Satellite Based IP Content Delivery Network

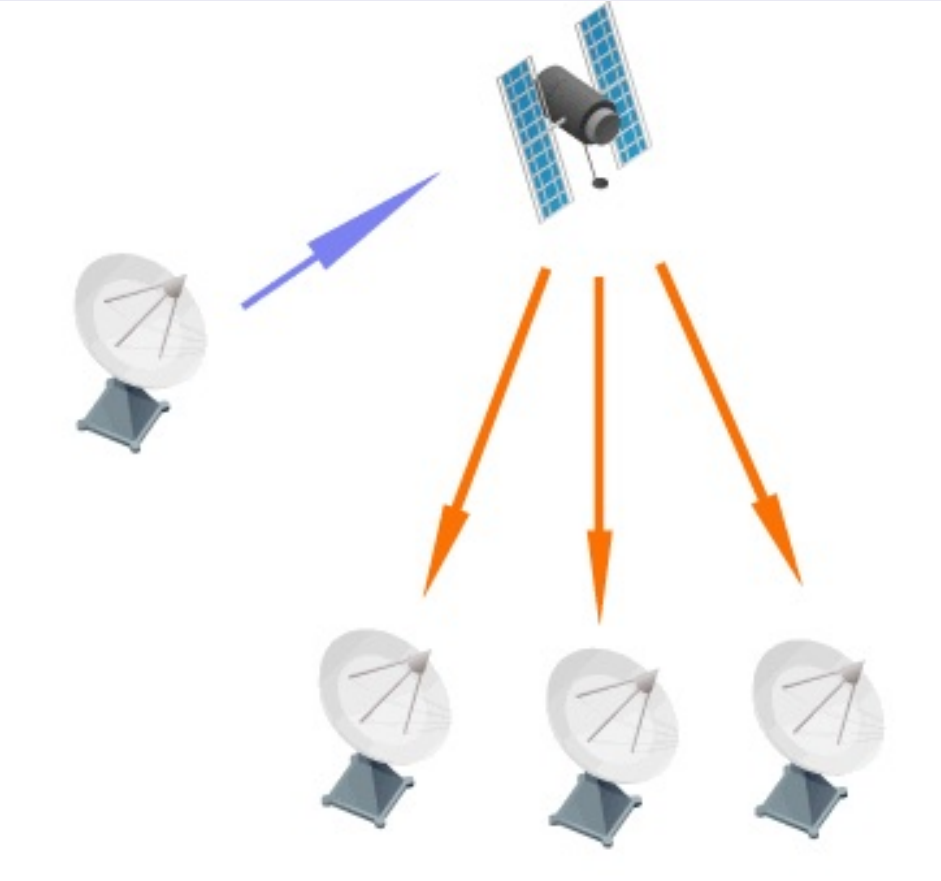
Taylor Jacob

ReCon Brussels 2017

27 Jan 2017

Receive Only Satellite Network

Network Diagram



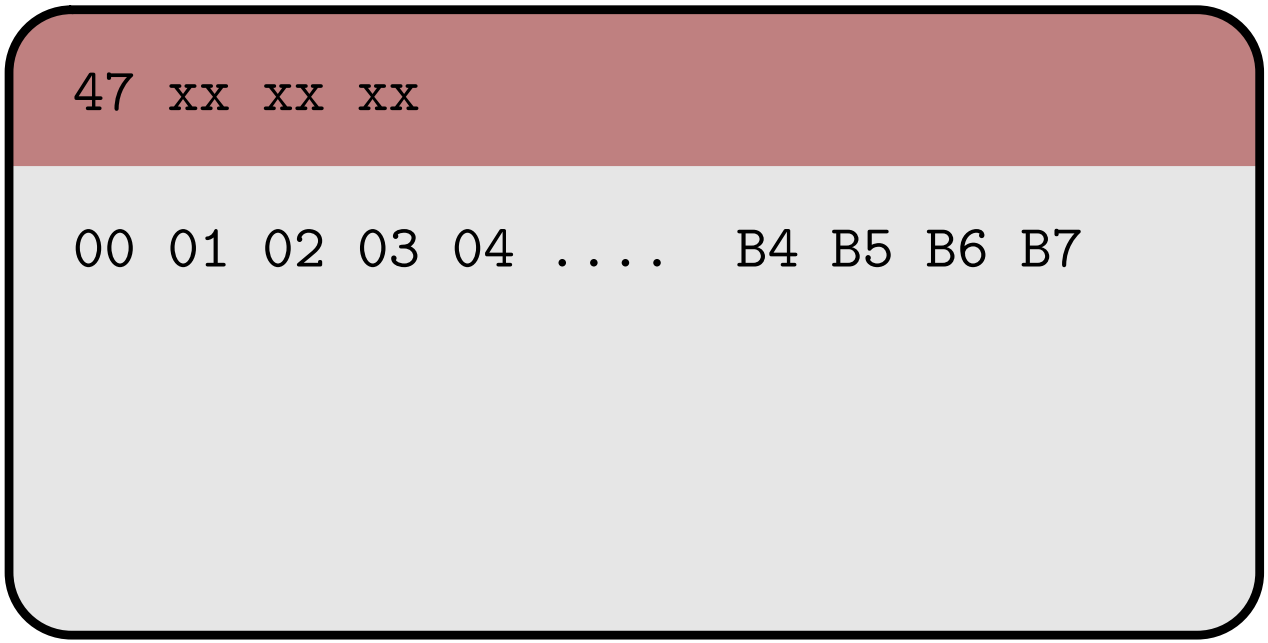
Hardware Setup

C/Ku Band Dishes and DVB-S2 Tuner Card

