Ultra HD over ATSC 1.0?!

We didn't say it'd be easy

WI Broadcasters Clinic 2024

Anton Kapela, Channel 3, Eugene - K03IM-D

CONTEXT

This talk is not about a translator, it’s a LP-D

This talk is about how someone used mostly open source tools to get UltraHD programs working over ATSC 1.0

What
How
Wher
e

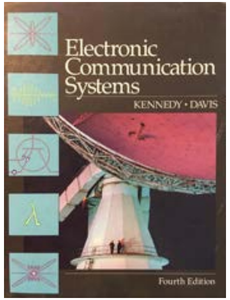
Past
Present
Future

Past: How'd we get here?

Glossary - Warning: buzzwords ahead!

- **AVC**: Advanced Video Coding, H.264 or MPEG-4 Part 10
- **HEVC**: High Efficiency Video Coding, H.265 and MPEG-H Part 2
- **SSIM/SSIMPLUS**: Structural similarity index measure, extensions matched to human subjective ratings
- **CODEC**: portmanteau of coder/decoder
- **SPTS**: single program transport stream
- **MPTS**: multi program transport stream
- **PCR**: Program Clock Reference
- **PTS/DTS**: Presentation Time Stamp/Decode Time Stamp
- **GOP**: group of pictures, usually all pictures between I/IDR frames
- **MiniGOP**: series of B frames between P frames (display order)
- **DBP**: Decoded Picture Buffer
- **I/IDR**: Intra-coded frame, depends on no other frames; "instantaneous decoder refresh" = intra coded frame + flush all prior frames in **DBP**
- **UHD**: Ultra HD resolution, ie. four panels of 1920x1080 = 3840x2160
- **RTT**: Round-trip time, usually referring to measured latency on a packet switched network
- **HRD**: An explicit signalling and buffering state *anticipation* model defined by a "hypothetical reference decoder"

Journey Into Wireless



Aironet
900Mhz
1999/00

Wi Fi

Mesh
Madison
2002

Wi-NOG
Wireless - Network Operators Group

5NINES

1996
Garage Sale

2001
Ooh!
WiFi

2006+
Wireless
for real

Past: But Who is this Anton guy?

KIM ZETTER SECURITY AUG 26, 2008 5:00 PM

Revealed: The Internet's Biggest Security Hole

Two security researchers have demonstrated a new technique to stealthily intercept internet traffic on a scale previously presumed to be unavailable to anyone outside of intelligence agencies like the National Security Agency. The tactic exploits the internet routing protocol BGP (Border Gateway Protocol) to let an attacker surreptitiously monitor unencrypted internet traffic anywhere in the [...]

SAVE

But Who is this Anton guy?

SNIPER IN MAHWAH & FRIENDS

It's all about market structure. "Pretium iustum mathematicum licet soli Deo notum"

NETWORK EFFECTS | PART II

26 June 2017 — 7 Comments

The nondisclosure agreements have lapsed. The Chicago to New Jersey microwave arms race has converged to a few winners. Many of the early participants must now be eying the runaway success (and the glaring shortcomings) of a certain HFT-associated bestseller, and thinking, yeah, "*I could do that.*"

It's a fantastic story, after all, and it hasn't really been told yet. It seems like there are two basic approaches. You could write a cinema-ready page-turner heavy on the skulduggery. Antennas knocked out of alignment the night before the jobs number. Unlicensed broadcasting on cognitive radio. Itinerant con men peddling futures on networks that will never exist. Or you could try to write a book with longer-term importance that draws the details within the larger context and paradoxes of the modern-day United States.

With the transect list in hand, it was straightforward to step through the firms and plot their antenna locations. A Chicago to New York line-of-sight-relay would presumably be obvious, even at a glance. First on the list was "AB Services", but the FCC website had suddenly slowed to an infuriating crawl. We waited for nearly a minute. Finally a map appeared on the screen, triggering a mixture of awe and disappointment.

A quick back of the envelope calculation indicated that even with off-the-shelf radio latencies, AB Networks, as licensed in the FCC database, was *easily* capable of beating Spread's fiber. Google linked the LLC to Anton Kapela and Alex Pilosov, two gentlemen who, if nothing else, appeared to have a variety of marketable tech skills. A *Wired* article from 2008 reported on a presentation they'd given at the DefCon hacker conference:

"...BGP eavesdropping has long been a theoretical weakness, but no one is known to have publicly demonstrated it until Anton 'Tony' Kapela, data center and

Runup to UHD

Worked in/near A/V/IPTV for much of my career; data centers, ISPs, microwave, mobility networks, you name it: if it moves bits, I touch(ed) it

..But Mostly Ignored Broadcast

Everything was great, until one day:

Happened to buy a UHD TV in 2020; nothing on the air besides MPEG2…

No ATSC 3 in Madison, and this dumb TV didn’t even support it!

..couldn’t let this stand; *had* to get something working with UHD resolutions

But ...

Question: Can 19.3 megabits do anything visually interesting?

...with HEVC?

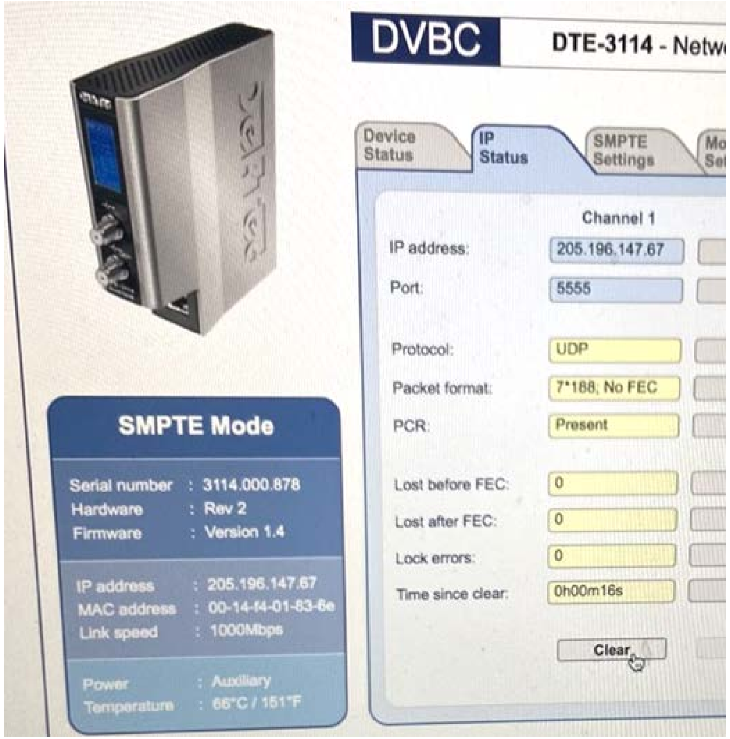
And: Can anything DECODE HEVC over a normal Transport Stream (TS) ...over the air (or cable)?

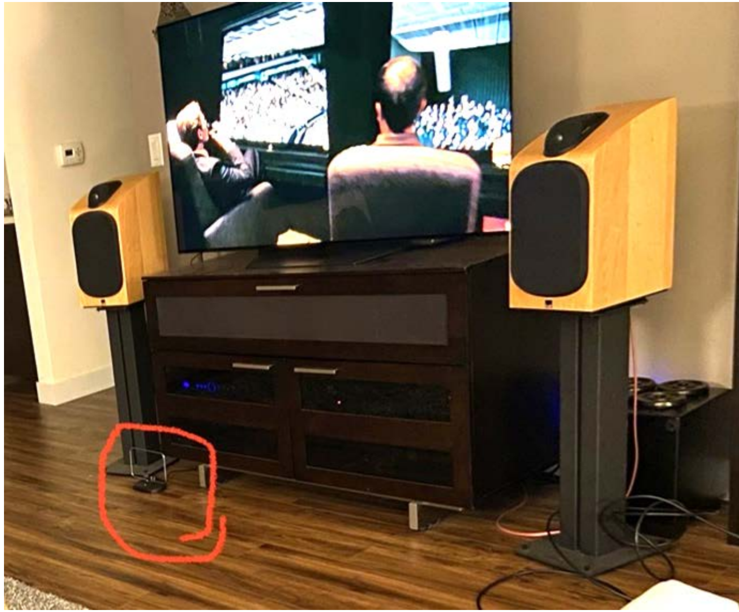
Let's get something working

- **Baby step**: Buy cheap Dektec DVB-C modulator; test with SPTS, no frills goal: one audio + video PID + ffmpeg playout working

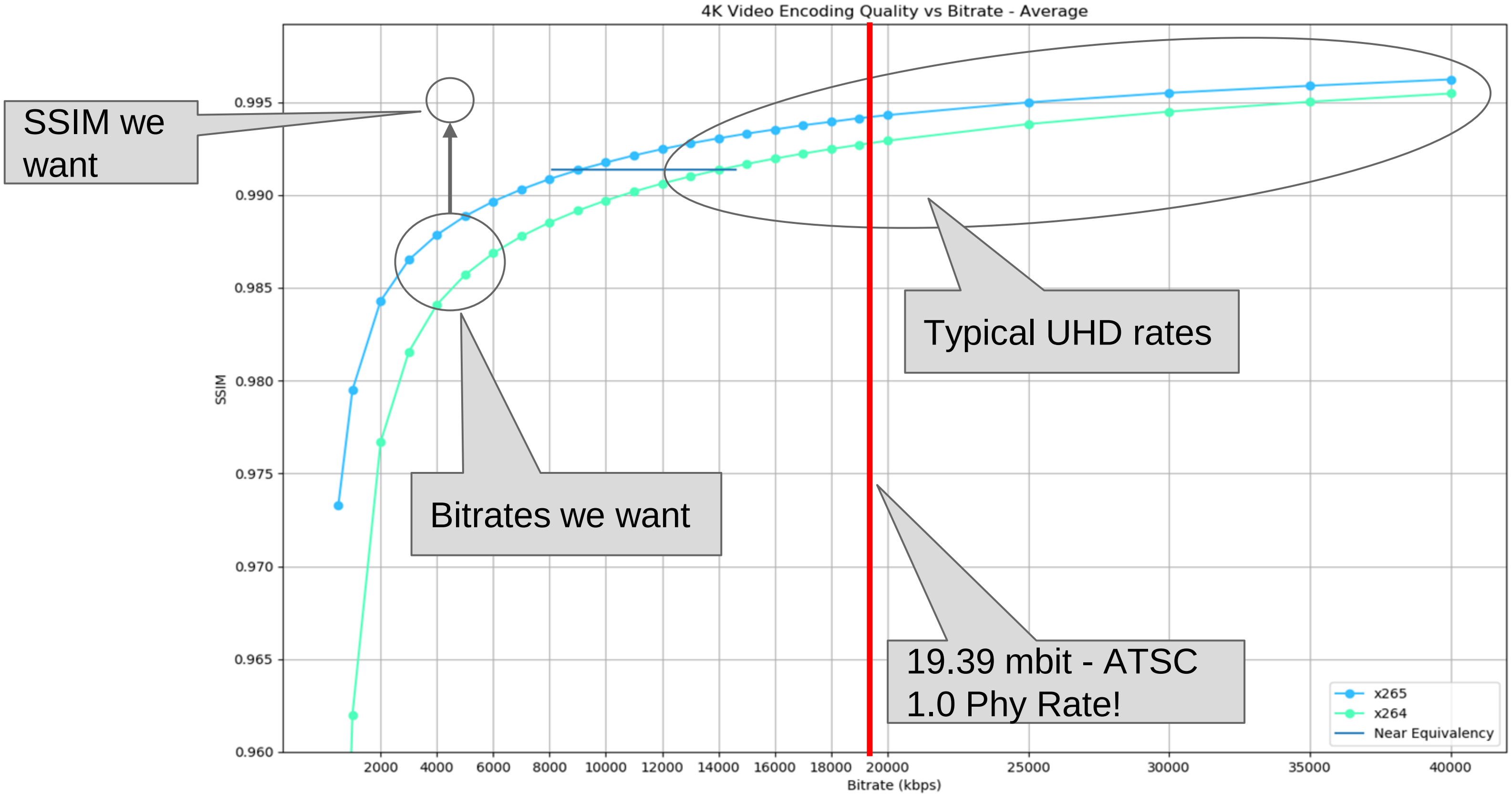
- Takeaway: It works, but **poorly**, out of the box:
  - PCR dynamics matter, TS shaping matters, GOP params ***really*** *matter*

Let's get something working





A Starting Point for UHD - Rate Targets Circa 2020

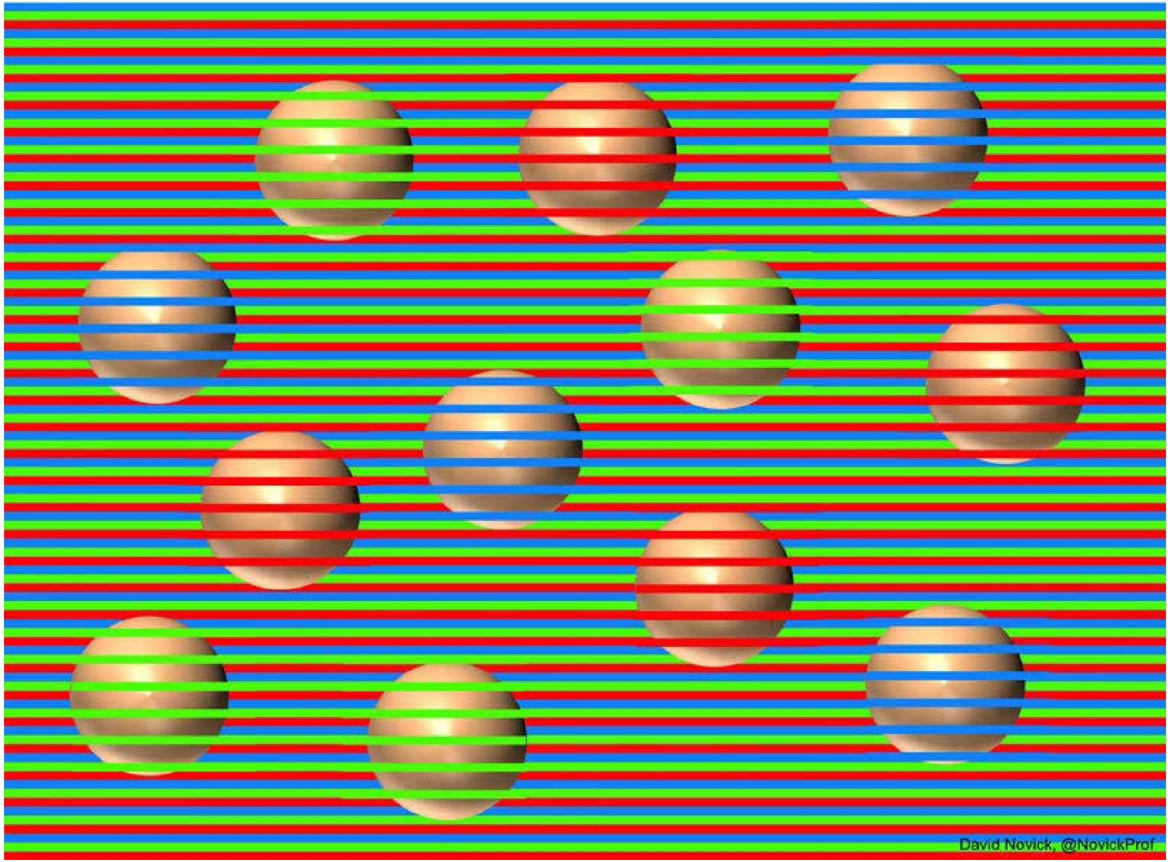


Credit: https://codecalamity.com/encoding-settings-for-hdr-4k-videos-using-10-bit-x265/

