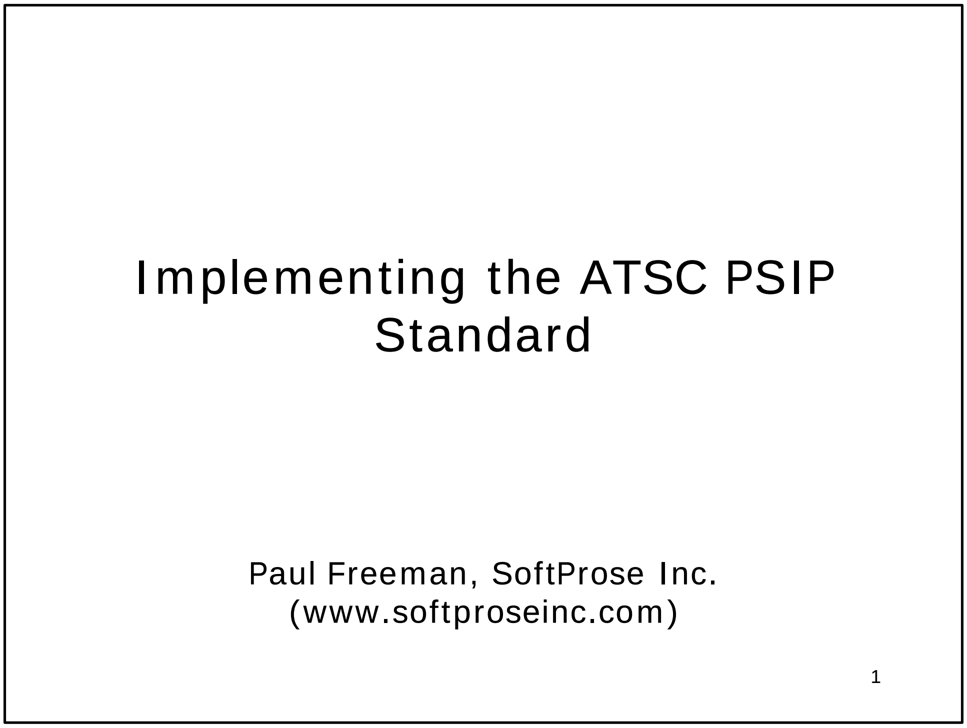
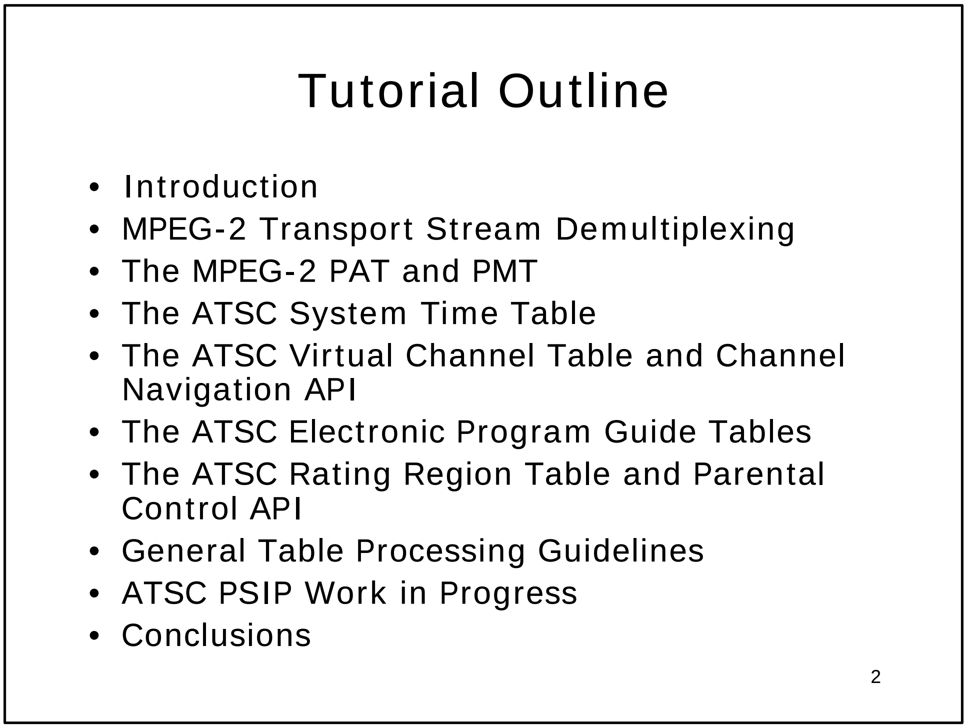


Implementing the ATSC PSIP Standard

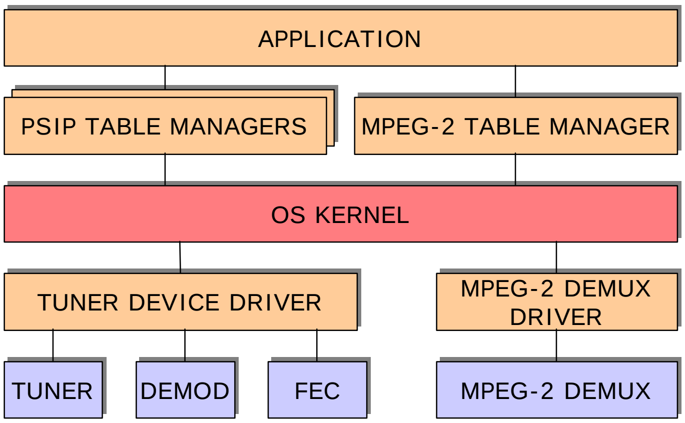
Paul Freeman, SoftProse Inc.
(www.softproseinc.com)



