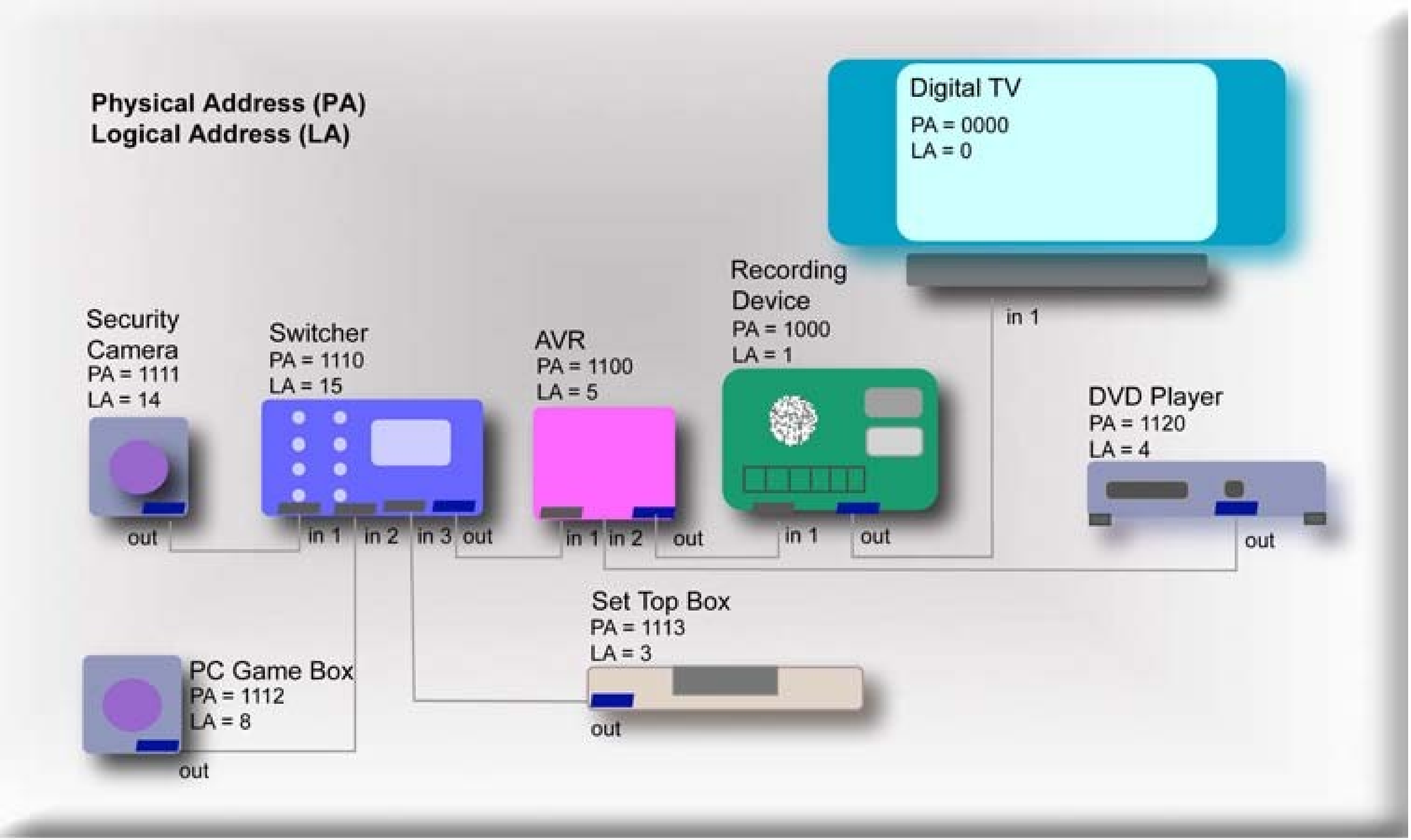
HDMI CEC (Consumer Electronics Control):