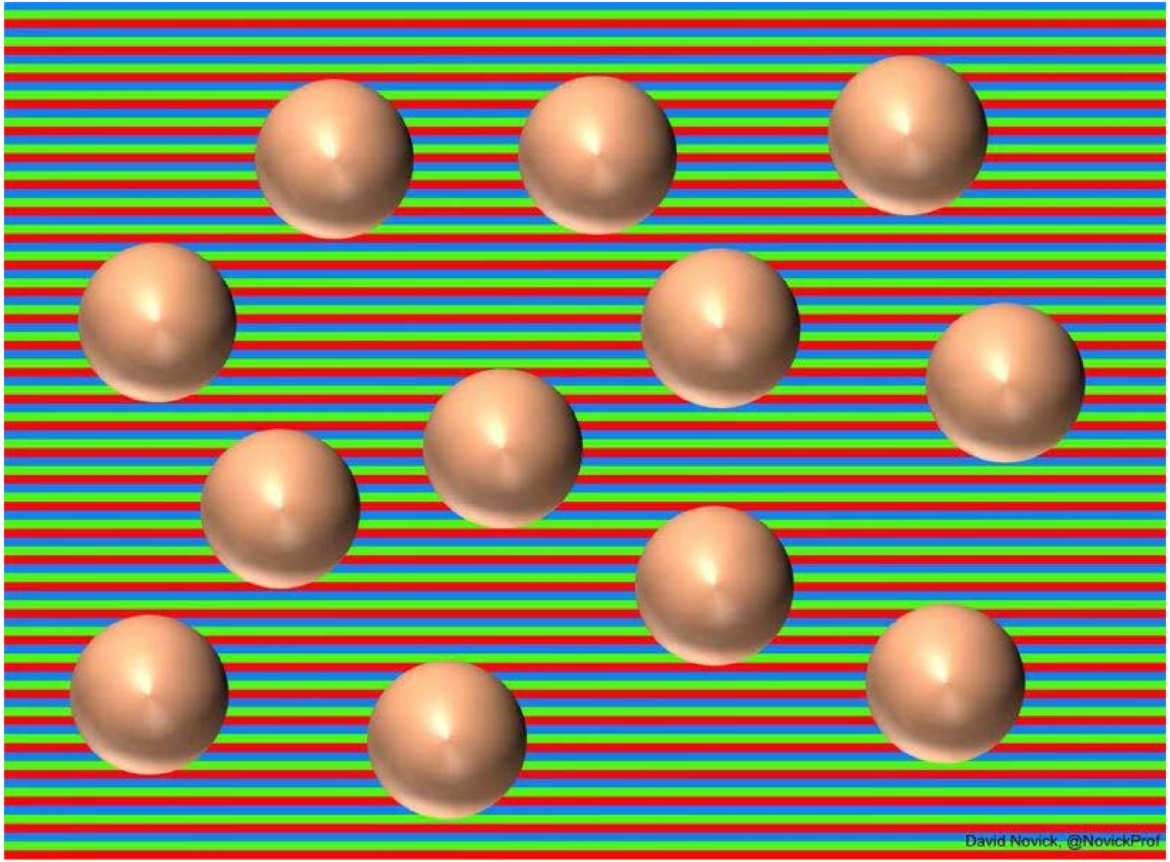
What's in a CODEC?

Lies.

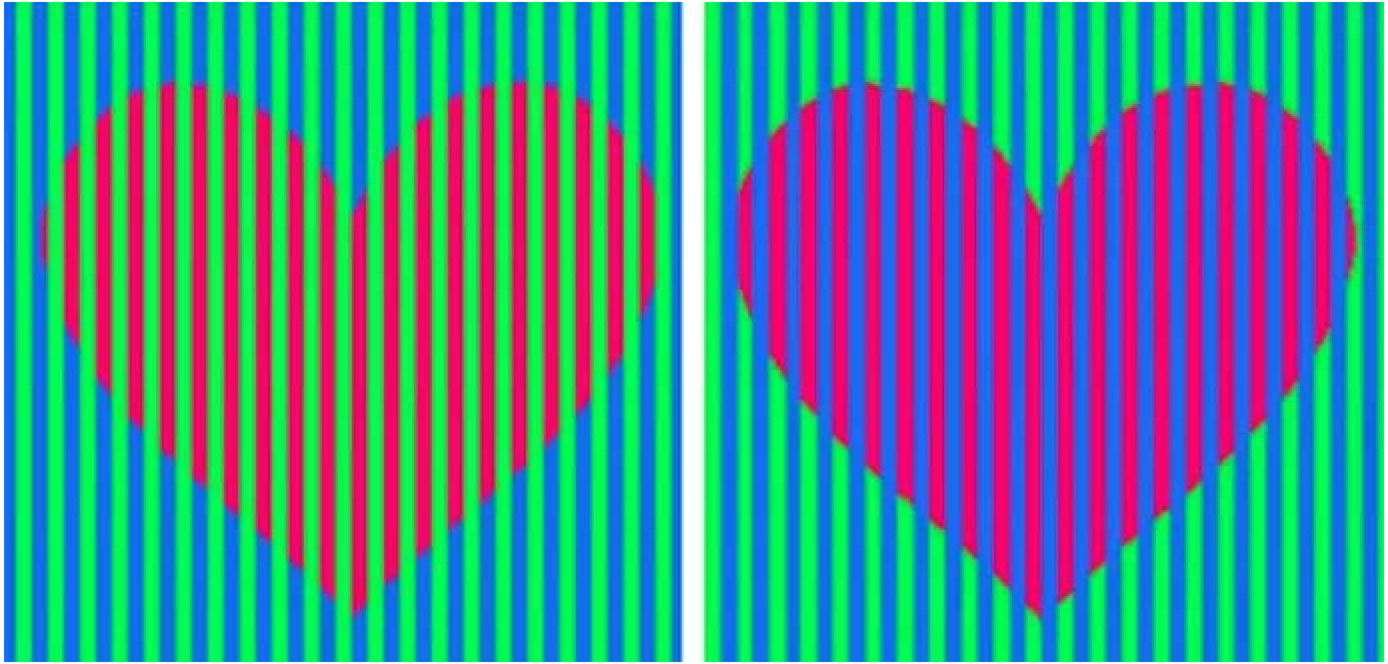
What's in a CODEC?



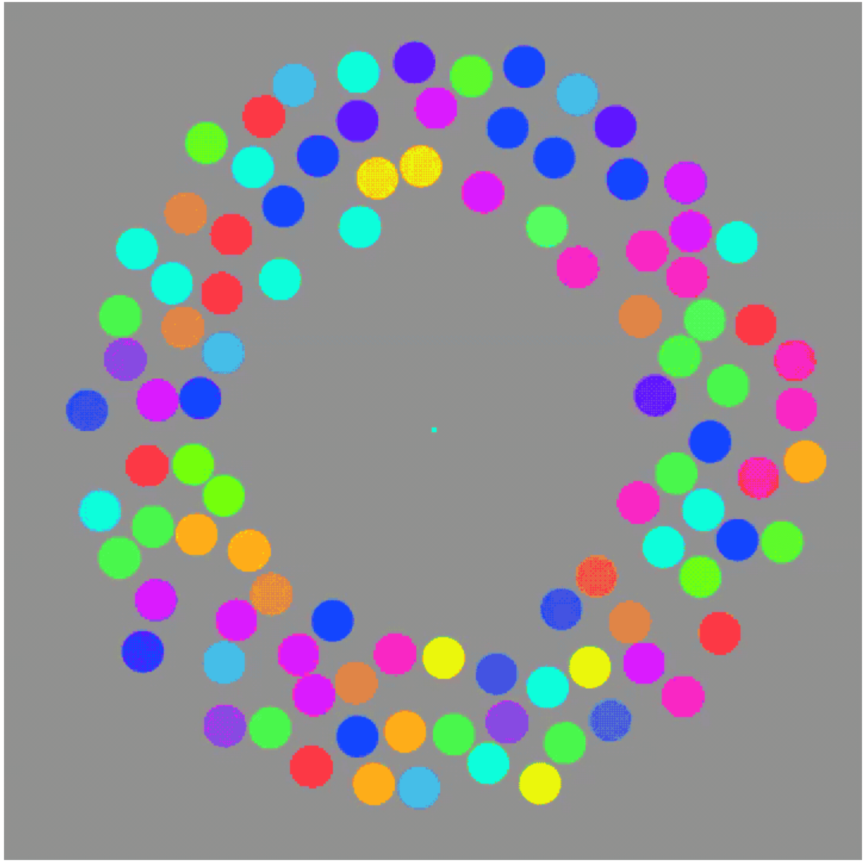
What's in a CODEC?

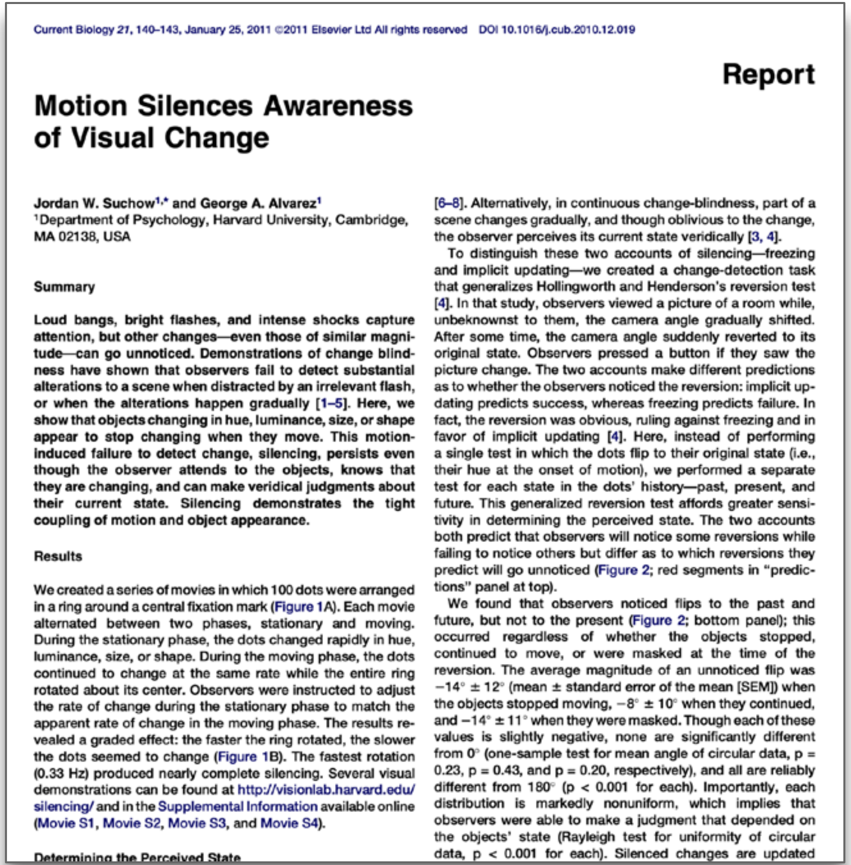


What's in a CODEC?



Q: What's in a CODEC? A: Silencing (c. 2010)





Current Biology *21*, 140–143, January 25, 2011 DOI 10.1016/j.cub.2010.12.019

Report

Motion Silences Awareness of Visual Change

Jordan W. Suchow[1,*] **and George A. Alvarez**[1]
[1]Department of Psychology, Harvard University, Cambridge, MA 02138, USA

Summary

Loud bangs, bright flashes, and intense shocks capture attention, but other changes—even those of similar magnitude—can go unnoticed. Demonstrations of change blindness have shown that observers fail to detect substantial alterations to a scene when distracted by an irrelevant flash, or when the alterations happen gradually [1–5]. Here, we show that objects changing in hue, luminance, size, or shape appear to stop changing when they move. This motion-induced failure to detect change, silencing, persists even though the observer attends to the objects, knows that they are changing, and can make veridical judgments about their current state. Silencing demonstrates the tight coupling of motion and object appearance.

Results

We created a series of movies in which 100 dots were arranged in a ring around a central fixation mark (Figure 1A). Each movie alternated between two phases, stationary and moving. During the stationary phase, the dots changed rapidly in hue, luminance, size, or shape. During the moving phase, the dots continued to change at the same rate while the entire ring rotated about its center. Observers were instructed to adjust the rate of change during the stationary phase to match the apparent rate of change in the moving phase. The results revealed a graded effect: the faster the ring rotated, the slower the dots seemed to change (Figure 1B). The fastest rotation (0.33 Hz) produced nearly complete silencing. Several visual demonstrations can be found at http://visionlab.harvard.edu/silencing/ and in the Supplemental Information available online (Movie S1, Movie S2, Movie S3, and Movie S4).

Determining the Perceived State

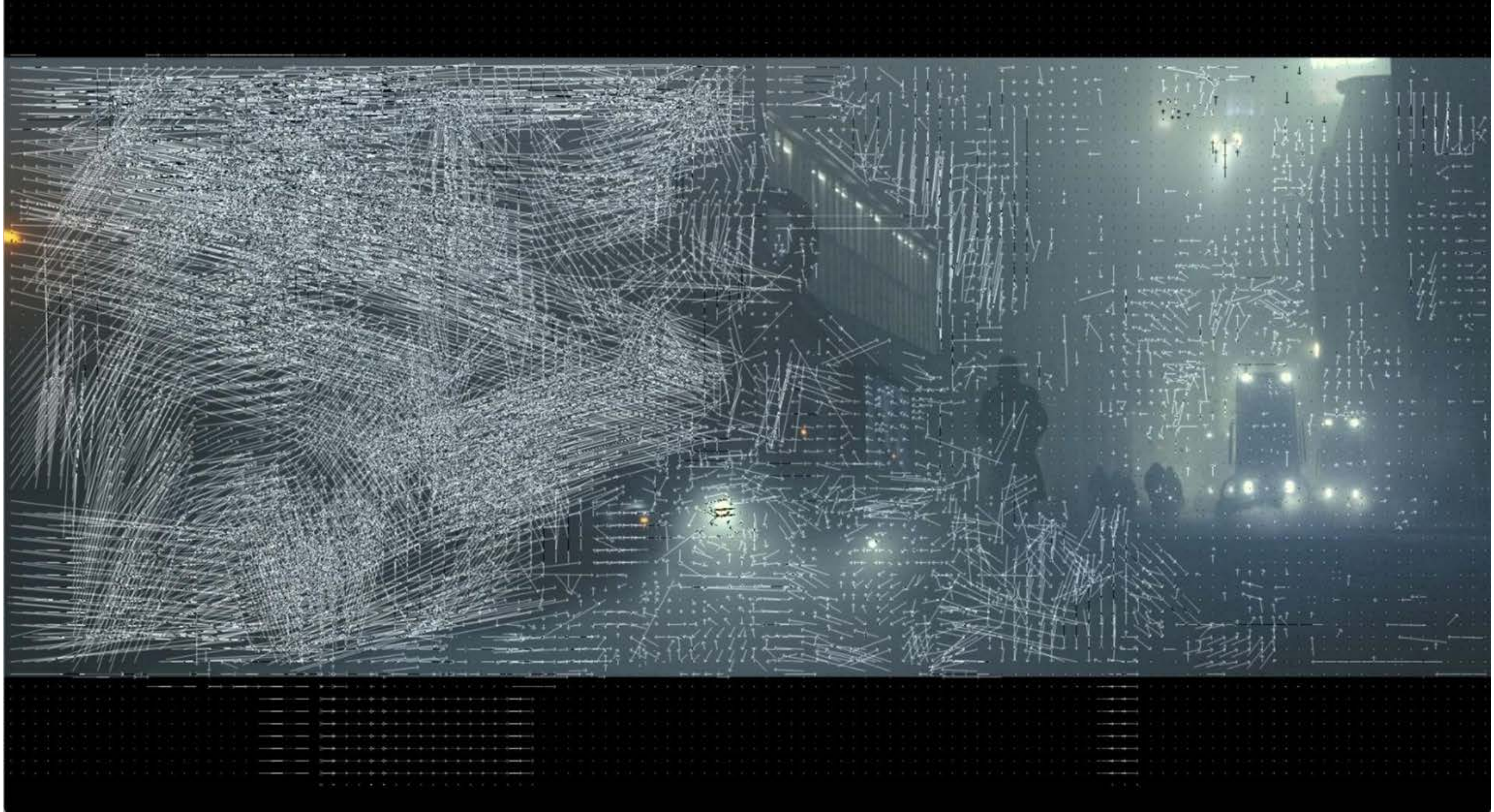
[6–8]. Alternatively, in continuous change-blindness, part of a scene changes gradually, and though oblivious to the change, the observer perceives its current state veridically [3, 4].