Tutorial Outline

- Introduction
- MPEG-2 Transport Stream Demultiplexing
- The MPEG-2 PAT and PMT
- The ATSC System Time Table
- The ATSC Virtual Channel Table and Channel Navigation API
- The ATSC Electronic Program Guide Tables
- The ATSC Rating Region Table and Parental Control API
- General Table Processing Guidelines
- ATSC PSIP Work in Progress
- Conclusions

Typical ATSC PSIP Software Block Diagram



APPLICATION

PSIP TABLE MANAGERS

MPEG-2 TABLE MANAGER

OS KERNEL

TUNER DEVICE DRIVER

MPEG-2 DEMUX DRIVER

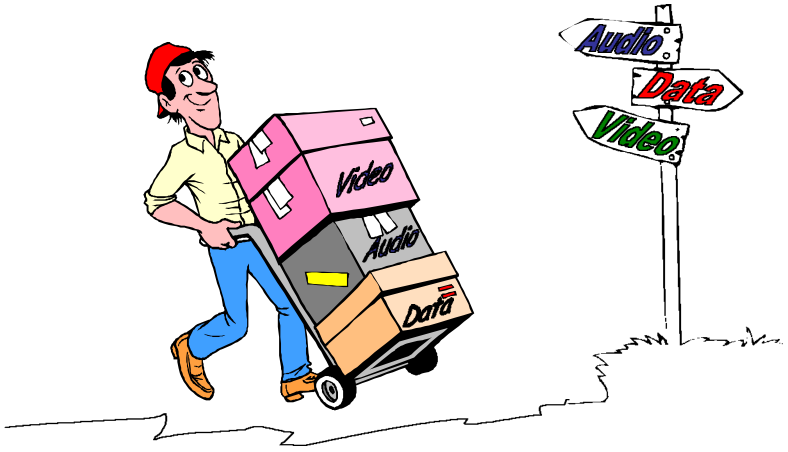
TUNER

DEMOD

FEC

MPEG-2 DEMUX

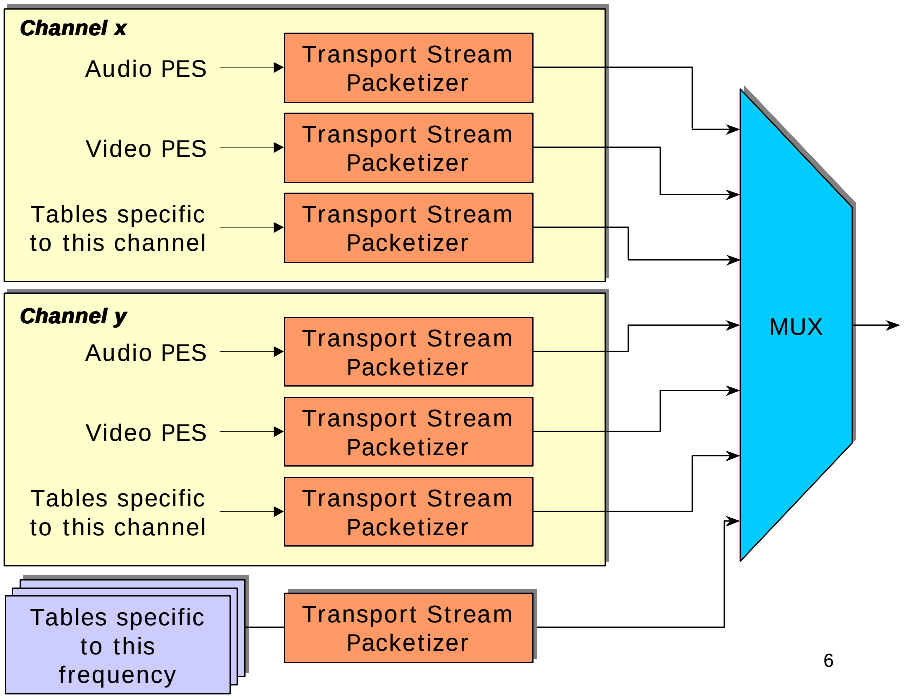
MPEG-2 Transport Stream Demultiplexing



The MPEG-2 Transport Stream

- Carries audio, video, and data for one or more channels from a single frequency
- Formatted as a multiplexed sequence of fixed size (188 byte) packets
- Each packet will contain data from one of the following sources:
  - A video packetized elementary stream
  - An audio packetized elementary stream
  - A table defined by the MPEG-2 standard
  - A table defined by the ATSC-PSIP standard

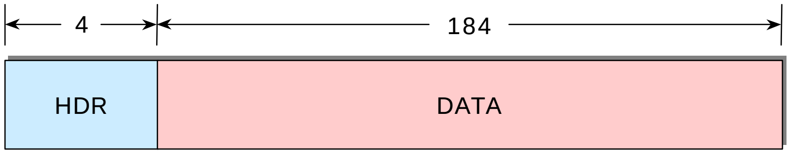
5



Channel x
Audio PES
Video PES
Tables specific to this channel
Transport Stream Packetizer
Transport Stream Packetizer
Transport Stream Packetizer
Channel y
Audio PES
Video PES
Tables specific to this channel
Transport Stream Packetizer
Transport Stream Packetizer
Transport Stream Packetizer
Tables specific to this frequency
Transport Stream Packetizer
MUX