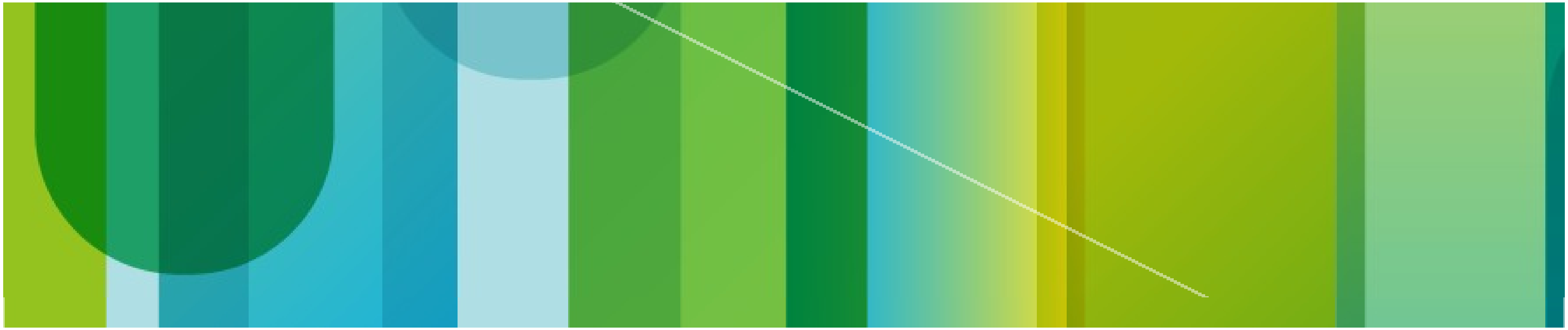
What? Why? How?

Hans Verkuil
Cisco Systems Norway



CEC @ 1 meter: 400 bits/s

What Is CEC?



Consumer Electronics Control

- An optional supplement to HDMI using pin 13 of the HDMI connector.
- Provides high-level control functions between the various audiovisual products in a user's environment.
- Based on the old AV.link scart standard (EN 50157-2-[123]).
- Implemented in HDMI receivers/transmitters and USB HDMI-passthrough devices.
- Data packets: 1 header byte + 0 to 15 data bytes.
- Very, very slow data rate ~400 bits/s.

Physical Address

Physical address: Hierarchy placement

Range:	0.0.0.0 - F.F.F.F
Root device:	0.0.0.0: HDMI Sink, typically a TV
First device:	1.0.0.0: Device connected to HDMI port 1 on root device
Second device:	2.0.0.0: Device connected to HDMI port 2 on root device
Third device:	1.1.0.0: Device connected to HDMI port 1 on first device