To distinguish these two accounts of silencing—freezing and implicit updating—we created a change-detection task that generalizes Hollingworth and Henderson's reversion test [4]. In that study, observers viewed a picture of a room while, unbeknownst to them, the camera angle gradually shifted. After some time, the camera angle suddenly reverted to its original state. Observers pressed a button if they saw the picture change. The two accounts make different predictions as to whether the observers noticed the reversion: implicit updating predicts success, whereas freezing predicts failure. In fact, the reversion was obvious, ruling against freezing and in favor of implicit updating [4]. Here, instead of performing a single test in which the dots flip to their original state (i.e., their hue at the onset of motion), we performed a separate test for each state in the dots' history—past, present, and future. This generalized reversion test affords greater sensitivity in determining the perceived state. The two accounts both predict that observers will notice some reversions while failing to notice others but differ as to which reversions they predict will go unnoticed (Figure 2; red segments in "predictions" panel at top).

We found that observers noticed flips to the past and future, but not to the present (Figure 2; bottom panel); this occurred regardless of whether the objects stopped, continued to move, or were masked at the time of the reversion. The average magnitude of an unnoticed flip was $-14° \pm 12°$ (mean ± standard error of the mean [SEM]) when the objects stopped moving, $-8° \pm 10°$ when they continued, and $-14° \pm 11°$ when they were masked. Though each of these values is slightly negative, none are significantly different from 0° (one-sample test for mean angle of circular data, $p = 0.23$, $p = 0.43$, and $p = 0.20$, respectively), and all are reliably different from 180° ($p < 0.001$ for each). Importantly, each distribution is markedly nonuniform, which implies that observers were able to make a judgment that depended on the objects' state (Rayleigh test for uniformity of circular data, $p < 0.001$ for each). Silenced changes are updated

Applied Silencing - forward & backward MV

Motion Est. Range = 128



Motion Est. Range = 512

All of the books in the world contain no more information than is broadcast as video in a single large American city in a single year. Not all bits have equal value.

- Carl Sagan

Visual Entropy: search and encode!

Setup a "personal cloud" to explore transcoding options in HEVC with x.265

Didn't anticipate the iterative work & refinement. Almost gave up: commercial encoders make it *so easy* to get reasonable results

That's not the "hacker way"

~750 GHz lets you figure it out with brute-force

What does that look like?

Run through variations of 1000's of these:

Average Case: “Offline” transcode example - HDR10 to 4.5mbit ABR+VBR:

```
$ ffmpeg -fflags +discardcorrupt -ec guess_mvs+deblock+favor_inter -i input.mkv -
vf'scale=3840:2160:force_original_aspect_ratio=decrease,
pad=3840:2160:-1:-1:color=black'-c:a ac3 -ac 2 -ar 48000 -c:v libx265 -preset medium -
x265-params 'hdr-opt=1:repeat-
headers=1:colorprim=bt2020:transfer=smpte2084:colormatrix=bt2020nc:master-
display=G(13250,34500)B(7500,3000)R(34000,16000)WP(15635,16450)L(10000000,50):max-
cll=3201,386:keyint=72:ref=5:bframes=3:b-adapt=2:bitrate=4000:vbv-maxrate=4500:vbv-
bufsize=8000:merange=256:me=hex:no-open-gop=1:hrd=1:aq-mode=3:pmode=1:rect=1:rc-
lookahead=36' -map_metadata -1 -map_chapters -1 output.mkv
```

~3 FPS

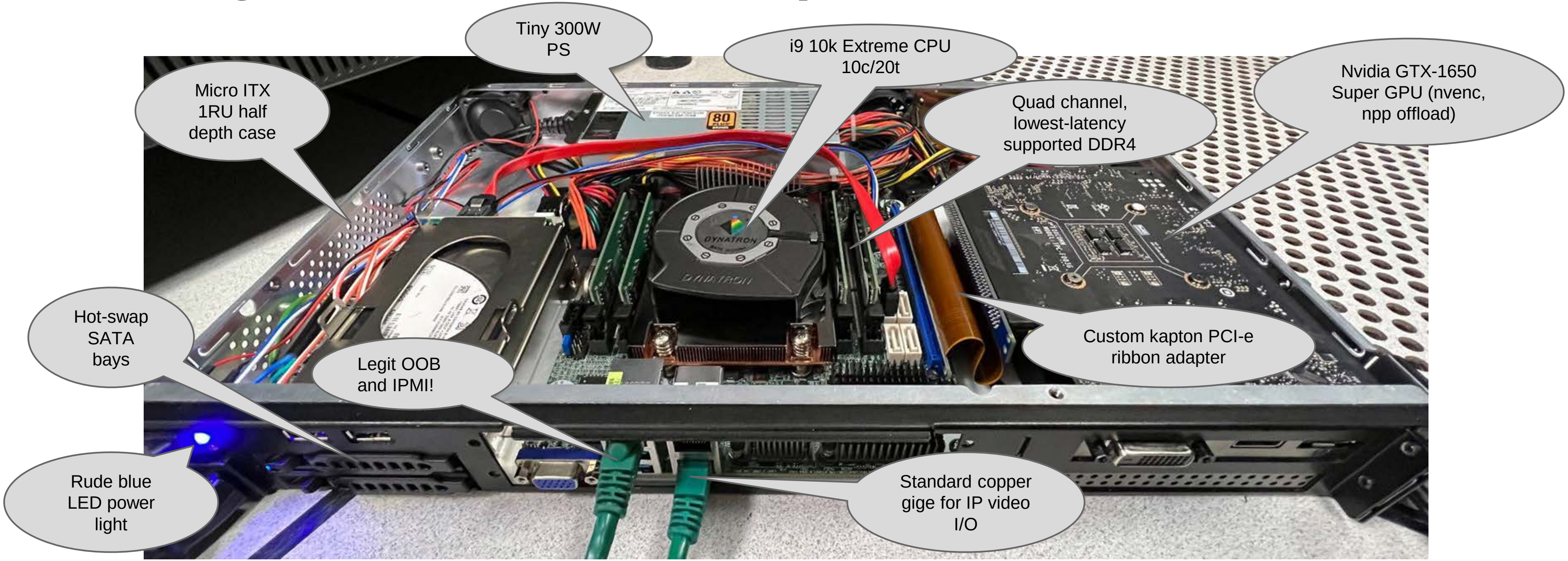
This is as good as it gets in 2022/23: ~20-threads per node, latest-greatest x.265 and Intel Xeon E5 v4 cpus, 3840x2160 main10

~70 watt/seconds ***per frame, per node***

Result: taking *non-noisy* things close to the **visual SSIM rate floor: ~4.5 Mbits coded rate @2160**

Ok, what about *live* HEVC UHD transcoding?

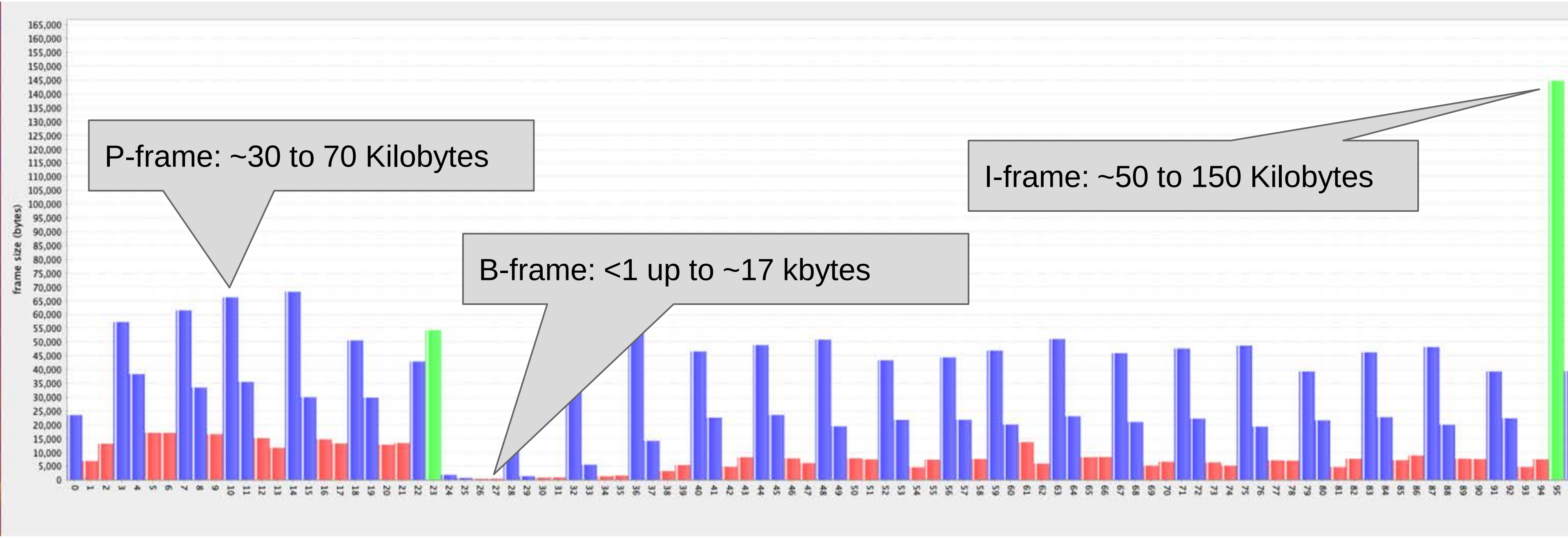
- Nothing commercially *optimal* for me: wanted good VBR at low bitrates, cheap, off the shelf
    - First person to mention “but SVT or mainconcept ...” gets booted
- Own integration: FOSS code, standard x86 parts, Nvidia GPU, CUDA/NVENC API



Magic: Live transcode NASA UHD to 10-bit SDR at ~4 Mbits VBR on GPU:

```
while :; do ffmpeg -fflags +discardcorrupt -i
udp://238.1.1.1:2000?fifo_size=400000\?overrun_nonfatal=1\&buffer_size=10000
000 -map 0:i:4161 -map 0:i:4164 -vf fps=fps=30000/1001 -pix_fmt p010le -c:v
hevc_nvenc -g 90 -preset p7 -b:v 3.2M -maxrate 4.5M -bufsize 8M -bf 2 -refs
5 -rc-lookahead 27 -weighted_pred 0 -b_ref_mode each -nonref_p 0 -spatial_aq
1 -temporal_aq 1 -aq-strength 8 -forced-idr 1 -strict_gop 1 -c:a copy -
program title="NASA-4K":st=0:st=1 -color_primaries bt709 -color_trc bt709 -
colorspace bt709 -f mpegts -mpegts_start_pid 160 -max_interleave_delta
200000 -muxpreload 2 -flush_packets 0
udp://224.2.2.254:1466?pkt_size=1316\&bitrate=8121600 ; sleep 1 ; done
```

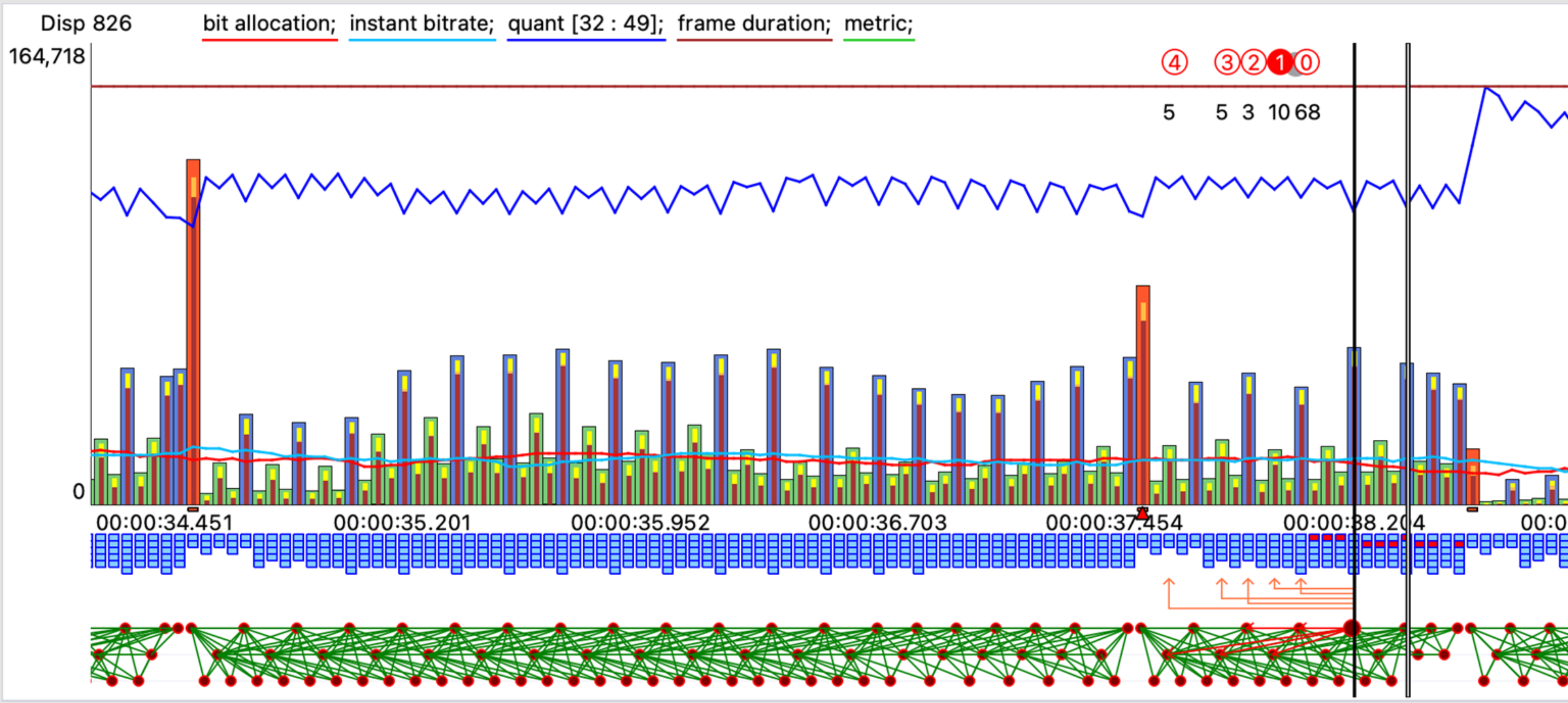
Coded UHD pictures on the wire:



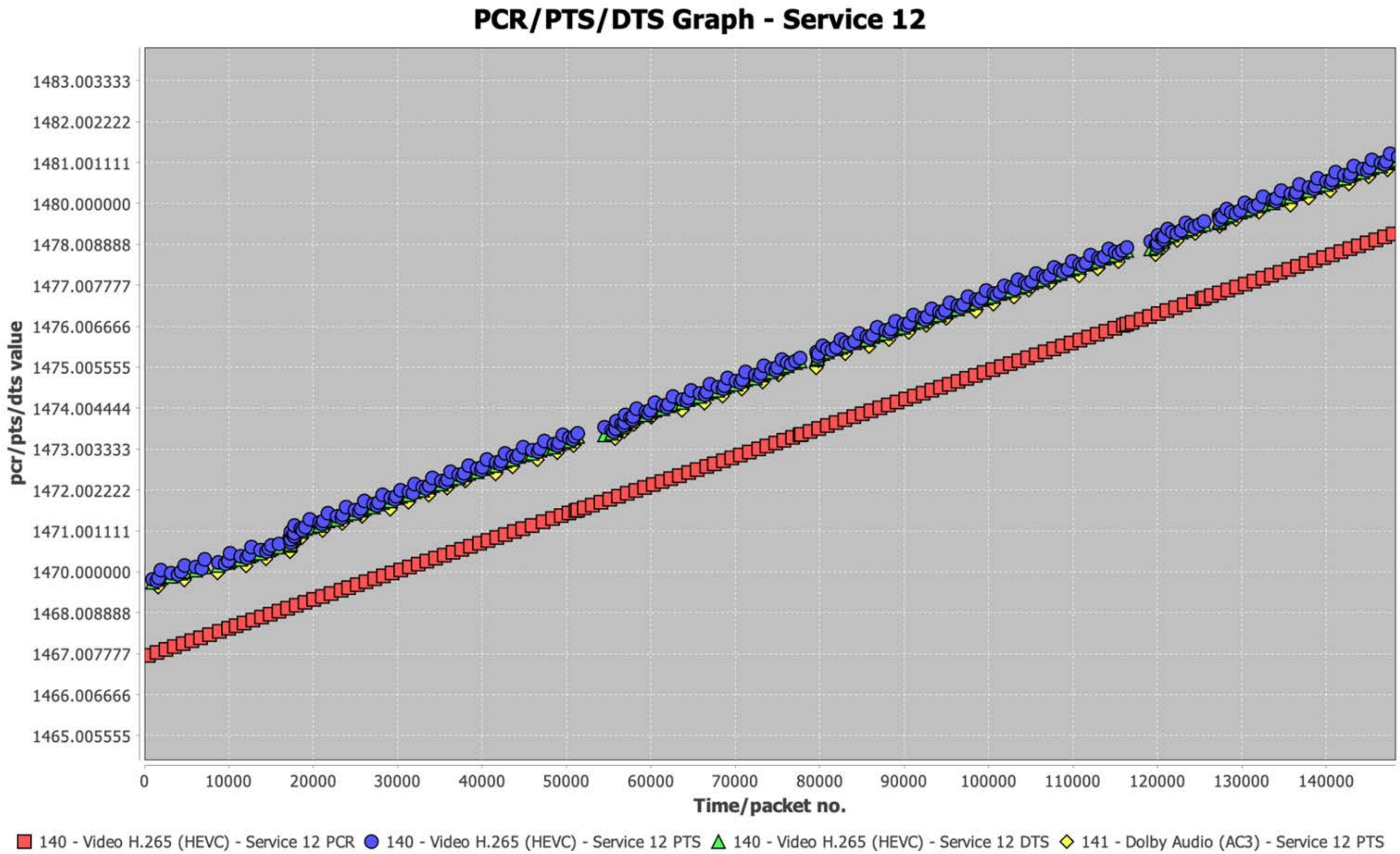
Viewed another way…Presentation order

Props to Elecard StreamEye!