Software Setup

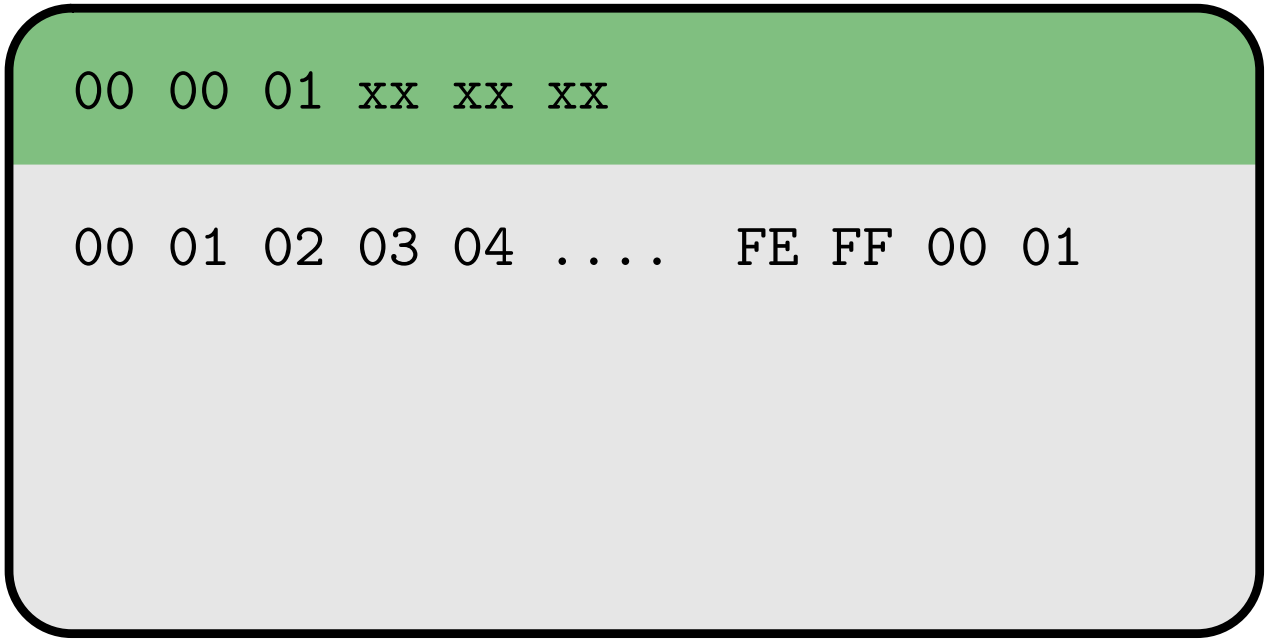
- **dvbsnoop** - transport stream analyzing tool
- **linux dvbtools** - szap-s2, dvbtraffic
- **standard unix tools** - grep, sort, uniq, etc
- **custom software** - gnu C v4l/linux dvb-api