6

Transport Stream Packets

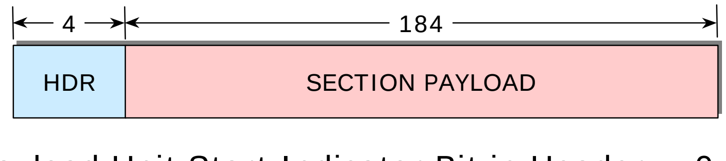


- Important fields in the packet header:
  - Sync byte - 8 bit fixed pattern at start of header
  - Packet ID (PID) - 13 bits used to identify type of packet
  - Payload Unit Start Indicator - 1 bit flag indicates a new PES packet or table section begins in this transport stream packet

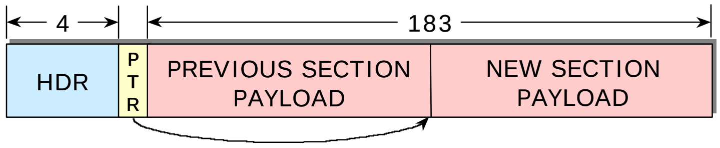
Table Packetization

- The MPEG-2 standard defines 3 tables, and the ATSC-PSIP standard defines 6.
- Each table is transmitted in one or more variable length “sections”.
- Each section is transmitted in one or more MPEG-2 transport stream packets.
- The MPEG-2 transport demultiplexor IC demultiplexes packets based on the packet ID.
- Sections from different tables may share the same packet ID.

The Payload Unit Start Indicator for Table Sections

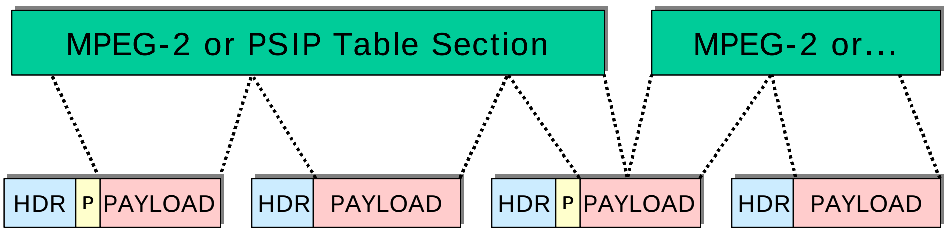


Payload Unit Start Indicator Bit in Header = 0



Payload Unit Start Indicator Bit in Header = 1

Payload Unit Start Indicator Example



Payload Unit Start Indicator = 1

Payload Unit Start Indicator = 0

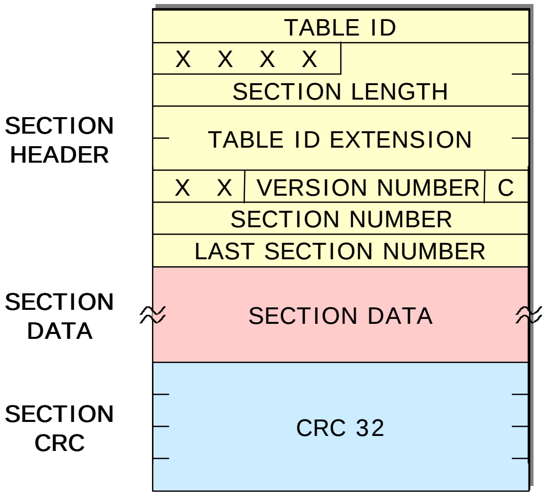
Payload Unit Start Indicator = 1

Payload Unit Start Indicator = 0

Table Demultiplexing Pseudo-code

```
lock to transport stream;
while (1)
{
    receive next packet;
    if (PID was requested)
    {
        if (payload unit start indicator == 1)
            found_section_boundary[PID] == TRUE;
        if (found_section_boundary[[PID])
        {
            add section data to buffer;
            if (complete buffer received)
            {
                if (section matches filter)
                    generate interrupt to CPU;
                else
                    reclaim buffer;
            }
        }
    }
}
```

Common Section Format



SECTION HEADER

SECTION DATA

SECTION CRC

TABLE ID

X X X X

SECTION LENGTH

TABLE ID EXTENSION

X X | VERSION NUMBER | C

SECTION NUMBER

LAST SECTION NUMBER

SECTION DATA

CRC 32

Section Filtering

- Tables are rebroadcast continuously.
- Reparsing the same table is a waste of resources.
- Upper layers must stay registered for most tables in order to detect version changes.
- Filtering sections by version number and table ID extension fields can help.

13

Typical MPEG-2 Transport Demultiplexor Driver API Calls

- select_video_PID();
- select_audio_PID();
- stop_video();
- stop_audio();
- establish_section_request();
- update_section_request();
- abort_section_request();
- others, for lip sync, etc.