→



TS packet time stamps on the wire:



So what's the ~~problem~~ challenge of VBR streams?

- It's just not nice to mix packetized elementary stream (PES) objects (ie. encoded video frames) that are >2 orders of mag different in size, into the same transport stream (TS), while holding a constant output rate, with competing programs, unless we do some special things in the multiplexer:

1. Hold *just enough* TS packets in input buffers
2. "Smear" TS packets over time at defined output rate (~19,392,658 bits/sec for ATSC 1.0/8VSB)
3. Dequeue packets from input queues using something like fair queuing, with deficit-weighted round robin (FQ-DWRR)
4. Interpose TS packets from all streams in a *mostly-PCR-position* preserving manner
5. Try to ensure output DTS value is always greater than PCR value (ie. carefully manage burst & average rate sum from encoders, discard some TS packets if we're "under water")

Enablers for ultra-low rate HD/UHD 1

Absolutely Essential: to reach the *visual-coding-floor* in HEVC, we need:

- Large coded picture buffer, with HRD signaling
- Long & fully dynamic GoP construction
- Fully dynamic mode decisions
- Exhaustive/full-frame motion search area
- Mixed (B and P) reference frames
- “Smart” bit allocation/rate distortion across the ***whole*** GoP ***and*** mini-GoPs

Enablers for ultra-low rate HD/UHD 2

These encoder features (in previous slide) have substantial compute/power cost, generally **not** fully exploited by OTT/streamers, and impractical at YouTube/etc scale.

They are *eminently* practical for broadcast:

one screen target, *one* rate target, *one* program/stream.

Enablers for ultra-low rate HD/UHD Cont.

Channel 3 Refinement wishlist:

- Efficient *hierarchical* motion estimation in HEVC
- Flexible direct and synthetic prediction modes (ie. “grain simulation” and vector graphics)
- Dynamic NAL SEI support for HDR10/+ and other codec metadata (implied in ATSC3, but maybe not everywhere)

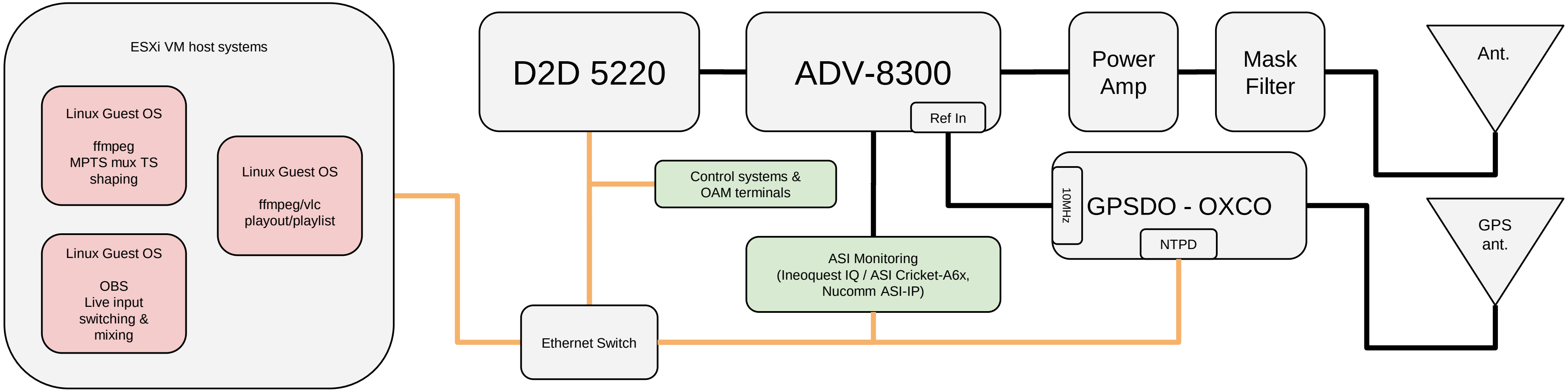
Present: What's Channel 3 up to?

K03IM-D Rundown

If it works on DVB-C, maybe it'll also work over ATSC 1.0

Just add: **PSIP, TVCT, EAS** switching

8VSB Bcast Lab, and Channel 3



Enter: D2D Tech

After semi-exhaustive research, determined D2D Technologies has best price/perf/hackability ratio* (Linux inside!) for muxing & "finishing" (ie. PSIP, EIT, ETT, other ATSC 1 adornments): https://d2dtechnologies.com/d2flex5220/

Got to know Steve Doll and Jessica Colyer, agreed to "try supporting" more codecs in VBR modes

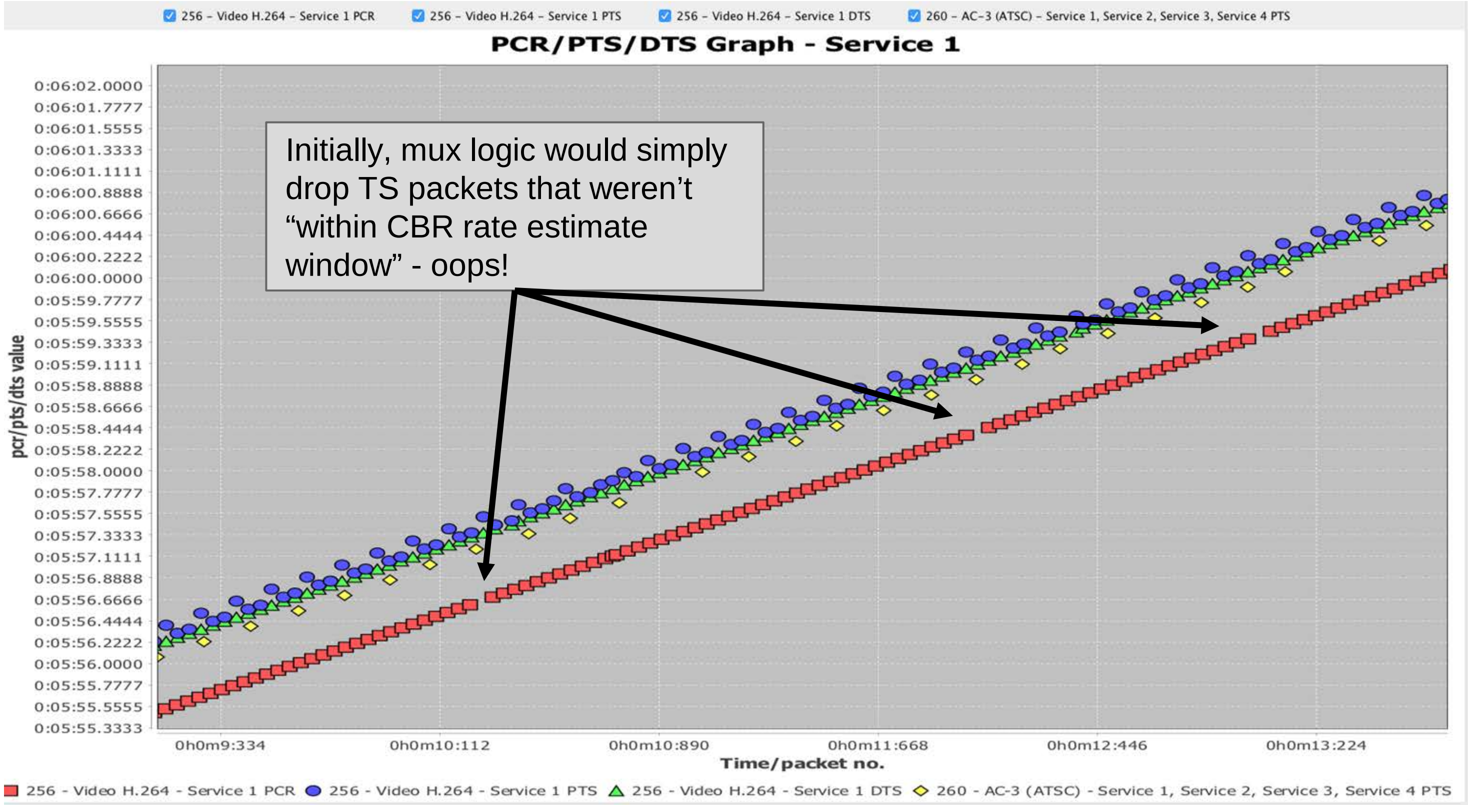
With mux hardware in hand, I:

1. Setup 8VSB/ATSC modulator & lab parts
2. Tested everything
3. Sent bug reports
4. Got new code
5. Goto #2

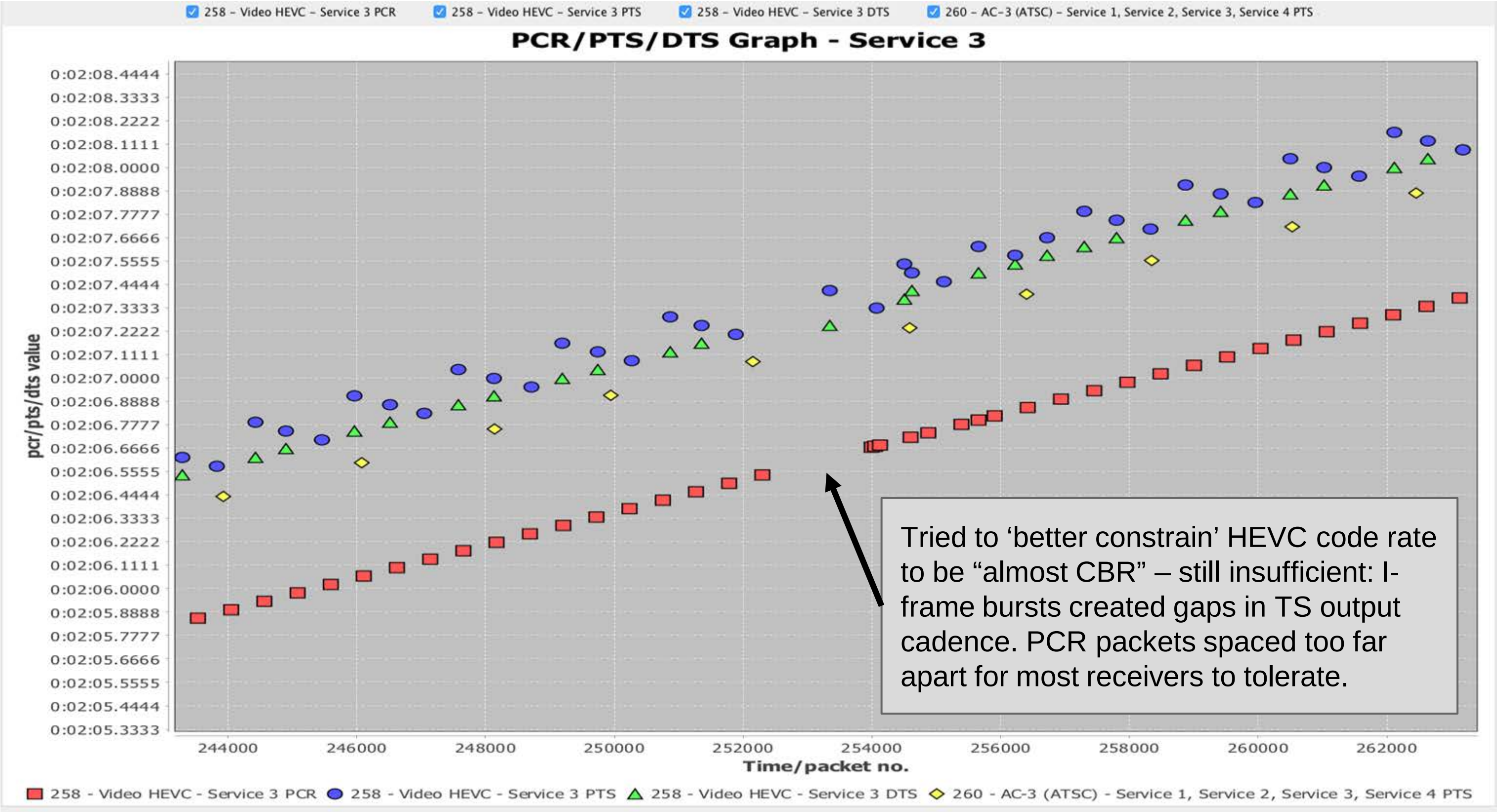
Caveats

- **NOTE**: The D2D 5220 mux is doing what FFmpeg, today in March 2024 cannot: generate PSIP stuff, *robustly* combine programs into a multi-program output TS composite, without "losing its mind" and halting all outputs when any single input is lost, complete lack of gapless live input switching/reversion, etc.
- FFmpeg alone cannot generate a ready-to-go ATSC 1.0 MPTS appropriate for OTA broadcast

