR 17930489 bps
```

Output Session Stats:

```
=====  
State: ON, Uptime: 0 days 00:31:33  
TP Packets: In 6297330, PCR 6395, PSI 4416,  
             Drop 12801, Forward 6280113, Insert 6029  
Errors: Info Overrun 0, Info Error 0, Block 0, Overdue 54210,  
             Invalid Rate 0, Underflow 0, Overflow 0
```

Bitrate: Measured 16433824 bps

PAT Info:

=====

Version 26, TSID 8724, len 16, section 0/0
Program 1: PMT 32

Input PMT Info:

=====

Program 1, Version 28, PCR 100, Info len 0
PID 100: Type 27, Info len 6, (lang eng)

Output PMT Info:

=====

Program 1, Version 5, PCR 49, Info len 6, (CA SYS-ID 10101, PID 79)
PID 49: Type 27, Info len 6, (lang eng)

Output PID Map:

=====

PID 32 -> 48
PID 100 -> 49

ECMG接続ステータスを表示するコマンド :

```
acdc-cbr8-2#show cable video encryption dvb ecmg id 1 connection
```

```
-----  
-----  
ECMG ECMG          ECMG   CA Sys   CA Subsys  PID      Lower  Upper  Streams/  Open  
Streams/  Auto Chan Slot  ECMG      ECMG  
ID  Name          Type    ID        ID        Source  limit  limit  ECMG      ECMG  
ID                Connections Application  
-----  
-----  
1    NAGRA_ELK          nagra   0x2775   0x3       auto    48     8190   1         1  
Enabled  RP    1          Tier-Based
```

ECMG Connections for ECMG ID = 1

```
-----  
-----  
Conn Conn    IP          Port   Channel Conn    Open  
-ID  Priority Address      Number ID     Status  Streams  
-----  
-----  
1    1         10.48.88.12  3337  1     Open    1  
-----  
-----
```

注 : cBR-8でECMを受信すると、ECMはキャッシュに保存され、ECMGとの接続が失われた場合、キャッシュされたECMは新しいECMを受信するまで暗号化に使用されます。

ECMGで

デバッグが有効になっているので、ECMGとSCSの間で交換されたすべてのメッセージを確認できます (最初の図に示すコールフローを参照)。

```
cisco@simulcrypt:~$ sudo tsecmg -p 3337 -v -d7  
debug level set to 7  
* Debug: setting socket reuse address to 1  
* Debug: binding socket to 0.0.0.0:3337  
* Debug: server listen, backlog is 5
```

```
* TCP server listening on 0.0.0.0:3337, using ECMG <=> SCS protocol version 2
* Debug: server accepting clients
* Debug: received connection from 88.88.88.89:56102
* Debug: server accepting clients
* 88.88.88.89:56102: 2018/12/06 14:38:35: session started
* Debug: received message from 88.88.88.89:56102
    channel_setup (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0001
    ECM_channel_id = 0x0001
    Super_CAS_id = 0x27750003

* Debug: sending message to 88.88.88.89:56102
    channel_status (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0003
    ECM_channel_id = 0x0001
    section_TSpkt_flag = 1
    AC_delay_start = 200
    AC_delay_stop = 200
    delay_start = 200
    delay_stop = 200
    transition_delay_start = -500
    transition_delay_stop = 0
    ECM_rep_period = 100
    max_streams = 0
    min_CP_duration = 10
    lead_CW = 1
    CW_per_msg = 2
    max_comp_time = 100

* Debug: received message from 88.88.88.89:56102
    stream_setup (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0101
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    ECM_id = 0x0001
    nominal_CP_duration = 100

* Debug: sending message to 88.88.88.89:56102
    stream_status (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0103
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    ECM_id = 0x0001
    access_criteria_transfer_mode = 0

* Debug: received message from 88.88.88.89:56102
    CW_provision (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0201
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    CP_number = 0
    access_criteria (20 bytes) =
        C9 72 BF D7 70 1E 6D 28 06 9A E8 5F 5D 70 1D 63 AC 1A EC 4A
    CP = 0
    CW (8 bytes) = 4E 0A 45 9D DC 10 4A 36
    CP = 1
    CW (8 bytes) = AB FF 00 AA 9C 4F 11 FC