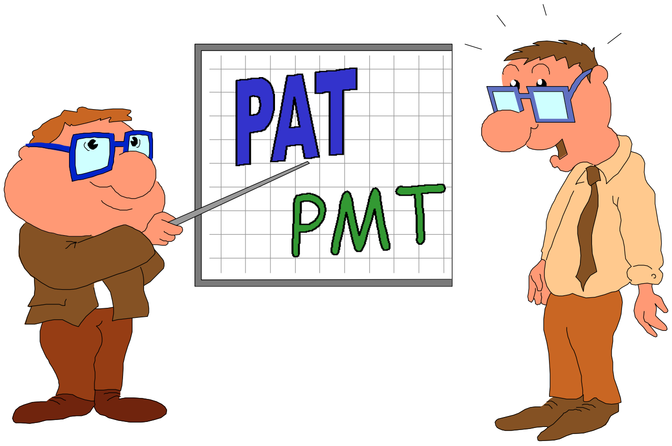
14

Desired Features for an MPEG-2 Transport Demultiplexor

- Ability to recognize section boundaries for MPEG-2 PSI and ATSC-PSIP tables
- Filtering capability on section headers
- Ability to specify a unique filter pattern for multiple tables sharing the same PID
- Optional CRC checking on section data

15

The MPEG-2 PAT and PMT



16

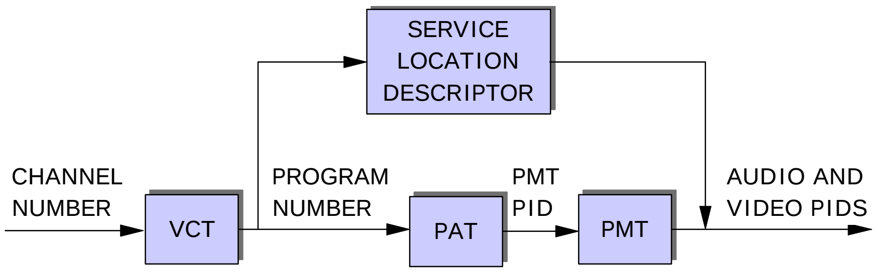
The Program Association Table

- Each ATSC “channel” corresponds to an MPEG-2 “program”.
- The Program Association Table provides the PID of the Program Map Table for each program in a transport stream.
- The Program Association Table is transmitted as the only table on a well known PID (PID zero).

The Program Map Table

- The Program Map Table defines the PIDs for each elementary stream associated with a program.
- Multiple audio streams may be defined. The receiver should check for a language descriptor.
- If the viewer changes language preferences, the PMT should be re-parsed to check for a matching audio elementary stream.

Selecting the Audio and Video PIDS



SERVICE LOCATION DESCRIPTOR

CHANNEL NUMBER

VCT

PROGRAM NUMBER

PAT

PMT PID

PMT

AUDIO AND VIDEO PIDS

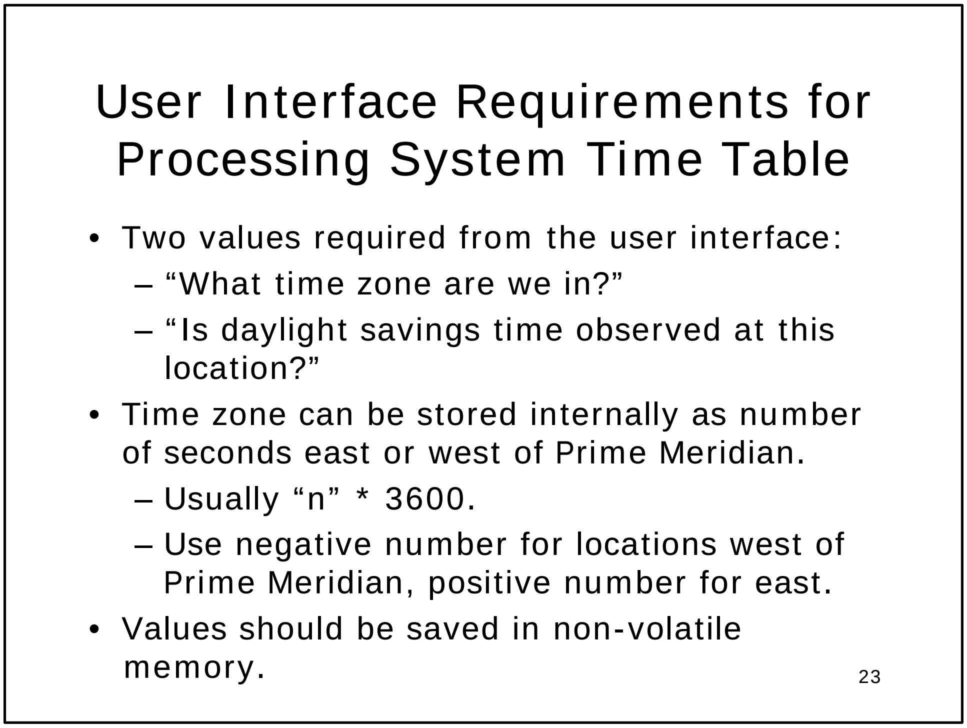
Continuous Monitoring of Tables

- Can the PMT PID, or the audio, video, or PCR PID change without changing channels?
  - – Some people say “no”.
  - – MPEG-2 specification does not seem to say.
- Our recommendation:
  - – Once a program is playing, stay registered for next version of PAT and PMT.
  - – If the PAT changes, see if the PMT PID has changed. If so, acquire a new PMT.
  - – If the audio, video, or PCR PID change, reprogram the MPEG-2 demux.