Where we started - Broken VBR



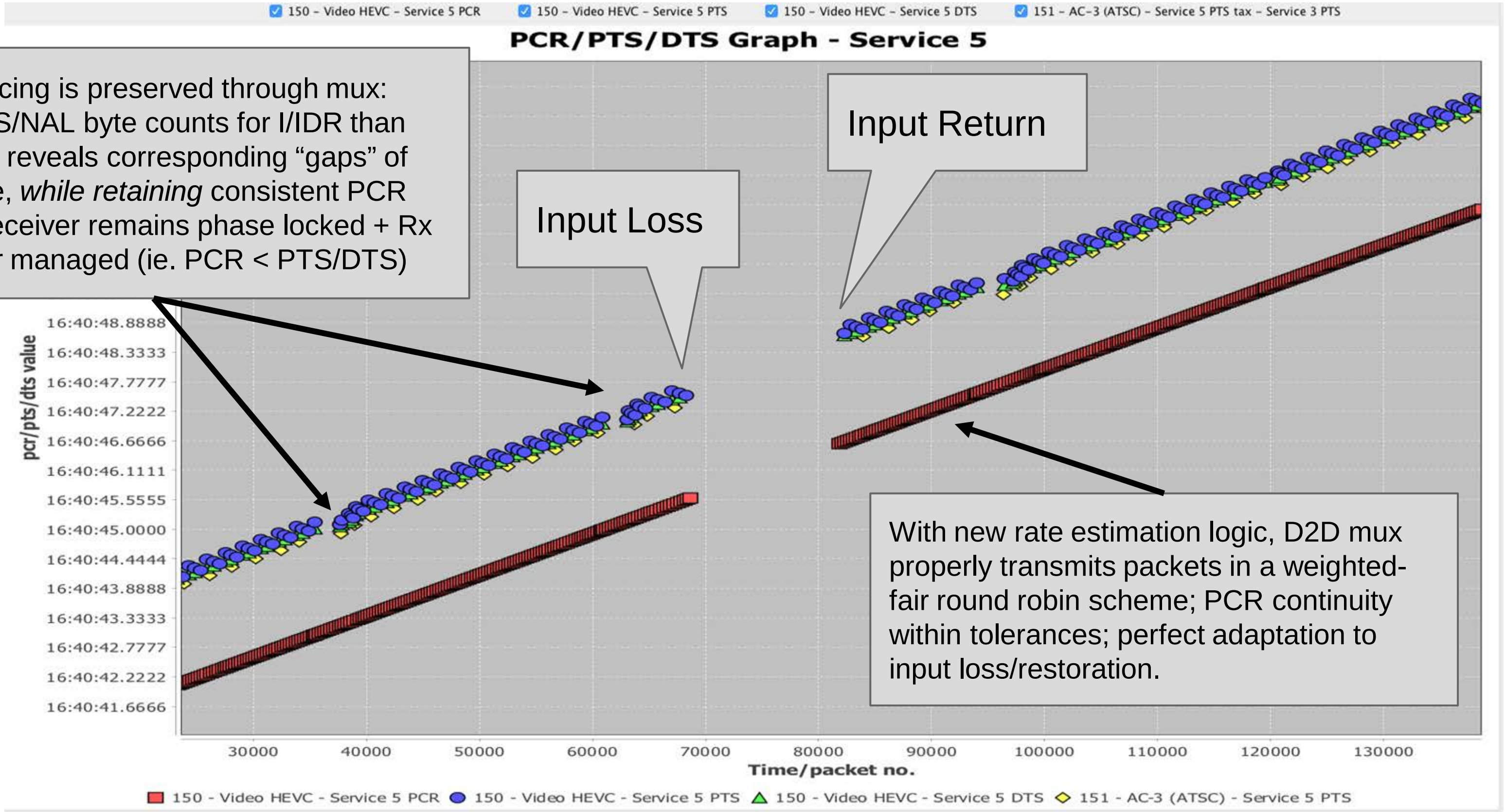
Where we started - Broken VBR handling



How to support VBR in MPTS?

- New (Thanks Steve!) rate estimation for input -> mux -> output routines
- We now *only* count bytes between PCRs (ie. out rate = sum input rate)
- Count TS packet bytes between TS with PCR, ignore input null padding, save system tick deltas between PCRs as a 64 bit integer: do math
  - robust min/max/mean/sum of squares vs mean
  - handily tolerates missing PCRs
- Output round-robin is 1/n portionally fair among buffered TS inputs
- Current algorithm now scales down to ~**18 kbits**/sec TS rates
  - Can now reliably mux ultra-low rate video, audio-only, and ancillary low-speed data programming
  - AAC ADTS 8 kb/s mode --> 23.1kbits/s over TS, now works
  - AC3 32 kb/s mode --> 45.1kbits/s over TS, also now works

Where we are now - Robust VBR



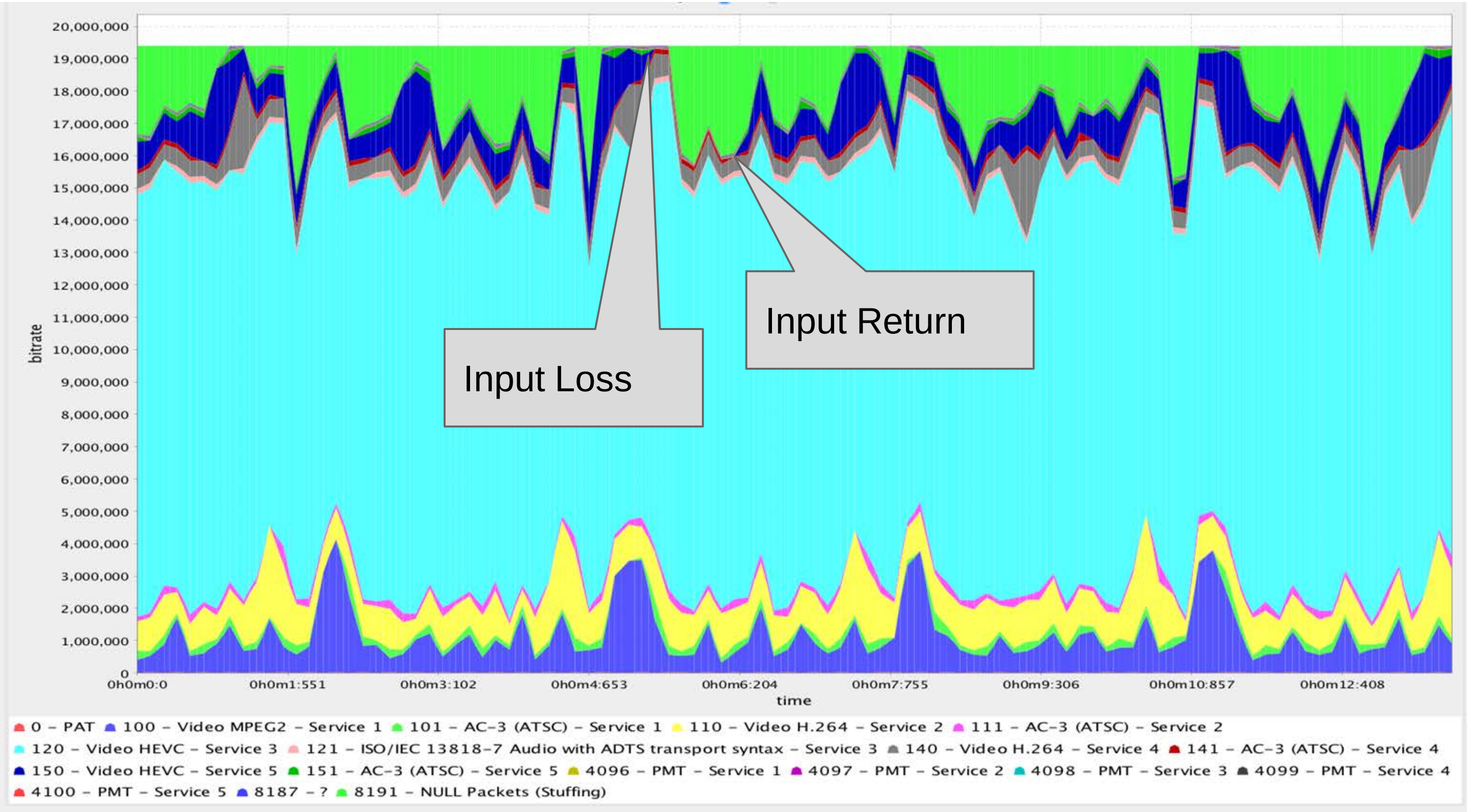
TS/PES pacing is preserved through mux: greater PES/NAL byte counts for I/IDR than B/P-frames reveals corresponding “gaps” of PTS in time, *while retaining* consistent PCR intervals: receiver remains phase locked + Rx slack buffer managed (ie. PCR < PTS/DTS)

Input Loss

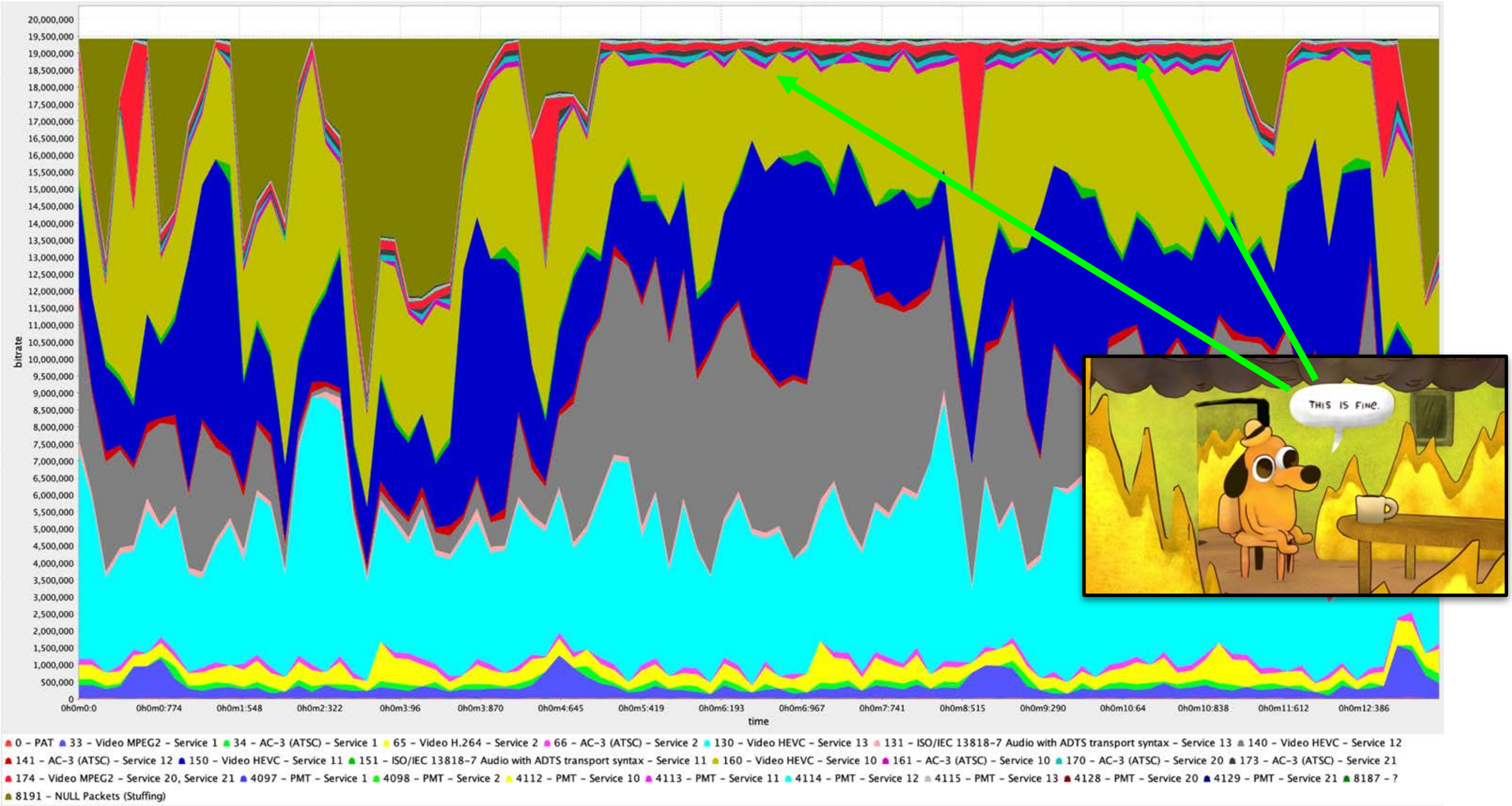
Input Return

With new rate estimation logic, D2D mux properly transmits packets in a weighted-fair round robin scheme; PCR continuity within tolerances; perfect adaptation to input loss/restoration.