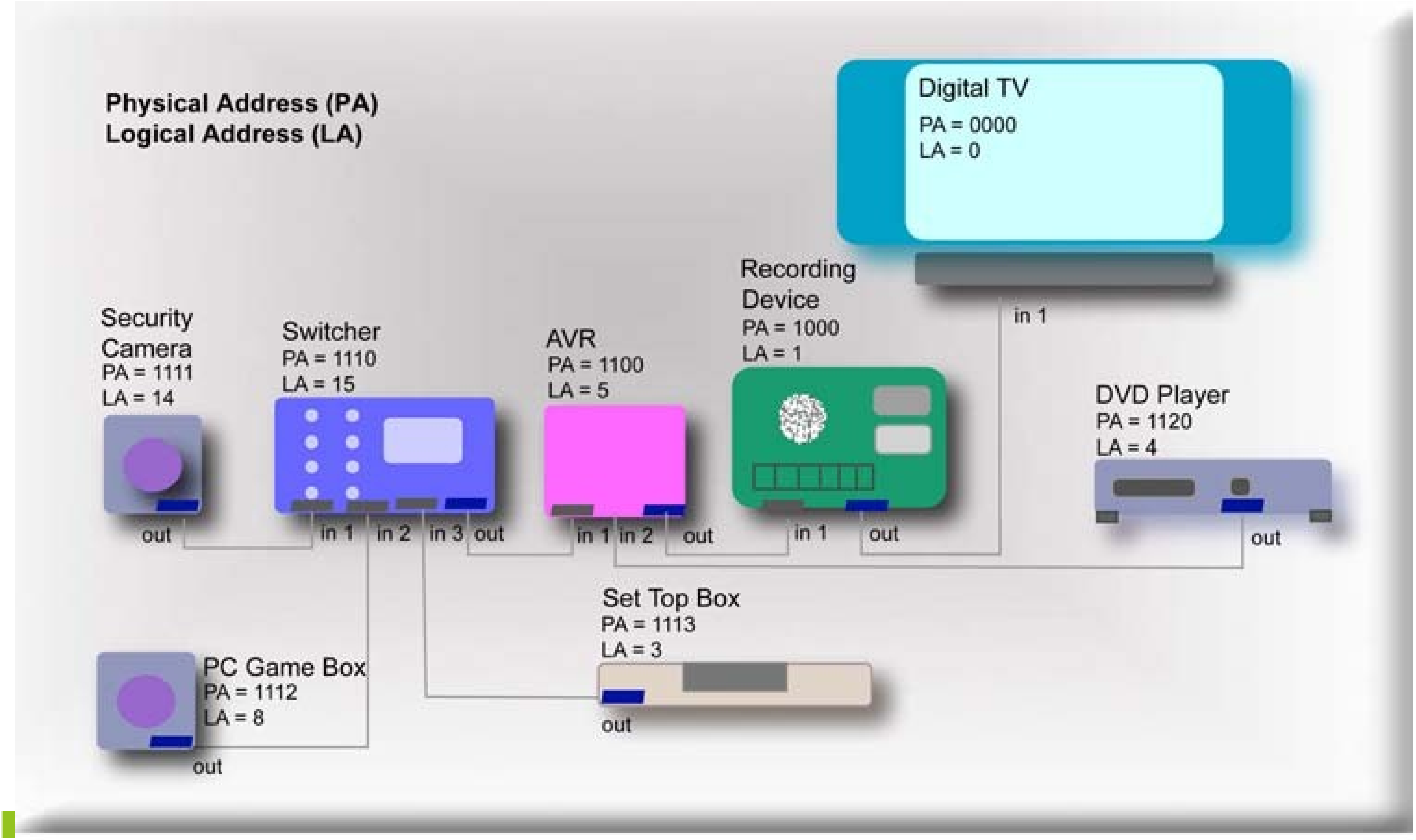
Sources get their Physical address from the EDID obtained from the sink.

Logical Address

Logical address 0-15: product type dependent
Not really an address, more like a nickname.

0: TV (root device)
1: Recording device 1
2: Recording device 2
3: Tuner 1
4: Playback device 1
5: Audio system
6: Tuner 2
7: Tuner 3
8: Playback device 2
9: Recording device 3
10: Tuner 4
11: Playback device 3
12: Backup 1
13: Backup 2
14: Specific use
15: Unregistered (as Initiator address)
Broadcast (as Destination address)

Topology Example



Trade Names

- Anynet+ (SAMSUNG)
- BRAVIA Link (Sony)
- Kuro Link (Pioneer)
- EasyLink (Philips)
- SimpLink (LG)
- VIERA link (Panasonic)

Why Implement CEC?

End-User Features

- One Touch Play: allows a device to be played and become the active source with a single button press.
- System Standby: enables the user to switch all devices to the Standby state with one button press.
- One Touch Record: offers a What You See Is What You Record (WYSIWYR) facility, meaning that whatever is shown on the TV screen is recorded on a selected Recording Device.
- Timer Programming: allows the user to program the timers in a Recording Device from an EPG running on a TV or STB.
- Deck Control: enables a device to control (e.g. play, fast forward etc.) and interrogate a Playback Device (a deck).

End-User Features

- Tuner Control: allows a device to control the tuner of another device.
- Device Menu Control: enables a device to control the menu of another device by passing through user interface commands.
- Remote Control Pass Through: enables remote control commands to be passed through to other devices within the system.
- System Audio Control: allows an Audio Amplifier / Receiver to be used with the TV. The volume can be controlled using any the remote controls of any suitably-equipped devices in the system.