Typical MPEG-2 Table Manager API Calls

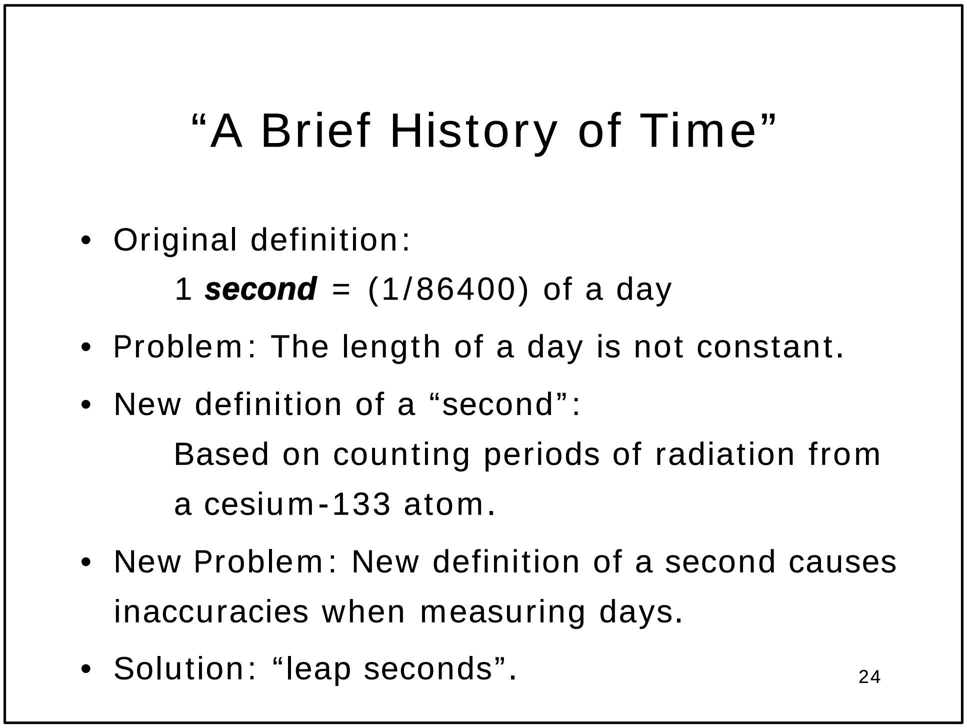
- set_program_number();
  - pass program number as parameter
  - MPEG-2 Table Manager will acquire and parse PAT and PMT to select PIDs
- set_pids();
  - pass audio, video, and PCR PIDs as parameters
- stop_decoding();
- set_language_defaults();
- get_language_defaults();
- get_languages();

The ATSC System Time Table



User Interface Requirements for Processing System Time Table

- Two values required from the user interface:
  - “What time zone are we in?”
  - “Is daylight savings time observed at this location?”
- Time zone can be stored internally as number of seconds east or west of Prime Meridian.
  - Usually “n” * 3600.
  - Use negative number for locations west of Prime Meridian, positive number for east.
- Values should be saved in non-volatile memory.

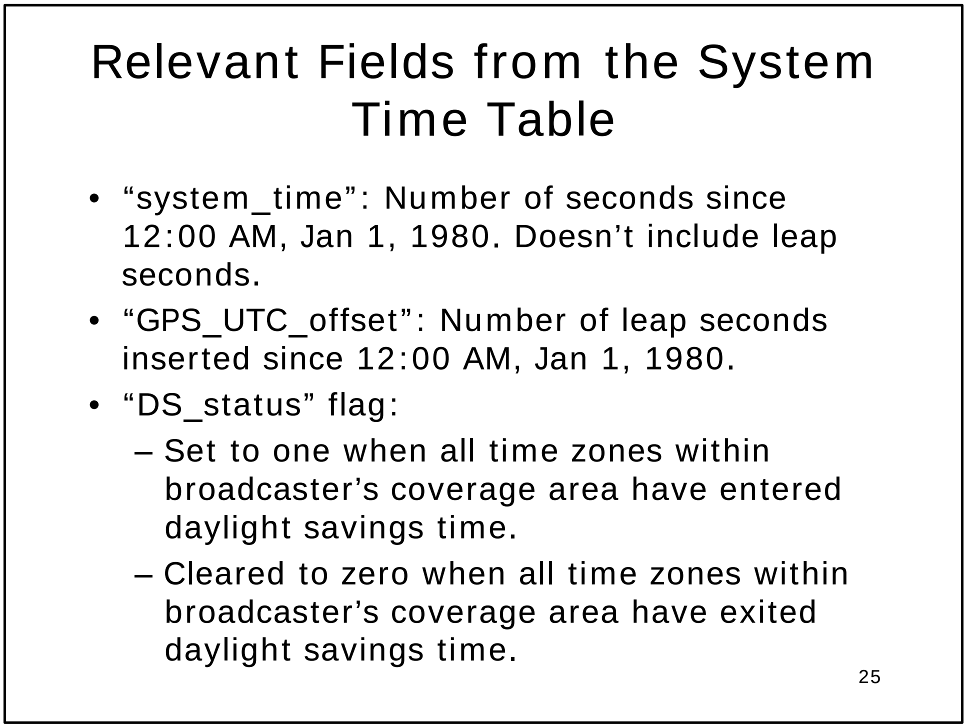


“A Brief History of Time”

- Original definition:

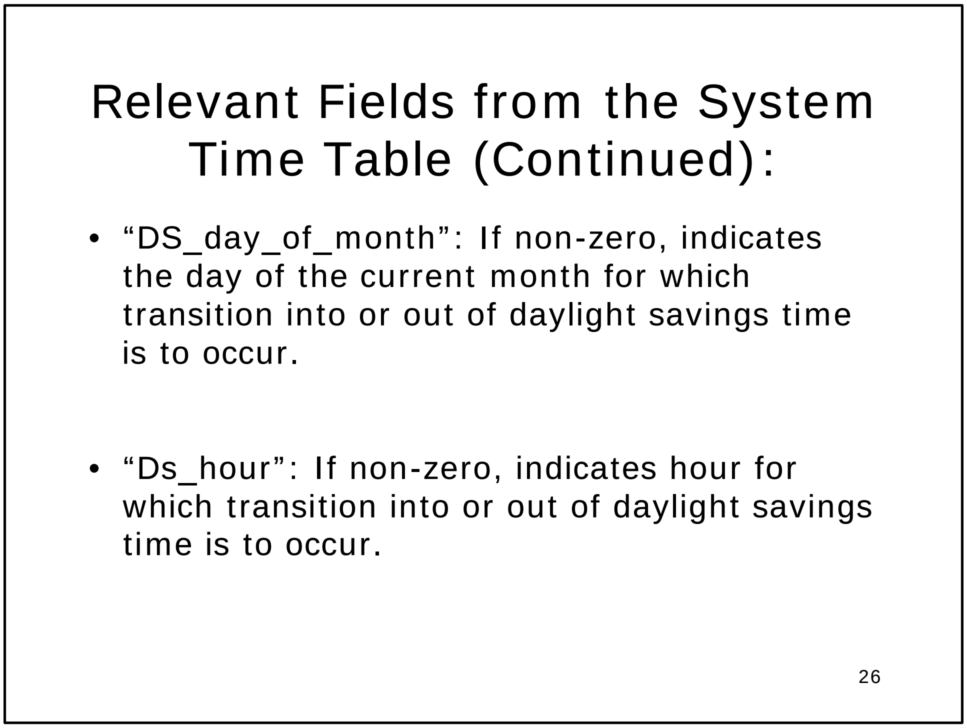
  1 ***second*** = (1/86400) of a day
- Problem: The length of a day is not constant.
- New definition of a “second”:

  Based on counting periods of radiation from a cesium-133 atom.
- New Problem: New definition of a second causes inaccuracies when measuring days.
- Solution: “leap seconds”.



Relevant Fields from the System Time Table

- "system_time": Number of seconds since 12:00 AM, Jan 1, 1980. Doesn't include leap seconds.
- "GPS_UTC_offset": Number of leap seconds inserted since 12:00 AM, Jan 1, 1980.
- "DS_status" flag:
  - Set to one when all time zones within broadcaster's coverage area have entered daylight savings time.
  - Cleared to zero when all time zones within broadcaster's coverage area have exited daylight savings time.



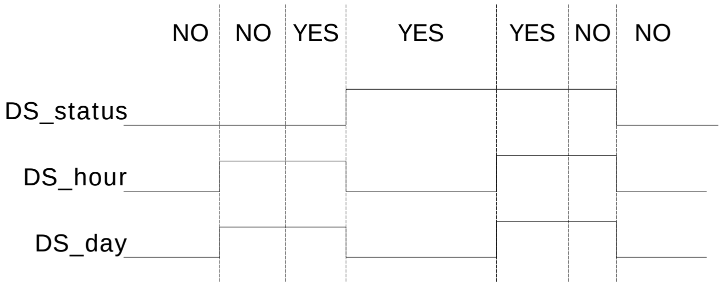
Relevant Fields from the System Time Table (Continued):

- "DS_day_of_month": If non-zero, indicates the day of the current month for which transition into or out of daylight savings time is to occur.
- "Ds_hour": If non-zero, indicates hour for which transition into or out of daylight savings time is to occur.

Calculating the Current Time from the System Time Table

- Obtain “system_time” field from STT.
- Subtract “GPS_UTC_offset” field from STT. The result is number of seconds since 12 AM, January 1, 1980. (In Greenwich, England).
- Add time zone adjustment value obtained from User Interface.
- Convert to date and time format. (Use “mktime()” if it is available in your C library).
- If daylight savings time is observed, check if we are in daylight savings time. (Explained later).
- If so, add 3600 to results from step 3, and reconvert to date/time format.

Daylight Savings Time Determination



Be careful: DS_status alone can’t reliably determine daylight savings time status.

Useful “C” Library Calls for Processing Time

- Most C environments define two encodings for time: “time_t” and “struct tm”.
  - time_t: encodes both date and time, typically in a single unsigned integer
  - struct tm: structure containing date and time fields
- Several calls are typically provided for switching between these two formats (see next page).
- For some calls, you need to have a “TZ” environment variable set correctly.

Useful “C” Library Calls for Processing Time (Continued)

- ctime(); Convert “time_t” to string format
- difftime(); Return difference (in seconds)

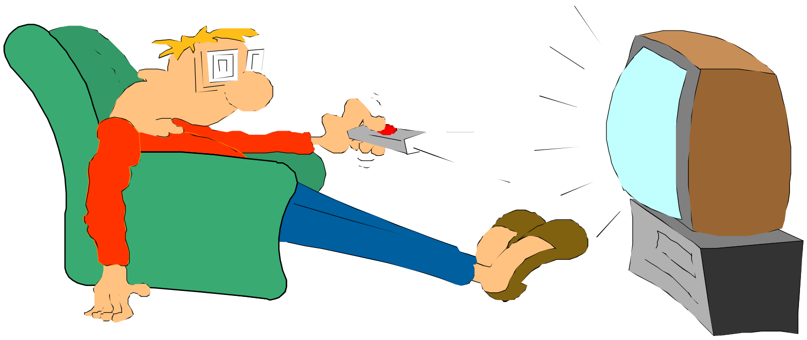
  between two “time_t” types
- gmtime(); Convert local time (expressed

  as “time_t”) to GMT (expressed as “struct tm”)
- localtime(); Convert “time_t” to “struct tm”
- mktime(); Convert “struct tm” to “time_t”
- strftime(); Convert “struct tm” to