MPEG Transport Stream



- Fixed Length 188 bytes
- Header Fixed Length 4 bytes
- Header always starts with 47h
- Body contains PES or PSI

Packetized Elementary Stream (PES) Format



- Variable Length (Header and Body)
- Header always starts with 00 00 01
- ES contents generally Audio or Video

Signal Analysis

Blind scanning a DVB Signal

Manual PID Identification - Step 1

"Empty" Mux

```
$ dvbtraffic
0000     15 p/s      2 kb/s     23 kbit
0011     15 p/s      2 kb/s     23 kbit
0320  47535 p/s   8727 kb/s  71493 kbit
1fff    512 p/s     94 kb/s    770 kbit
2000  48079 p/s   8827 kb/s  72312 kbit
-PID--FREQ-----BANDWIDTH-BANDWIDTH-
```

Standard Television Mux

```
$ dvbtraffic
0000      2 p/s      0 kb/s      4 kbit
0001      9 p/s      1 kb/s     14 kbit
0011      0 p/s      0 kb/s      1 kbit
0029      2 p/s      0 kb/s      4 kbit
0032      2 p/s      0 kb/s      4 kbit
0065   1158 p/s    212 kb/s   1742 kbit
0066    777 p/s    142 kb/s   1168 kbit
0067    793 p/s    145 kb/s   1193 kbit
<snip>
1622   5866 p/s   1076 kb/s   8823 kbit
1623    306 p/s     56 kb/s    461 kbit
1625      2 p/s      0 kb/s      4 kbit
1c51    135 p/s     24 kb/s    203 kbit
1ffe     51 p/s      9 kb/s     77 kbit
1fff   3327 p/s    610 kb/s   5004 kbit
2000  18429 p/s   3383 kb/s  27718 kbit
-PID--FREQ-----BANDWIDTH-BANDWIDTH-
```

Manual PID Identification - Step 2

dvbsnoop output

```
$ dvbsnoop -n 1 0x320
dvbsnoop V1.4.50 -- http://dvbsnoop.sourceforge.net/

------------------------------------------------------------
SECT-Packet: 00000001   PID: 800 (0x0320), Length: 1360 (0x0550)
Time received: Tue 2017-01-17  21:22:33.535
------------------------------------------------------------
  0000:  3e 75 4d e0 e0 c1 00 00  60 5e 00 01 45 00 05 40   >uM.....‘^..E..@
<snip>
  0540:  04 40 da d2 07 25 73 b9  26 60 d2 ee 00 00 00 00   .@...%s.&‘......

PID:  800 (0x0320)

Guess table from table id...
DSM-CC DATAGRAM-decoding....
Table_ID: 62 (0x3e)  [= DSM-CC - private data section  // DVB datagram]
<snip>
IP_datagram_bytes:
<snip>
    Destination address: e0e0e0e0 [= 224.224.224.224]

    UDP_datagram_bytes:
        Source port: 63889 (0xf991)
        Destination port: 8001 (0x1f41)
        Length: 1324 (0x052c)
        Checksum: 55994 (0xdaba)
        Data
              0000:  00 01 24 05 91 47 a6 fb  47 7e 18 00 00 00 00 00   ..$..G..G~......
<snip>
              0520:  26 60 d2 ee                                       &‘..

Checksum: 0 (0x00000000)
============================================================
```

Multicast IP Traffic Examination using dvbsnoop

dvbtraffic output piped into grep

```
$ dvbsnoop 0x320 | grep "^    Destination address"
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
    Destination address: e0e0e0e0 [= 224.224.224.224]
```