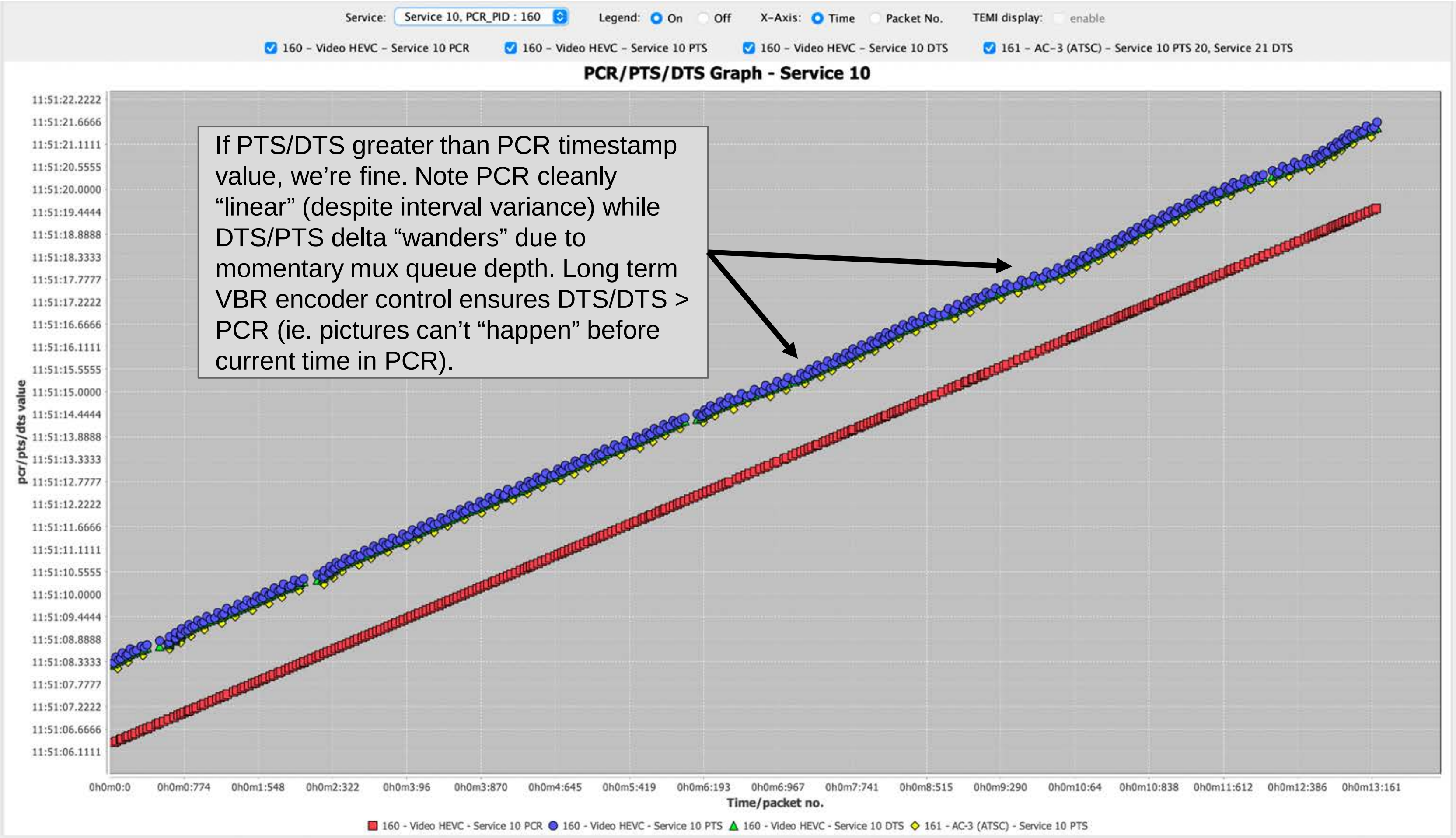
Where we are now - Robust VBR



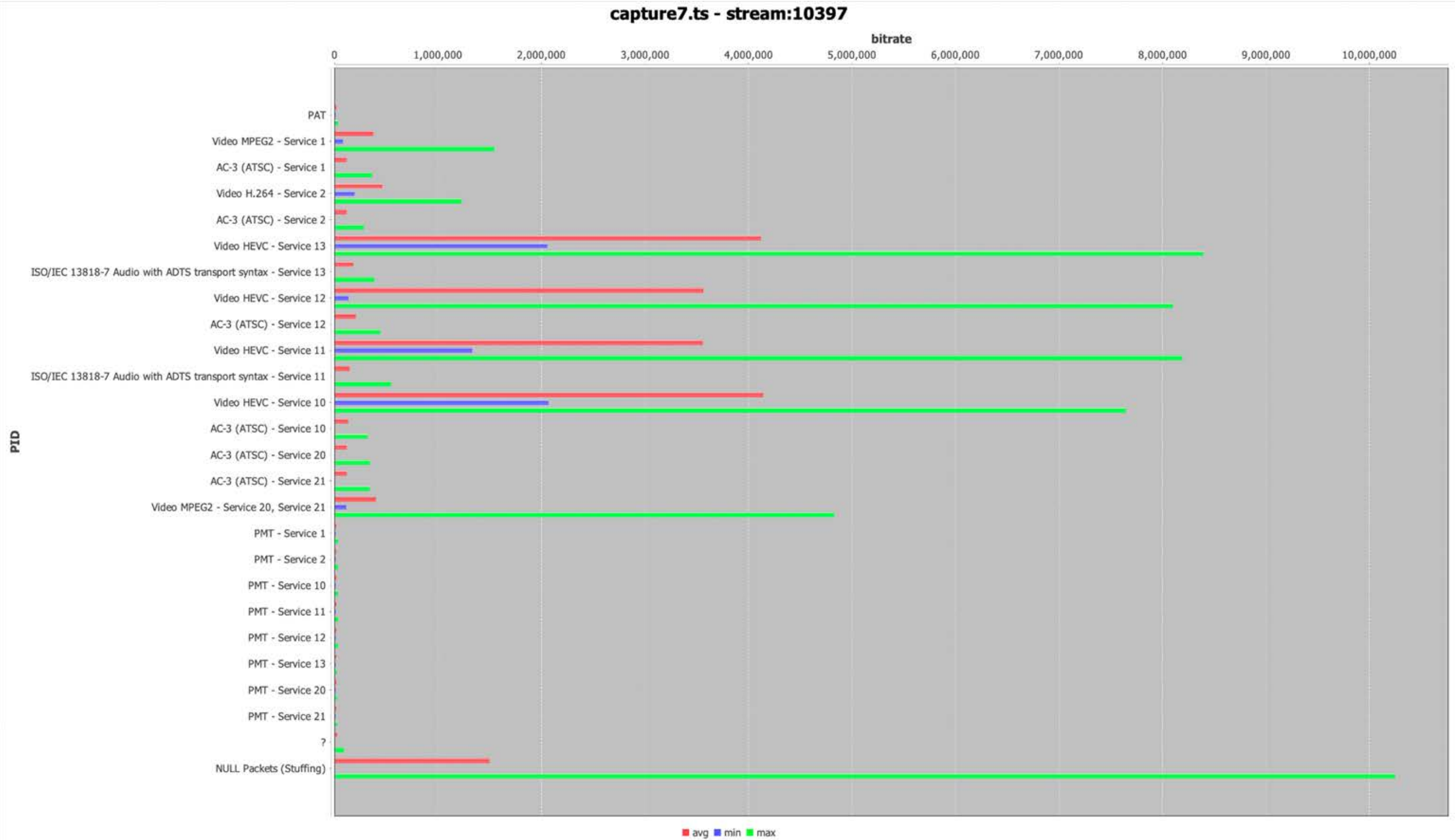
Where we are now - Robust When Maxed Out



Oversubscribed mux rate is fine, within limits



Where we are now - Maxed Out 4x UHD progs



Hurdle: Cleared, next:

Take this on the road:

- Landed on K03IM-D, Eugene, OR

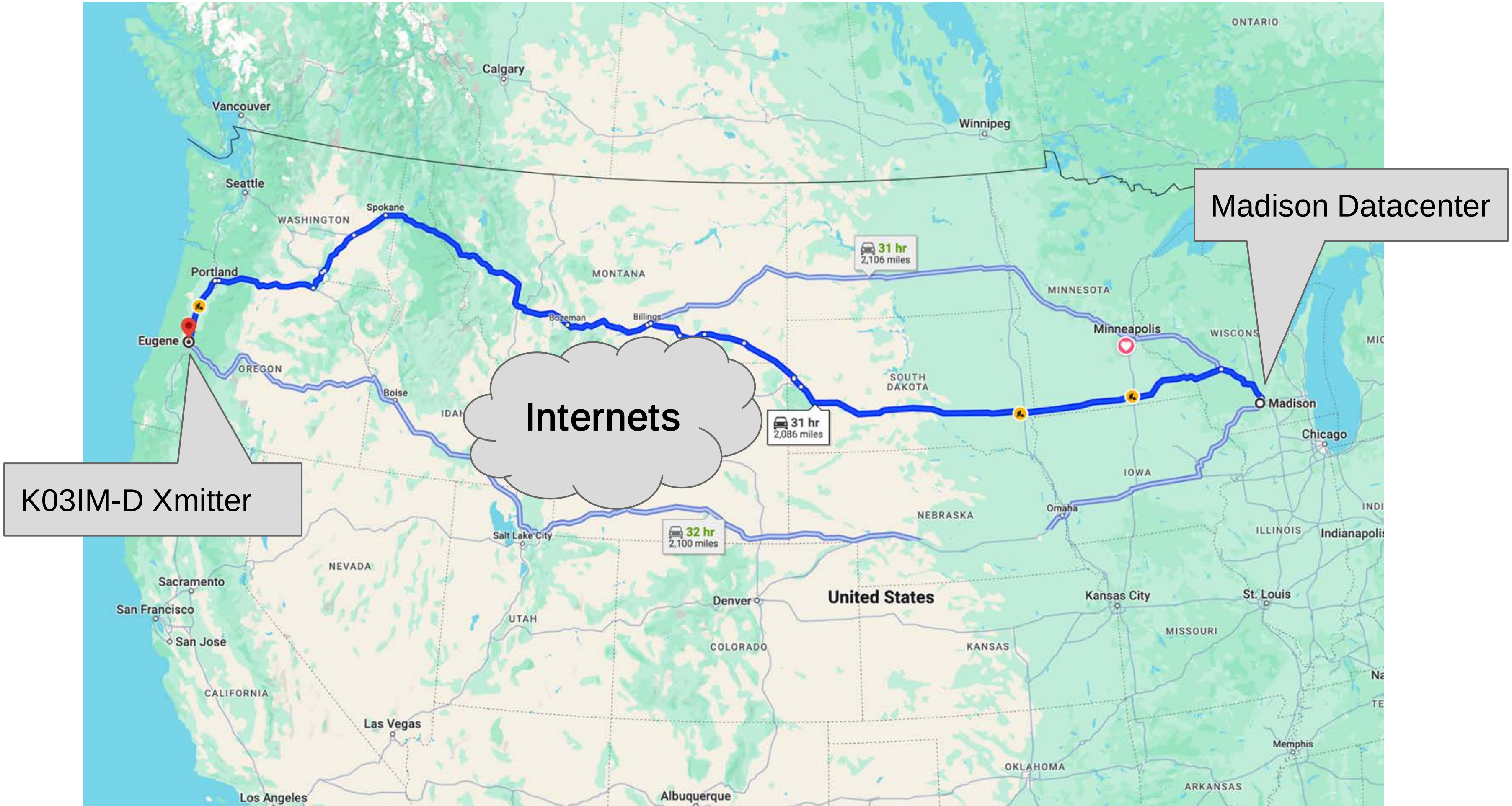
https://www.rabbitears.info/market.php?request=print_station&facility_id=185855

https://enterpriseefiling.fcc.gov/dataentry/public/tv/publicFacilityDetails.html?facilityId=185855

- Find content networks (TCN, Funroads, etc.), negotiate

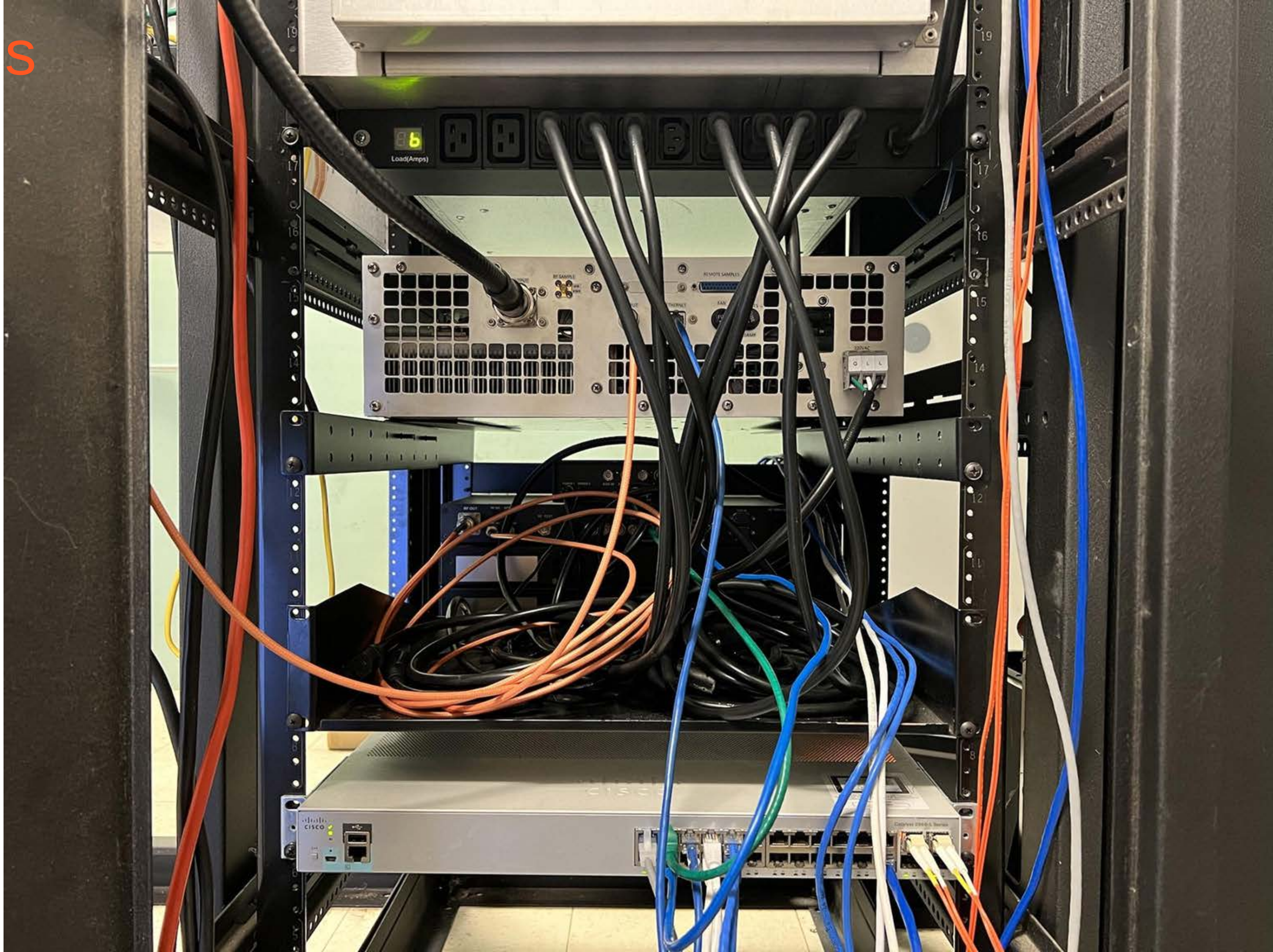
Channel 3 Eugene Photo Gallery: https://imgur.com/a/IyNAxlz