Typical API Calls for System Time

- set_time_zone();
- set_dst_observance();
- get_broadcaster_time();
- enable_time_updating();
- disable_time_updating();
- request_notification_at_time();

The ATSC Virtual Channel Table and Channel Navigation API

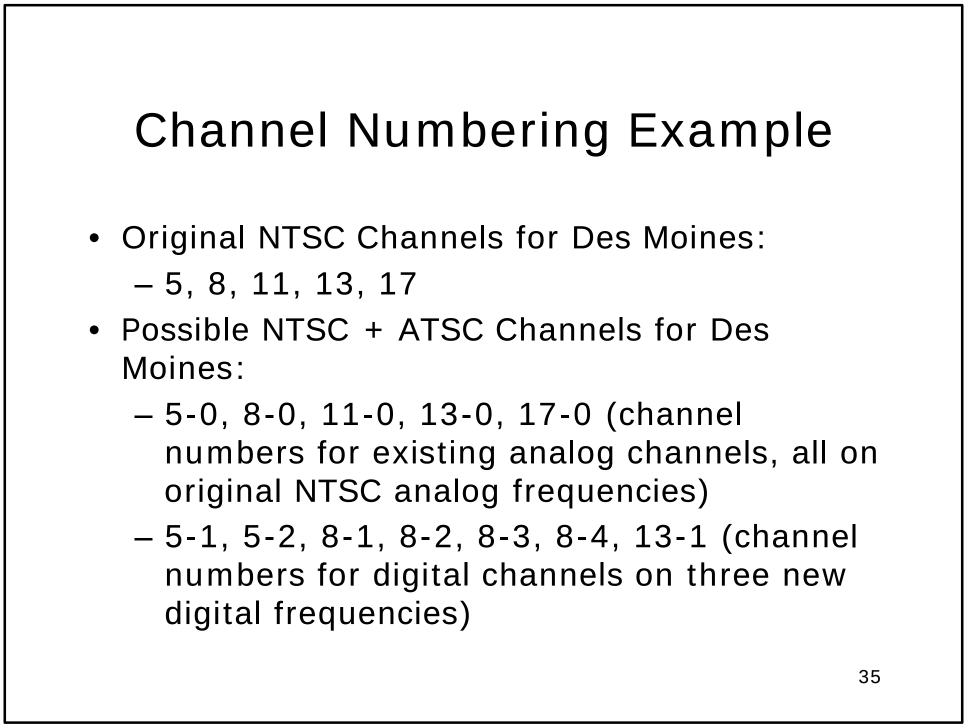


Channel Navigation Issues

- The necessary system information tables are distributed among several frequencies.
- The receiver must be able to determine which frequencies are being used.
- Information for analog channels may or may not be described in the received PSIP tables.
- Most receivers can only be tuned to a single frequency at a time. Cached information from other frequencies can become obsolete without warning.
- Virtual channel numbers have no correspondence to physical frequencies.

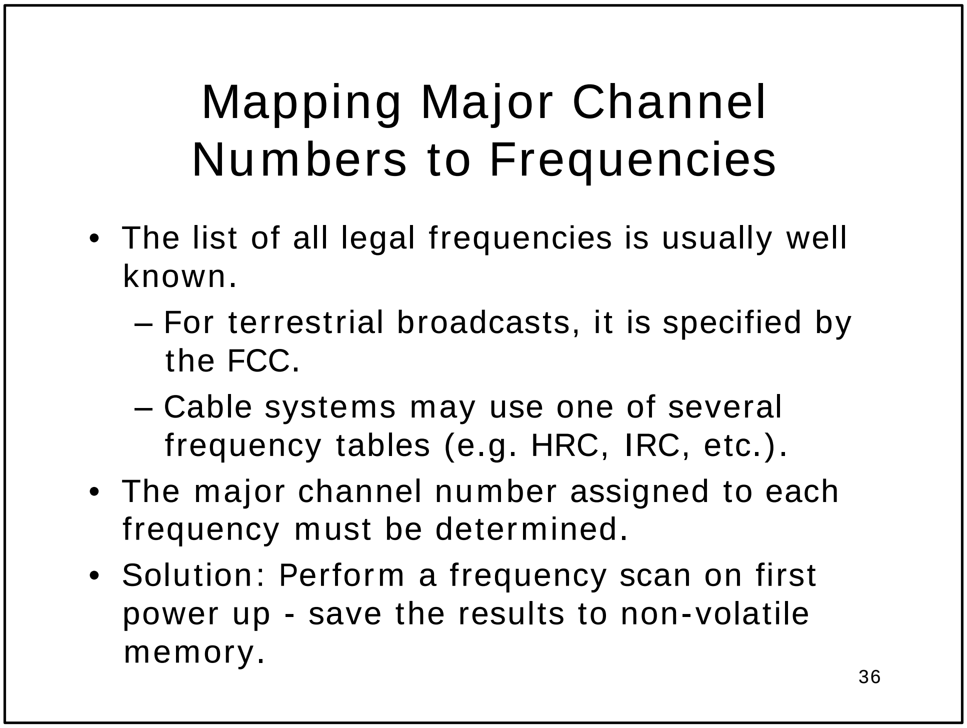
ATSC PSIP Channel Numbering

- Two part channel number:
  - Major channel number ⇒ Frequency
  - Minor channel number ⇒ Sub channel on frequency
- In USA: The major channel number is same for both analog and digital frequencies. (It is equal to the NTSC channel number for existing broadcasters.)
- In USA: A minor channel number of zero is used for a broadcaster’s analog channel.