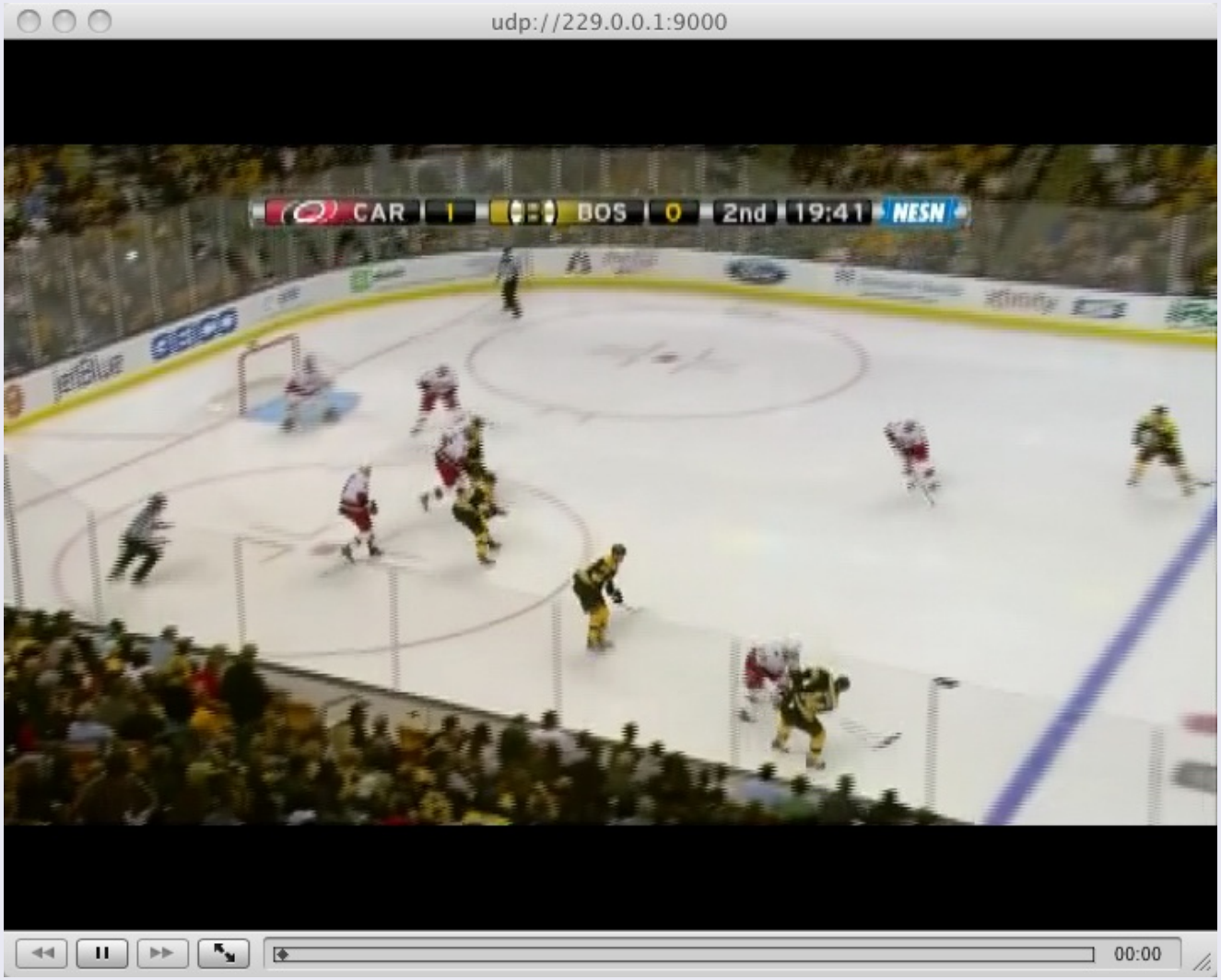
UDP Packet Analysis

UDP Packet Sample

```
<snip>
  0150:  00 00 00 00 00 00 00 00  47 40 21 32 07 10 05 3e   ........G@!2...>
  0160:  b4 f2 fe 7f 00 00 01 e0  00 00 84 80 05 21 29 f7   .............!).
  0170:  c8 53 00 00 01 00 01 1b  b3 eb b8 00 00 01 b5 84   .S..............
  0180:  44 47 84 00 00 00 01 b2  47 41 39 34 03 d4 ff fc   DG......GA94....
  0190:  80 80 fd 80 80 fa 00 00  fa 00 00 fa 00 00 fa 00   ................
  01a0:  00 fa 00 00 fa 00 00 fa  00 00 fa 00 00 fa 00 00   ................
  01b0:  fa 00 00 fa 00 00 fa 00  00 fa 00 00 fa 00 00 fa   ................
  01c0:  00 00 fa 00 00 fa 00 00  fa 00 00 ff 00 00 01 01   ................
  01d0:  1a ac 04 05 2b 00 00 01  02 1a ac 04 05 2b 00 00   ....+........+..
  01e0:  01 03 1a ac 04 05 2b 00  00 01 04 1a 57 39 e9 e2   ......+.....W9..
  01f0:  a9 cf 15 c5 4c e7 be f8  5a ab 80 b8 e8 1b ae 10   ....L...Z.......
  0200:  cc 9d 13 a6 a0 88 c4 ab  ed 48 ea 3a 8c a4 14 a8   .........H.:....
  0210:  b3 72 8b 31 47 00 21 d3  b4 79 e2 42 d9 e0 8f fa   .r.1G.!..y.B....
  0220:  55 11 aa 0f 75 05 86 16  1e 0b 71 26 97 73 5c c6   U...u.....q&.s..
  0230:  42 a0 ed d9 c6 7f 73 0a  23 28 42 6f b8 e2 56 fa   B.....s.#(Bo..V.
<snip>
```

Encapsulated Linear TV

NHL Centre Ice