Program Flow: ~2100 miles, ~48.3 Msec RTT



Cool site pics

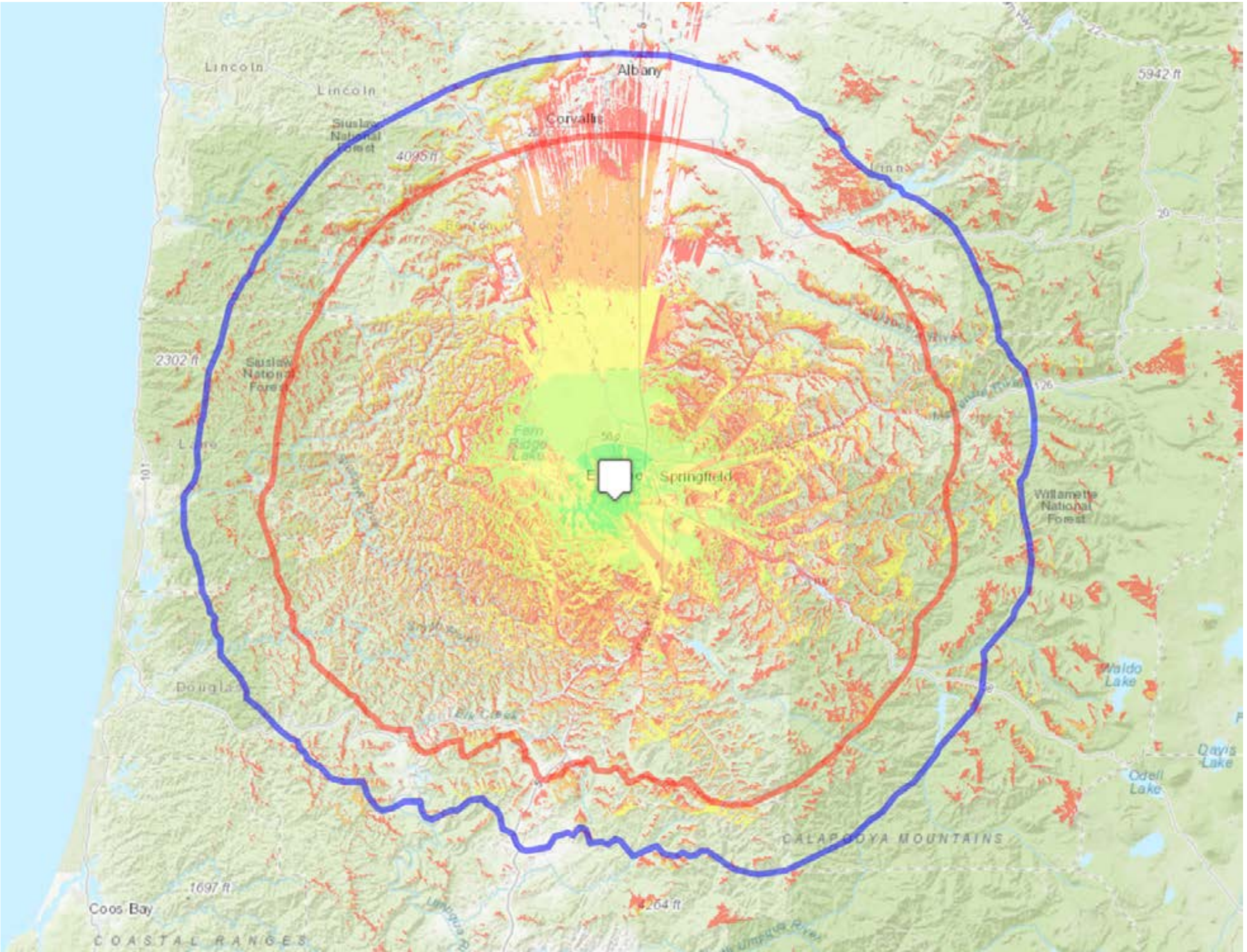
Cool site pics



Cool site pics

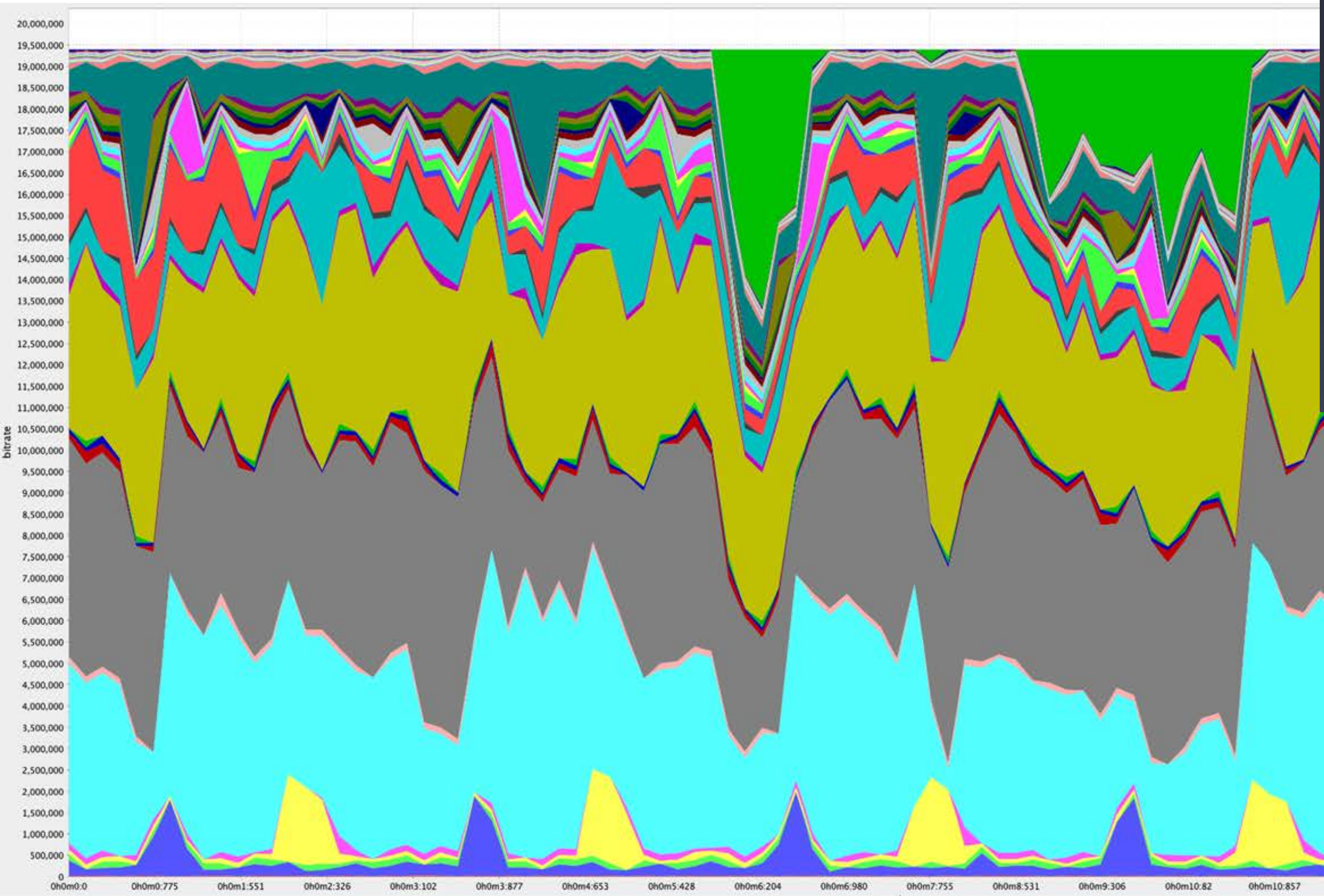
What about measurements/performance?

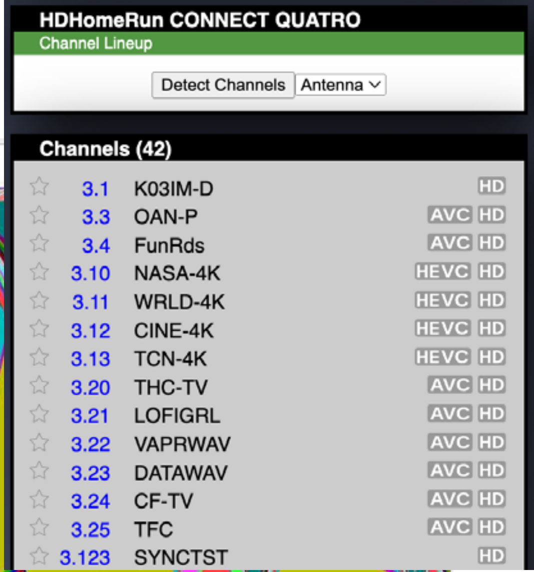
L/R & Contours



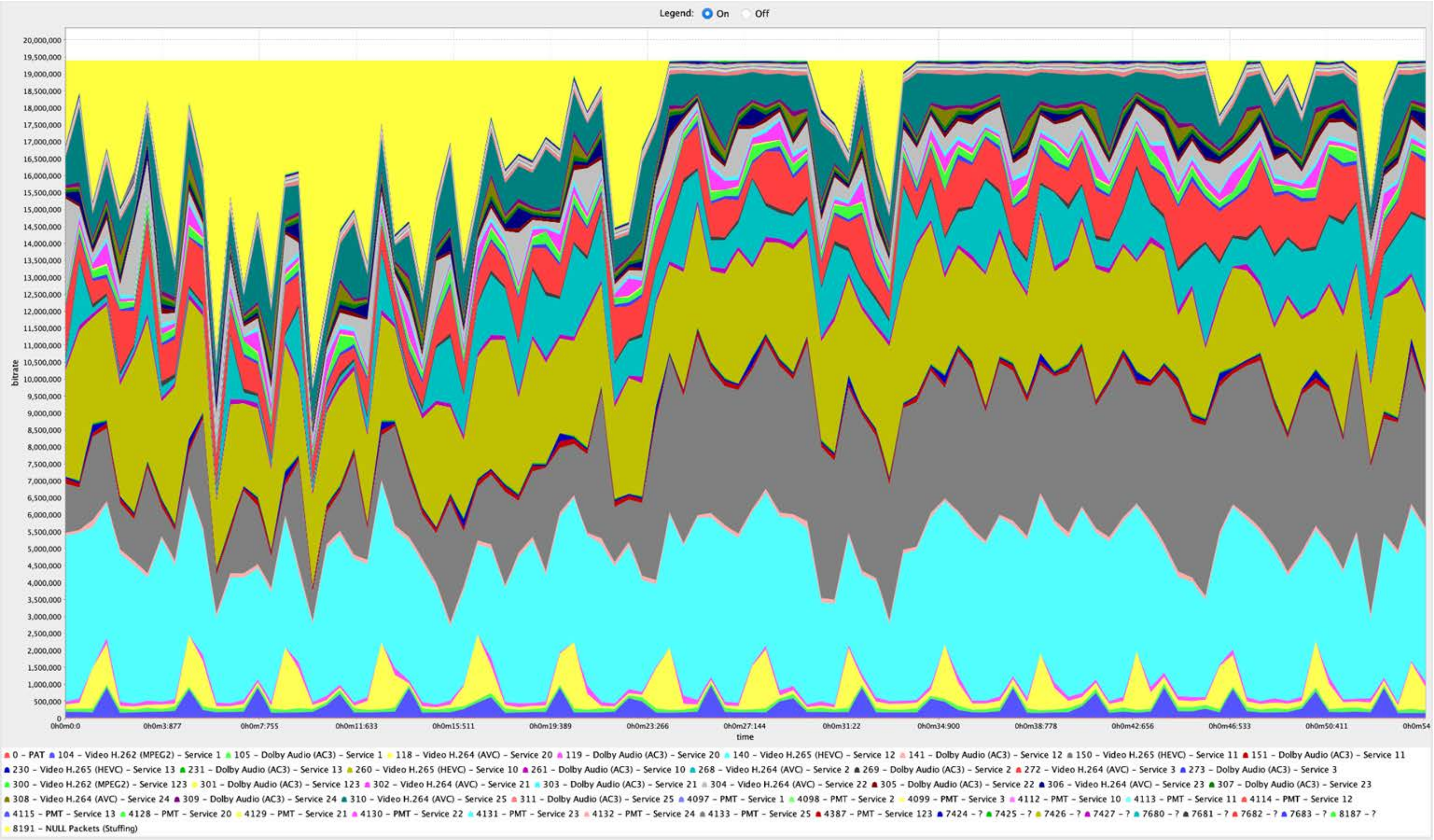
Albany
Corvallis
Lincoln
Lincoln
Siuslaw National Forest
4095 ft
5942 ft
Linn
2302 ft
Siuslaw National Forest
Fern Ridge Lake
Springfield
Willamette National Forest
Douglas
Waldo Lake
Davis Lake
Odell Lake
CALAPOOYA MOUNTAINS
1697 ft
Coos Bay
4264 ft
COASTAL RANGES