Supporting Features

- Device OSD Name Transfer: enables devices to upload their preferred OSD name to the TV. The TV can then use this name in any menus associated with that device.
- Device Power Status: allows the current power status of a device to be discovered.
- OSD Display: enables a device to use the on-screen display of the TV to display text strings.
- Routing Control: allows the control of CEC Switches for streaming of a new source device.
- System Information: queries the system to determine device addresses and language.
- Dynamic Audio Lipsync: used by sinks to announce their audio latency.

Supporting Features

- Vendor Specific Commands: allows a set of vendor-defined commands to be used between devices of that vendor.
- Audio Rate Control: allows an Amplifier to fractionally increase or decrease the playback rate of an audio source.
- Audio Return Channel Control: controls the Audio Return Channel (ARC) part of the HDMI Ethernet and Audio Return Channel (HEAC).
- Capability Discovery and Control: controls HDMI Ethernet Channel (HEC) part of HEAC.

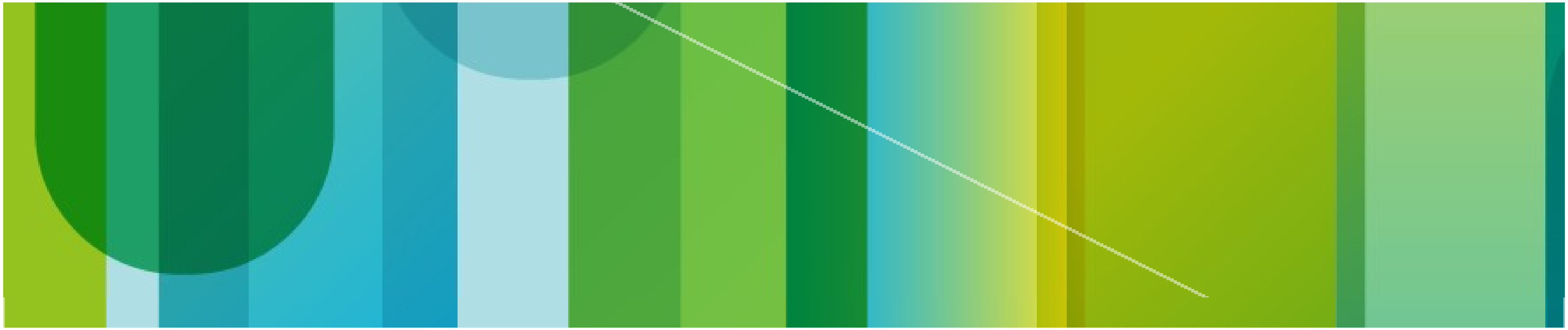
Capability Discovery and Control (CDC) for HEAC (HDMI Ethernet and Audio Return Channel)

- HEAC provides a full duplex connection between HDMI devices which conforms to 100Base-TX IEEE 802.3 Standard. This is defined as HDMI Ethernet Channel (HEC): very rarely implemented since wifi is used instead.
- HEAC provides audio data streaming which conforms to IEC 60958-1 standard from an HDMI Sink to an HDMI Source or repeater. This is defined as Audio Return Channel (ARC): much more common, used to e.g. transfer TV audio to an AV Receiver.
- For ethernet both the Utility (14) and Hot Plug Detect (19) lines are used. ARC can run in single mode (only pin 14 is used) or common mode (both pins are used).
- HEAC uses CDC for Capability Discovery and Control.
- If the Hot Plug Detect pin is used, then special CDC messages replace the hotplug signal.

Problems

- CEC is optional, so no guarantees.
- CEC version 1.4 leaves too much to the implementor, leading to inconsistent implementations. Version 2.0 is more strict.
- Ad-Hoc protocol: a clear example of a protocol where vendors hack something which then becomes part of the standard.
- Painfully slow and small payload.

How Is CEC Implemented?