Non-Linear Packets - First Attempts

Mangled Video Frame

Non-Linear Packets - Header Analysis

Grepping the header - First 0x10 bytes

```
$ dvbsnoop 0x320 | grep "^                0000:"
0000:  00 01 24 05 d5 5a fc a8  ae da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  85 48 10 00 00 00 00 00   ..$....q.H......
0000:  00 01 24 05 d5 5a fc a8  af da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  86 48 10 00 00 00 00 00   ..$....q.H......
0000:  00 01 24 05 d5 5a fc a8  b0 da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  87 48 10 00 00 00 00 00   ..$....q.H......
0000:  00 01 24 05 d5 5a fc a8  b1 da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  88 48 10 00 00 00 00 00   ..$....q.H......
```

Grepping the header - First 0x20 bytes

```
$ dvbsnoop 0x320 | grep "^                0000:" -A 1
0000:  00 01 24 05 e1 8a e7 71  85 48 10 00 00 00 00 00   ..$....q.H......
0010:  4a 83 f1 90 45 9c 52 41  29 0f 0e 87 e8 bb 30 3e   J...E.RA).....0>
--
0000:  00 01 24 05 e1 8a e7 71  86 48 10 00 00 00 00 00   ..$....q.H......
0010:  47 d6 93 9d 59 ab 12 f8  29 8c d0 30 12 44 f8 cb   G...Y...)..0.D..
--
0000:  00 01 24 05 e1 8a e7 71  87 48 10 00 00 00 00 00   ..$....q.H......
0010:  0e b4 43 05 74 9b cf 45  69 f4 7c 25 d4 58 3f 2b   ..C.t..Ei.|%.X?+
```