* Debug: sending message to 88.88.88.89:56102
```

```
ECM_response (ECMG<=>SCS)
protocol_version = 0x02
message_type = 0x0202
ECM_channel_id = 0x0001
ECM_stream_id = 0x0001
CP_number = 0
ECM_datagram (188 bytes) =
  47 5F FF 10 00 80 70 35 80 AA 03 00 30 00 10 00 08 4E 0A 45 9D DC
  10 4A 36 00 11 00 08 AB FF 00 AA 9C 4F 11 FC 00 12 00 14 C9 72 BF
  D7 70 1E 6D 28 06 9A E8 5F 5D 70 1D 63 AC 1A EC 4A FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

* Debug: received message from 88.88.88.89:56102

```
channel_test (ECMG<=>SCS)
protocol_version = 0x02
message_type = 0x0002
ECM_channel_id = 0x0001
```

* Debug: sending message to 88.88.88.89:56102

```
channel_status (ECMG<=>SCS)
protocol_version = 0x02
message_type = 0x0003
ECM_channel_id = 0x0001
section_TSpkt_flag = 1
AC_delay_start = 200
AC_delay_stop = 200
delay_start = 200
delay_stop = 200
transition_delay_start = -500
transition_delay_stop = 0
ECM_rep_period = 100
max_streams = 0
min_CP_duration = 10
lead_CW = 1
CW_per_msg = 2
max_comp_time = 100
```

* Debug: received message from 88.88.88.89:56102

```
stream_test (ECMG<=>SCS)
protocol_version = 0x02
message_type = 0x0102
ECM_channel_id = 0x0001
ECM_stream_id = 0x0001
```

* Debug: sending message to 88.88.88.89:56102

```
stream_status (ECMG<=>SCS)
protocol_version = 0x02
message_type = 0x0103
ECM_channel_id = 0x0001
ECM_stream_id = 0x0001
ECM_id = 0x0001
access_criteria_transfer_mode = 0
```

トラブルシューティング

cBR-8では、対応するスーパーバイザプラットフォームのトレースをデバッグまたはノイズレベルに設定して、暗号化の問題をトラブルシューティングできます (最後に通知レベルを復元する

ことを忘れないでください)。

```
set platform software trace sup-veman rp active scs debug
```

cBR-8とECMG間のメッセージの正しい交換は次のようになります。

```
show platform software trace message sup-veman rp active reverse
```

```
12/07 15:34:43.963 [scs]: [47872]: (debug): ECMG Send channel_setup for channel_id 1
12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Received channel_status for channel_id 1
12/07 15:34:43.965 [scs]: [47872]: (info): ECMG Channel 0 setup to ip 10.48.88.12 port 3337
12/07 15:34:43.965 [scs]: [47872]: (debug): Open stream 1
12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Send stream_setup for channel_id 1, stream_id 1
12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Received stream_status for channel_id 1,
stream_id 1
12/07 15:34:43.965 [scs]: [47872]: (info): ECMG Stream 1 setup to ip 10.48.88.12 port 3337
12/07 15:34:43.965 [scs]: [47872]: (debug): Request ECM for CP 0
12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Send CW_provision with 20 AC bytes for
channel_id 1, stream_id 1
12/07 15:34:43.966 [scs]: [47872]: (debug): Received ECM_response for channel_id 1, stream_id 1
12/07 15:34:43.966 [scs]: [47872]: (debug): ECMGp: Forward ECM pkts to SCS
12/07 15:34:43.966 [scs]: [47872]: (debug): Received ECM for CP 0
12/07 15:34:56.015 [scs]: [47872]: (debug): ECMG Send channel_test for channel_id 1
12/07 15:34:56.016 [scs]: [47872]: (debug): ECMG Received channel_status for channel_id 1
12/07 15:35:18.039 [scs]: [47872]: (debug): ECMG Send stream_test for channel_id 1, stream_id 1
12/07 15:35:18.042 [scs]: [47872]: (debug): ECMG Received stream_status for channel_id 1,
stream_id 1
```

関連情報

- DVB Simulcrypt技術仕様 (この記事の作成時点で最新) : [ETSI TS 103 197 V1.5.1\(2008-10\)](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)