Implementation Requirements

- CEC is highly asynchronous, blocking waits must be avoided.
- Replies to messages are out-of-order, not something userspace should have to deal with.
- It depends on the use-case which messages have to be handled by userspace or kernelspace, so both need to be supported.
- Creating and parsing messages should be standardized in a header that can be used both by userspace and kernelspace.
- Needs to support HDMI receivers (V4L2), transmitters (DRM, V4L2) and USB CEC dongles.

CEC Framework

- Creates a /dev/cecX device node.
- The driver determines the level of control userspace is allowed.
- Drivers implement the low-level CEC adapter operations.
- Drivers also implement high-level CEC functionality such as ARC and hotplug detect messages.
- The framework deals with the details of the protocol and the asynchronous aspects.
- The framework processes the core CEC messages automatically (unless userspace enables passthrough mode).
- The framework allows monitoring the CEC line.

CEC Framework

- cec-funcs.h contains static inline functions that fill and parse cec_msg structs.
- When transmitting a message you can asynchronously wait for a reply.

```
struct cec_msg {
        __u64 ts;
        __u32 len;
        __u32 status;
        __u32 timeout;
        __u32 sequence;
        __u8 msg[16];
        __u8 reply;
        __u8 reserved[35];
};

struct cec_msg msg;
cec_msg_init(&msg, 4, 0);
cec_msg_set_osd_string(&msg, CEC_OP_DISP_CTL_DEFAULT,
                       “Hello World”);
cec_msg_give_osd_name(&msg, true);
```

CEC Utilities

- cec-ctl supports all CEC messages (autogenerated code) and is a quick way of interactively configuring a CEC adapter and sending, receiving and monitoring CEC messages.
- cec-compliance will eventually do a proper CEC compliance test to see if the CEC implementation is correct (i.e. all messages that should be implemented are implemented).

CEC To Do

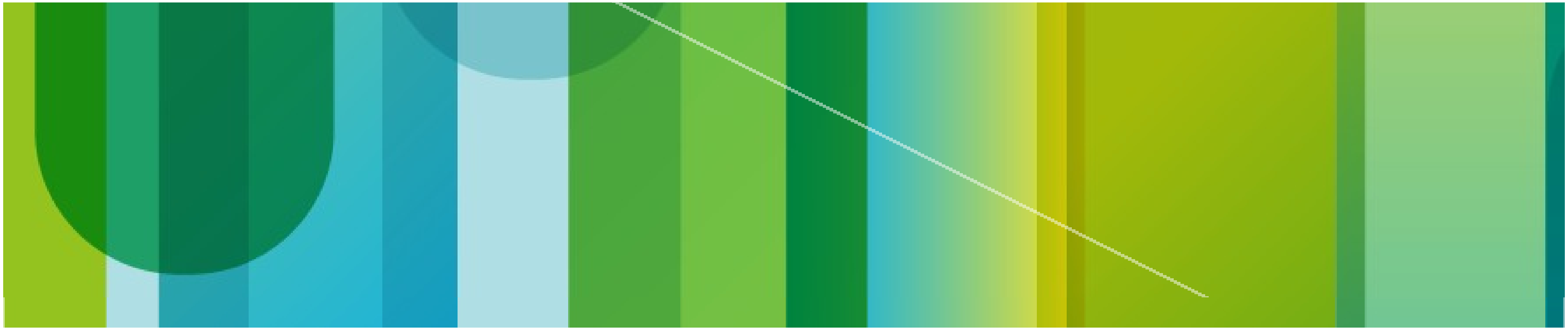
- What to do if a cable is disconnected and reconnected for a source: should the CEC adapter be disabled and re-enabled? Which means that you need to reconfigure. Or just update the physical address from the EDID (if necessary)? I think the latter, but I need to analyze this more closely.

Resources

Resources

- Git repository for the CEC framework: http://git.linuxtv.org/hverkuil/media_tree.git/log/?h=cec12
- Git repository for the cec-ctl/cec-compliance utilities: http://git.linuxtv.org/cgit.cgi/hverkuil/v4l-utils.git/log/?h=cec

Questions?



© 2014 Cisco and/or its affiliates. All rights reserved.