Non-Linear Packets - Field Identification

Grepping the header - First 0x10 bytes

```
$ dvbsnoop 0x320 | grep "^                0000:"
0000:  00 01 24 05 d5 5a fc a8  ae da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  85 48 10 00 00 00 00 00   ..$....q.H......
0000:  00 01 24 05 d5 5a fc a8  af da 00 00 00 00 00 00   ..$..Z..........
0000:  00 03 ab 00 e1 8a e7 71  a3 00 00 00 93 00 00 00   .......q........
0000:  00 01 24 05 e1 8a e7 71  86 48 10 00 00 00 00 00   ..$....q.H......
0000:  00 01 24 05 d5 5a fc a8  b0 da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  87 48 10 00 00 00 00 00   ..$....q.H......
0000:  00 01 24 05 d5 5a fc a8  b1 da 00 00 00 00 00 00   ..$..Z..........
0000:  00 01 24 05 e1 8a e7 71  88 48 10 00 00 00 00 00   ..$....q.H......
```

Finding Non-Payload Packets

Grepping the header - First 0x10 bytes

```
$ dvbsnoop 0x320 | grep "^      0000:  " | grep -v "^      0000:  00 01"
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 99 08 00 3e 23 fd a1                            ........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 06 08 00 af d3 11 23                            ........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
0000:  00 03 ab 00 91 47 a6 fb  a3 00 00 00 93 00 00 00   .....G.........
```