Mux Feb 2024, 14 programs

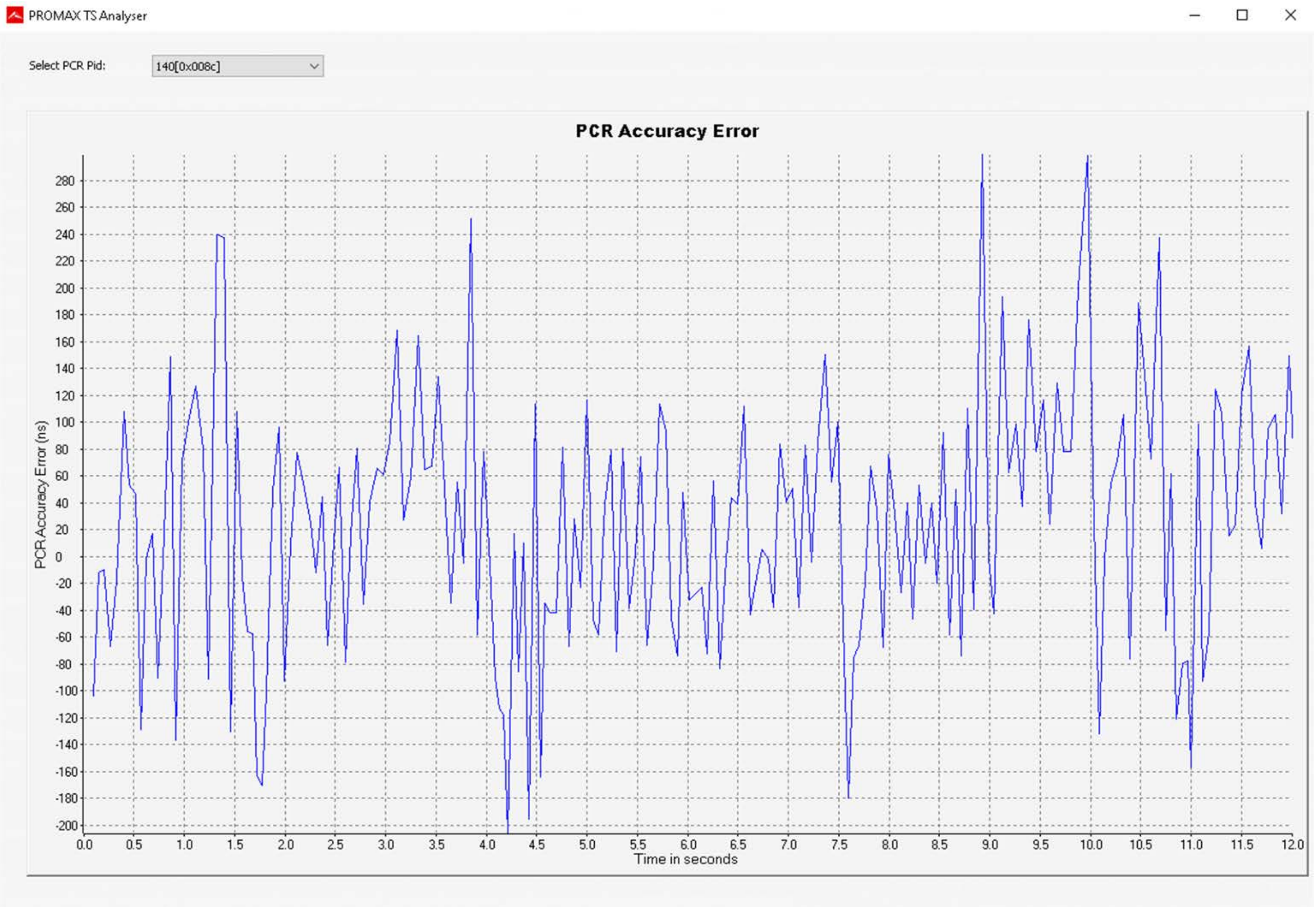




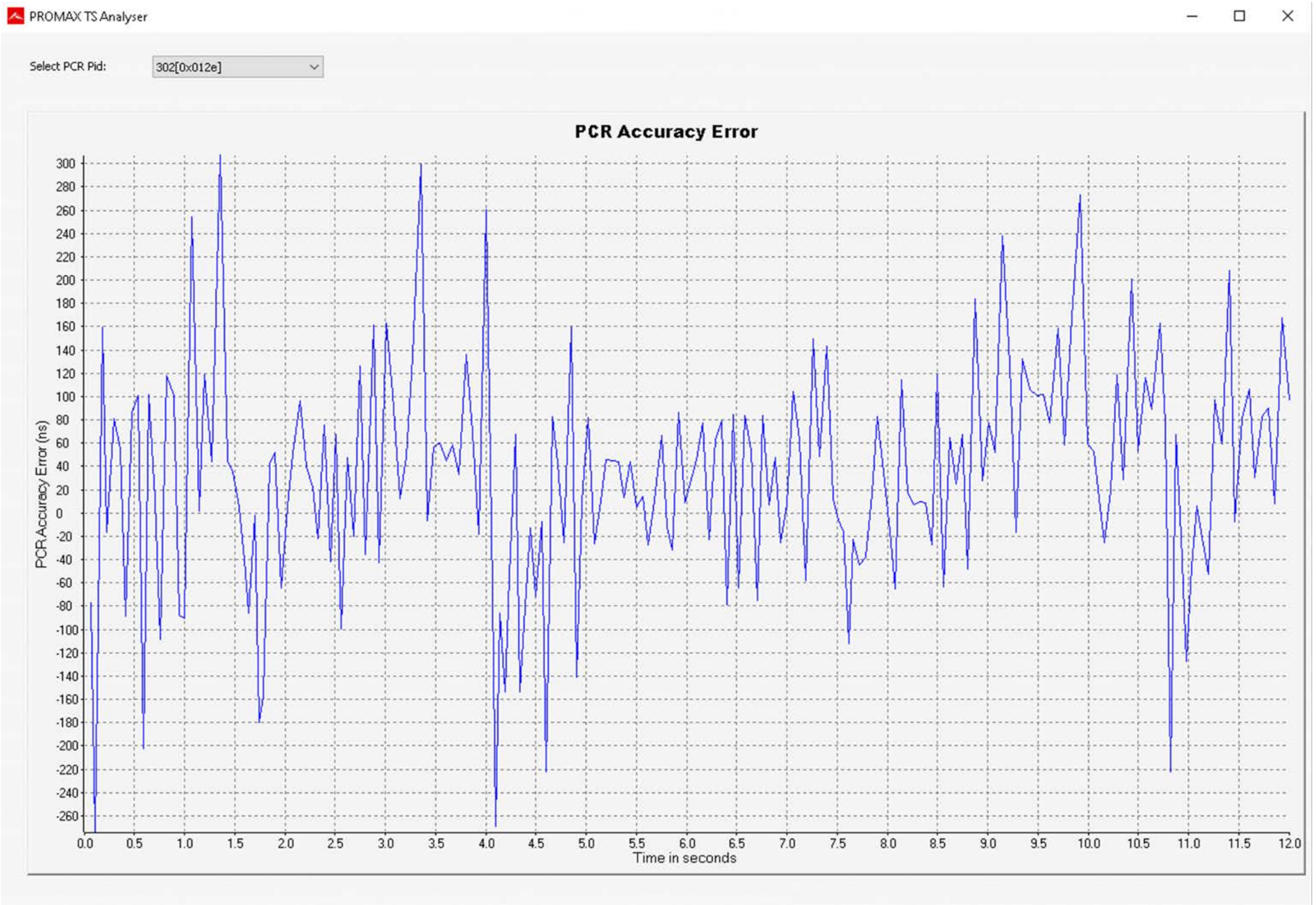
More Mux Pics



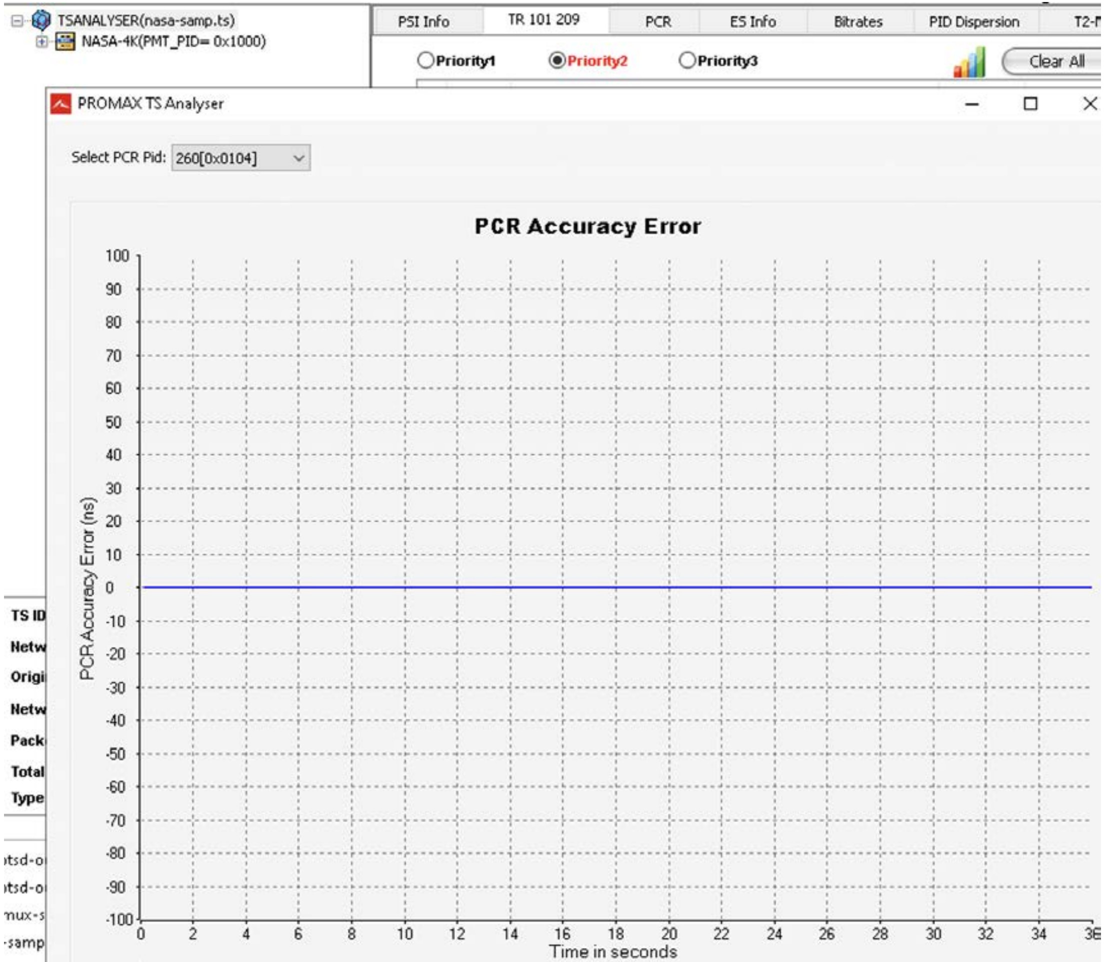
PCR Off Air Stats: Not bad!



PCR Off Air Stats: Not bad!



PCR From FFMPEG: Actually Perfect



Feedback from the field

Other subtle adjustments for “good display compatibility” with VBR-style AVC and HEVC:

- MPTS “transport buffering” burst-spreading is constrained by PCR to DTS/PTS delay - which is usually too short/small in most encoders & multiplexers, limiting VBR-ness
- Is the “DVB default” of ~750 msec *really* the limit? No: up to **~2000 msec** actually works on every HEVC display/decoder tested!
    - ~1700 msec seems more compatible with some AVC and MPEG2 video decoders
- Limiting HEVC reference frames - seems like 5 is workable everywhere, 8 for a few
- AVC decoders seem “stuck” at 4 refs, even on latest-gen UHD/HEVC-supporting TVs

Back To The Future!

What's next?

- Supporting Intermittent .2's, direct program bcast from cell phone -> air
  - Larix Broadcaster, OBS, others
- Ultra-dense music programming (HE-AACv2, EAC3), "slow TV"
- **Fast-flux** programming: subchannels may appear, disappear, change, at various timescales: minutes, hours, days
  - "Configuration as code"
  - Ad or Program Insertion On Steroids
- Datacasting integration: add support for Automatic Multicast Tunneling (AMT)
  - Engage "flash crowd" video offloading from Internet CDNs to broadcasters
  - We can **get paid** to haul IP packets over Multiprotocol Encapsulation (MPE) over TS packets!
- AI/ML methods to enhance AVC, HEVC, and other codecs
  - "Mostly real" - fully exhaustive comparison of all regions of all frames within a program, movie, etc.

Does the world really need *another* mux?

- In a word, yes
- Ground-up rethinking
- Written in Go
- Doing real-time stuff on non-real-time systems
- Channel 3 supports development
  - Spinoff to launch as Envelope Inc.

Questions?

tkapela@gmail.com

Appendix A

Semi-exhaustive FFMPEG notes & etc

```
while :; do srt-live-transmit -s 4000
"srt://source.host.net:1234?mode=caller&latency=2100&lossmaxttl=2000&&sndbuf
=10000000&oheadbw=100" udp://233.65.202.50:1234 ; sleep 1 ; done
```

RX stream from an SRT source, relay back out to mcast, output connection stats every 4000 msec, using fairly robust options - 2000 msec ARQ time to live, 2100 msec fixed sender/receiver latency, sender buffer allocation 10 megabytes, and overhead bandwidth of 100%

```
while :; do yt-dlp --socket-timeout 6 https://www.youtube.com/watch?v=xxxx -
f best -o - | mbuffer -q -W 32 -m 64k | ffmpeg -xerror -async 1 -re -i - -
map 0:0 -map 0:1 -c:a ac3 -ab 96k -cutoff 18000 -ac 2 -ar 48000 -vf
"minterpolate=fps=24000/1001:mi_mode=blend" -fps_mode cfr -c:v libx264 -g 72
-rc-lookahead 48 -preset veryslow -tune animation -b:v 400K -nal-hrd vbr -
maxrate:v 1.5M -bufsize 1.5M -me_method umh -me_range 64 -refs 4 -bf 4 -aq-
mode 3 -aq-strength 0.9 -qcomp .1 -flags +loop+qpel+cgop -map_metadata -1 -
map_chapters -1 -f mpegts -flush_packets 0
udp://233.65.202.201:1234?pkt_size=1316 ; sleep 1 ; done
```

Pulling from YT live source, conforming to 720p at lowest-possible bcastable frame rate of 23.976, low-bitate target

```
while :; do ffmpeg -i udp://233.65.202.201:1234 -c copy -f mpegts -
flush_packets 0 tcp://10.0.3.2:1423?pkt_size=3008 ; sleep 1 ; done
```

Listening to a mcast source, connecting to remote host via TCP, relaying A/V stream to target host

```
while :; do ffmpeg -fflags +genpts+discardcorrupt -copytb 0 -i
tcp://:1490?listen=1\&listen_timeout=10000\&timeout=20000000 -map 0:v -map
0:a -c:v copy -c:a copy -program title=Some Program Name:st=0:st=1 -f mpegts
-mpegts_start_pid 200 -flush_packets 0 -muxdelay 1.7 -muxpreload 4 -
max_interleave_delta 200000 -muxrate 2000320 -pcr_period 70
udp://10.0.3.19:1402?pkt_size=1316\&bitrate=2000320\&fifo_size=40000\&overru
n_nonfatal=1 ; sleep 1 ; done
```

Relaying a unicast TSoTCP stream in a CBR TS, to another TS mux, via unicast UDP

```
while :; do ./ffmpeg-6.1-amd64-static/ffmpeg -xerror -async 1 -threads:v 1 -i
udp://233.65.202.50:1234?fifo_size=200000\&overrun_nonfatal=1\&reuse=1\&timeout=900000000\
&buffer_size=10000000 \
-vf format=yuv444p10le,yadif=mode=0:deint=all,format=yuv420p,fps=fps=30000/1001 \
-fps_mode cfr \
-map 0:v:0 -map 0:a:0 \
-af "dynaudnorm=p=.25" \
-c:a ac3 -ac 2 -ab 96k -ar 48000 -cutoff 18000 \
-c:v libx264 -g 90 -preset veryslow -b:v 1M -maxrate:v 2M -bufsize 2M -refs 4 -bf 4 \
-aq-mode 3 -aq-strength 0.90 -b_qfactor 1.0 -b_qoffset 0.0 -qcomp 0.3 \
-dc 9 -subq 10 -weightp 2 -weightb 1 -bidir_refine 4 -mixed-refs 1 \
-8x8dct 1 -partitions all -direct-pred auto -nal-hrd vbr \
-rc-lookahead 30 -me_range 64 -me_method umh -trellis 2 -b_strategy 2 -b-pyramid 2 \
-fast-pskip 0 -flags +qpel+loop+cgop \
-intra_matrix
"8,8,9,9,10,10,11,11,8,9,9,10,10,11,11,12,9,9,10,10,11,11,12,12,9,10,10,11,11,12,13,13,10,
10,11,11,12,13,13,14,10,11,11,12,13,13,14,15,11,11,12,13,13,14,15,15,11,12,12,13,14,15,15,
16" \
-inter_matrix
"8,8,9,9,10,10,11,11,8,9,9,10,10,11,11,12,9,9,10,10,11,11,12,12,9,10,10,11,11,12,13,13,10,
10,11,11,12,13,13,14,10,11,11,12,13,13,14,15,11,11,12,13,13,14,15,15,11,12,12,13,14,15,15,
16" \
-f mpegts -max_interleave_delta 0 -flush_packets 0 udp://233.65.202.15:1234?pkt_size=1316
; sleep 1 ; done
```

Rx a slice-encoded (note: single decoder thread to avoid race conditions) 1080i program from mcast source, convert to 10 bit non-subsampled chroma, deinterlace/etc, convert back to yuv420 subsampled, force constant frame rate at 29.97, normalize audio level to ~-6 dBfs peak, various x.264 adjustments tuning encoder for “talking head” news programming; note custom quantizer matrix, output encoded stream towards mcast destination

```
IFS=$'\n' ; while :; do for i in `find "/mnt/space3/blah/" -name "*.mkv" -type f |shuf` ; do \
./ffmpeg-6.1-amd64-static/ffmpeg -fflags +discardcorrupt \
-ec guess_mvs+deblock+favor_inter -err_detect ignore_err \
-readrate_initial_burst 4 -async 1 -re -i "$i" \
-fps_mode cfr -map 0:v -map 0:a \
-ar 48000 -c:a eac3 -b:a 192k -cutoff 18000 -ac 6 -channel_layout "5.1" \
-color_primaries 1 -color_trc 1 -colorspace 1 \
-c:v libx264 -g 72 -preset veryslow -b:v 0.9M -maxrate:v 2M -bufsize 2M -refs 4 -bf 4 \
-psy-rd 1.1:0.5 -deblock -5:-3 -tune grain -b_qfactor 1.0 -b_qoffset 0.0 \
-dc 9 -subq 10 -weightp 2 -weightb 1 -aq-mode 3 -aq-strength 0.80 -qcomp 0.4 \
-mixed-refs 1 -8x8dct 1 -partitions all -direct-pred auto -nal-hrd vbr \
-rc-lookahead 36 -me_range 64 -me_method umh -trellis 2 -b_strategy 2 \
-b-pyramid 2 -bidir_refine 2 -fast-pskip 0 \
-flags +qpel+loop \
-intra_matrix
"8,8,8,9,8,9,11,11,11,11,11,11,13,12,13,13,13,13,13,13,13,13,13,13,13,14,14,14,17,17,17,13,13,13,14
,13,15,15,16,16,17,17,18,16,18,17,17,17,17,19,19,20,20,20,24,24,23,23,28,28,29,34,34,41" \
-inter_matrix
"8,8,9,9,10,10,11,11,8,9,9,10,10,11,11,12,9,9,10,10,11,11,12,12,9,10,10,11,11,12,13,13,10,10,11,11,
12,13,13,14,10,11,11,12,13,13,14,15,11,11,12,13,13,14,15,15,11,12,12,13,14,15,15,16" \
-vf "fps=24000/1001,format=yuv444p10le,atadenoise,setpts=PTS-STARTPTS,\
scale=iw*sar:ih:flags=lanczos,setsar=1,\
scale=1440:1080:force_original_aspect_ratio=decrease:flags=lanczos,\
pad=1440:1080:-1:-1:color=black,setsar=1,setdar=4/3,format=yuv420p" \
-af "asetpts=PTS-STARTPTS,volume=+3dB" \
-map_metadata -1 -map_chapters -1 \
-f mpegts -flush_packets 0 -shortest -mpegts_flags initial_discontinuity
udp://233.6.2.9:1090?pkt_size=1316 \
; done ; done
```

Pick a random file from a list of files matching a pattern, transcode this file at native playback rate, keeping multichannel audio intact (or conform to 6 chan/5.1 layout); conform all output to 1440x1080 at 4:3 aspect ratio, stop muxing when audio or video inputs have no more data (-shortest), to ensure output a/v pids of matching lengths; restart a/v PTS's from zero, and signal an explicit TS discontinuity to inform receivers that a new TS is starting

```
while :; do ./ffmpeg-6.1-amd64-static/ffmpeg -threads 16 -fflags +discardcorrupt -ec
guess_mvs+deblock+favor_inter -async 1 \
-i http://208.66.132.245:5004/auto/v15.1?overrun_nonfatal=1 \
-map 0:v:0 -map 0:a:0 \
-c:a ac3 -ac 2 -ab 96k -ar 48000 -cutoff 18000 \
-vf "format=yuv444p10le,yadif=mode=0:deint=all,fps=fps=30000/1001,\
hqdn3d=3:3:3:3,scale=720x480:flags=lanczos+accurate_rnd,\
hqdn3d=2:2:2:2,format=yuv420p" \
-fps_mode cfr \
-c:v mpeg2video -threads 16 -me_range 511 -intra_vlc 0 \
-tcplx_mask 0.3 -scplx_mask 0.2 -qcomp 0.06 \
-cmp satd -subcmp satd -mbcmp satd -dct int -idct int \
-qmin 2 -qmax 200 \
-trellis 0 -keyint_min 8 -g 60 -bf 3 -b_strategy 2 -brd_scale 2 -bidir_refine 2 -border_mask 2.0 \
-b:v 1.1M -maxrate 4M -bufsize 4M -dc 9 \
-intra_matrix "8,8,8,9,8,9,11,11,11,11,11,11,13,12,13,13,13,13,13,13,13,13,13,13,13,14,14,14,\
17,17,17,13,13,13,14,13,15,15,16,16,17,17,18,16,18,17,17,17,17,19,19,20,20,20,24,24,23,23,28,28,29,
34,34,41" \
-inter_matrix "8,8,8,8,8,8,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,10,10,10,10,10,10,10,\
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,11,11,11,11,11,11,11,11,11,12" \
-f mpegts -flush_packets 0 udp://233.65.202.151:1234?overrun_nonfatal=1\&pkt_size=1316 ; sleep 1 ;
done
```

Ultra-low-rate mpeg2 video encoding, targeting 1.1 mbits/sec for 16:9 720x480p - note for super-low rates, variable length coding table 0 (vlc) is recommended. We convert to non-subsampled 10 bit 4:4:4 chroma for deinterlacing, noise reduction, and scaling - ie. example assumes input is 1080i, lowrez progressive out. Max motion vector search area at 511 pixels. Set spatial and temporal masking to bias quantizer upwards for higher motion and higher complexity areas, leaving more bits for gradients/smooth areas. Less banding, more reduction on details that won't be visible to most viewers.