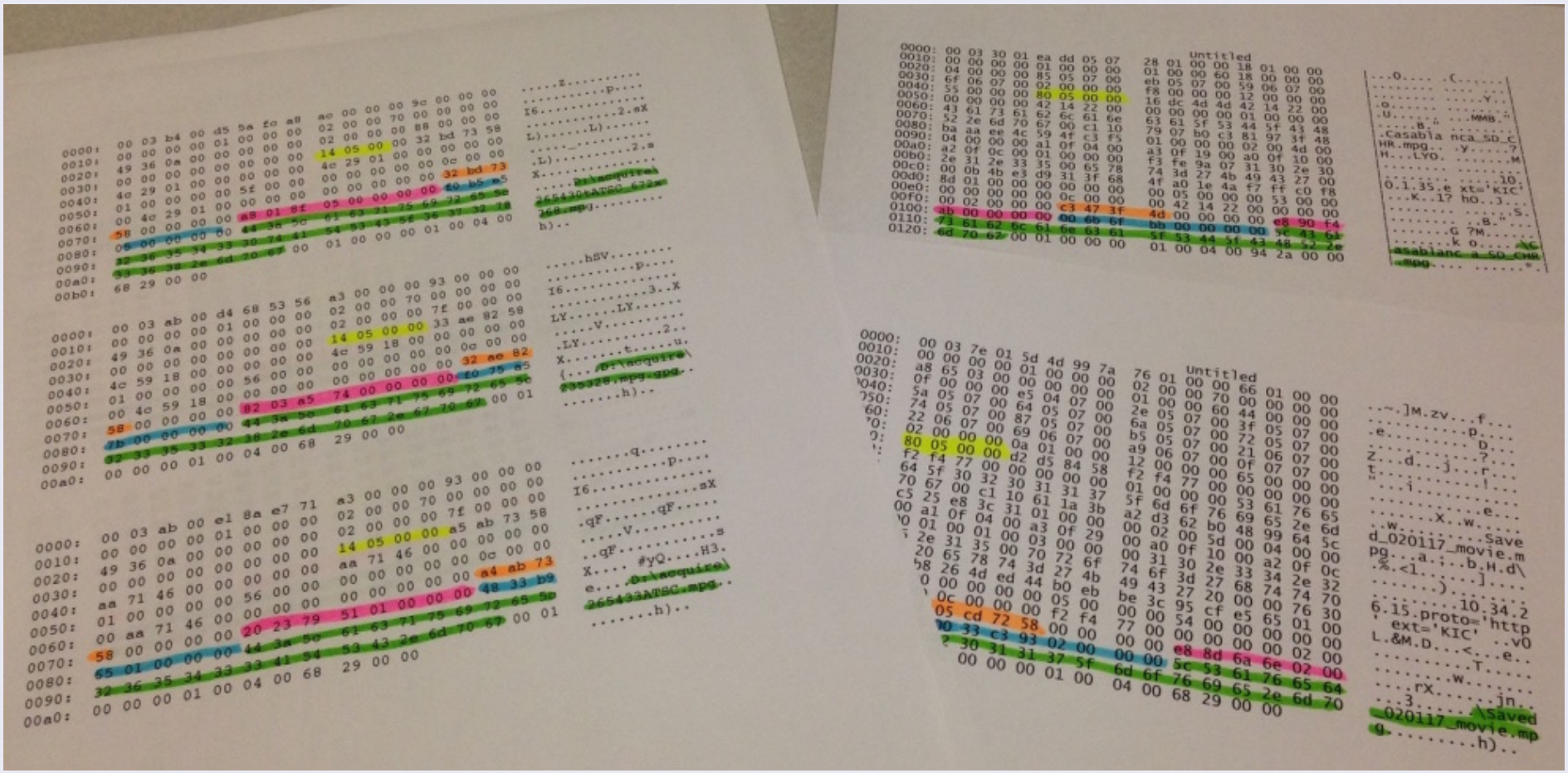
03 Packet Dissection

03 Packet Dump w/ Filename Visible

```
0000:  00 03 ab 00 aa f2 a6 af  a3 00 00 00 93 00 00 00   ................
0010:  00 00 00 00 01 00 00 00  02 00 00 70 00 00 00 00   ...........p....
0020:  49 36 0a 00 00 00 00 00  02 00 00 00 7f 00 00 00   I6..............
0030:  00 00 00 00 00 00 00 00  14 05 00 00 b9 c4 82 58   ...............X
0040:  90 bf 2b 00 00 00 00 00  90 bf 2b 00 00 00 00 00   ..+.......+.....
0050:  01 00 00 00 00 56 00 00  00 00 00 00 00 0c 00 00   .....V..........
0060:  00 90 bf 2b 00 00 00 00  00 00 00 00 00 b8 c4 82   ...+............
0070:  58 00 00 00 00 84 63 95  d1 00 00 00 00 40 c7 28   X.....c......@.(
0080:  de 00 00 00 00 44 3a 5c  61 63 71 75 69 72 65 5c   .....D:\acquire\
0090:  32 33 35 33 33 32 41 54  53 43 2e 6d 70 67 00 01   235332ATSC.mpg..
00a0:  00 00 00 01 00 04 00 68  29 00 00                 .......h)..
```

03 Packet Token Identification

Highlighted Printouts



New Problems

- **data volume** - Able to fill a 2TB drive in 1-2 days
- **overall bitrate** - Unable to perform disc operations without read/write buffer overruns

First Attempts at dealing with IO Bottlenecks

vm.dirty_background_ratio=10

```
$ vmstat -n 1
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 2952356  45356 580008    0    0     0     0  429  871  2  2 97  0  0
 0  0      0 2943524  45364 588088    0    0     0 45077  497  916  0  2 97  2  0
 0  0      0 2935544  45372 596472    0    0     0  1441  589  935  1  4 92  3  0
 0  0      0 2927080  45372 604664    0    0     0     0  422  857  0  1 99  0  0
 0  0      0 2918508  45372 612828    0    0     0     0  432  879  1  3 96  0  0
 0  0      0 2910060  45372 621140    0    0     0     0  434  881  1  3 97  0  0
 0  1      0 2901124  45384 628960    0    0     0 90133  522  999  0  2 94  4  0
 0  0      0 2893168  45396 637312    0    0     0 30196  912 1014  1  3 79 17  0
 0  0      0 2884628  45396 645608    0    0     0     1  450  933  0  1 99  0  0
```

vm.dirty_background_ratio=0

```
$ vmstat -n 1
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 1736504  45676 1761948    0    0   121   440  177  503  1  1 98  1  0
 0  0      0 1728304  45676 1770188    0    0     0  8208  490  929  1  2 97  0  0
 0  1      0 1719732  45684 1778468    0    0     0  8212  506  955  2  3 94  1  0
 0  0      0 1711532  45688 1786640    0    0     0  8216  489  912  0  0 97  3  0
 0  0      0 1703084  45688 1794840    0    0     0  8204  491  937  1  4 95  0  0
 1  0      0 1694512  45688 1803128    0    0     0  8164  482  941  1  3 96  0  0
```

Forward Error Correction

File Block 0

```
$ {2 Block File} -C -n 32 -skip 0
00000000  46 69 6c 65 3d 73 63 68  65 64 5f 63 6f 72 65 37  |File=sched_core7|
00000010  2d 38 2d 34 5f 6d 75 6c  74 69 70 6c 65 2d 37 35  |-8-4_multiple-75|
```

File Block 1

```
$ {2 Block File} -C -n 32 -skip 1300
00000000  46 69 6c 65 3d 73 63 68  65 64 5f 63 6f 72 65 37  |File=sched_core7|
00000010  2d 38 2d 34 5f 6d 75 6c  74 69 70 6c 65 2d 37 35  |-8-4_multiple-75|
```

Galois Field

Galois Field Important Properties

- Any operation on element is another element in the field
- Finite - 2^n elements in a GF

GF for 2^8 elements

$$GF(2^8) = x^8 + x^4 + x^3 + x^2 + 1 \tag{1}$$

Galois Field Math Operations

Galois Field Math Operations

- **addition** - XOR
- **subtraction** - XOR
- **multiplication** - lookup table of 2^{16} elements
- **division** - lookup table of 2^{16} elements

Vandermonde FEC

Equations for Vandermonde n/k FEC

$$y_k = x_n * G \tag{2}$$

$$x_n = z_n * A^{-1} \tag{3}$$

Variable	Description
x_n	Original Data
y_k	Transmitted Codewords
z_n	Received Codewords
G	Generator Matrix
A^{-1}	Repair Matrix

FEC Example Part 1

Vandermonde Matrix

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \cdots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \cdots & \alpha_m^{n-1} \end{bmatrix} \tag{4}$$

FEC Example Part 2

2x3 Generator Matrix

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \tag{5}$$

Figure: 2/3 Vandermode Matrix

$$\begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \end{bmatrix} \tag{6}$$

Figure: 2/3 Generator Matrix

FEC Example Part 3

Identity Matrix

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{7}$$

FEC Example Part 4

Generating Code Words

$$y_k = x_n * G \tag{8}$$

$$y_k = \begin{bmatrix} 1F & F7 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \end{bmatrix} \tag{9}$$

$$y_k = \begin{bmatrix} (1F \cdot 1) + (F7 \cdot 0) & (1F \cdot 0) + (F7 \cdot 1) & (1F \cdot 3) + (F7 \cdot 2) \end{bmatrix} \tag{10}$$

$$y_k = \begin{bmatrix} 1F + 0 & 0 + F7 & 21 + F3 \end{bmatrix} \tag{11}$$

$$y_k = \begin{bmatrix} 1F & F7 & D2 \end{bmatrix} \tag{12}$$

FEC Example Part 5

Generating the Repair Matrix

$$A = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix} \tag{13}$$

$$A^{-1} = \begin{bmatrix} 1 & 8F \\ 0 & 8E \end{bmatrix} \tag{14}$$

FEC Example Part 6

Regenerating Original Data

$$x_n = z_n * A^{-1} \tag{15}$$

$$x_n = \begin{bmatrix} 1F & D2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 8F \\ 0 & 8E \end{bmatrix} \tag{16}$$

$$x_n = \begin{bmatrix} (1F \cdot 1) + (D2 \cdot 0) & (1F \cdot 8F) + (D2 \cdot 8E) \end{bmatrix} \tag{17}$$

$$x_n = \begin{bmatrix} 1F + 0 & 9E + 69 \end{bmatrix} \tag{18}$$

$$x_n = \begin{bmatrix} 1F & F7 \end{bmatrix} \tag{19}$$

C Example of FEC Header

C Example

```
/* 2/X FEC */
uint8_t fec_g_matrix_2_x[4] = {
    3,2,
    2,3
    };

/* 4/X FEC */
uint8_t fec_g_matrix_4_x[16] = {
    27,28,18,20,
    28,27,20,18,
    18,20,27,28,
    20,18,28,27
    };
```

Questions

Questions and Answers