Channel Numbering Example

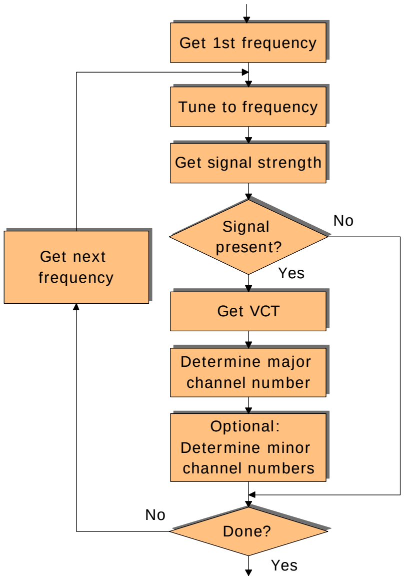
- Original NTSC Channels for Des Moines:
  - 5, 8, 11, 13, 17
- Possible NTSC + ATSC Channels for Des Moines:
  - 5-0, 8-0, 11-0, 13-0, 17-0 (channel numbers for existing analog channels, all on original NTSC analog frequencies)
  - 5-1, 5-2, 8-1, 8-2, 8-3, 8-4, 13-1 (channel numbers for digital channels on three new digital frequencies)



Mapping Major Channel Numbers to Frequencies

- The list of all legal frequencies is usually well known.
  - For terrestrial broadcasts, it is specified by the FCC.
  - Cable systems may use one of several frequency tables (e.g. HRC, IRC, etc.).
- The major channel number assigned to each frequency must be determined.
- Solution: Perform a frequency scan on first power up - save the results to non-volatile memory.

Frequency Scan for Digital Channels



Get 1st frequency

Tune to frequency

Get signal strength

Signal present?

No

Yes

Get VCT

Determine major channel number

Optional: Determine minor channel numbers

Done?

No

Yes

Get next frequency

What About Analog Channels?

- Don’t rely on a broadcaster’s PSIP data to find analog channels.
  - Information for analog channels may or may not be present in PSIP data.
  - Even if it is, analog signal may be too weak to receive clearly.
- Solution: Perform a similar frequency scan using the analog tuner to find analog channels.

What About Frequencies With No PSIP Data?

- Some stations are broadcasting digital, but are not yet broadcasting PSIP data.
- Modify frequency scan algorithm as follows:
  - If VCT is requested, but not received, register for the PAT instead.
  - Assign minor channel numbers from one to "n" consecutively to channels in PAT
  - Assign a major channel number corresponding to NTSC RF frequency
  - Always stay registered for the VCT when tuned to the frequency to detect when broadcaster does start transmitting PSIP

Typical Channel Management API Calls

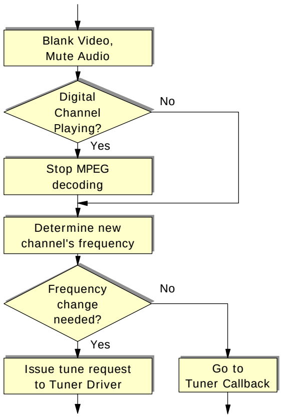
- tune();
- get_signal_strength();
- add_digital_frequency();
- add_analog_channel();
- channel_up();
- channel_down();
- channel_flashback();
- set_channel_number();
- get_channel_number();
- get_channel_info();

Channel Change API Calls

- Channel Management process can immediately determine the correct frequency, but may not know the specific minor channel number.
- Must first tune to the new frequency to update the channel map, then select the proper minor channel number.
- Channel change sequence involves up to four different threads of execution. (See following flowcharts).
- The goal is to perform the channel change sequence as fast as possible.

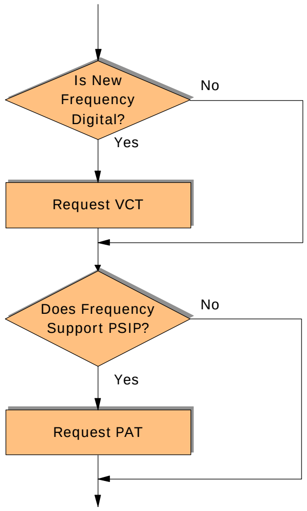
41

Channel Change: Channel Management Thread



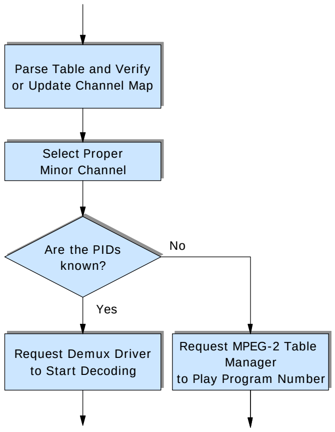
42

Channel Change: Tuner ISR Thread (Callback Function)



43

Channel Change: Channel Management Thread (Part 2)


Parse Table and Verify or Update Channel Map
Select Proper Minor Channel
Are the PIDs known?
No
Yes
Request Demux Driver to Start Decoding
Request MPEG-2 Table Manager to Play Program Number

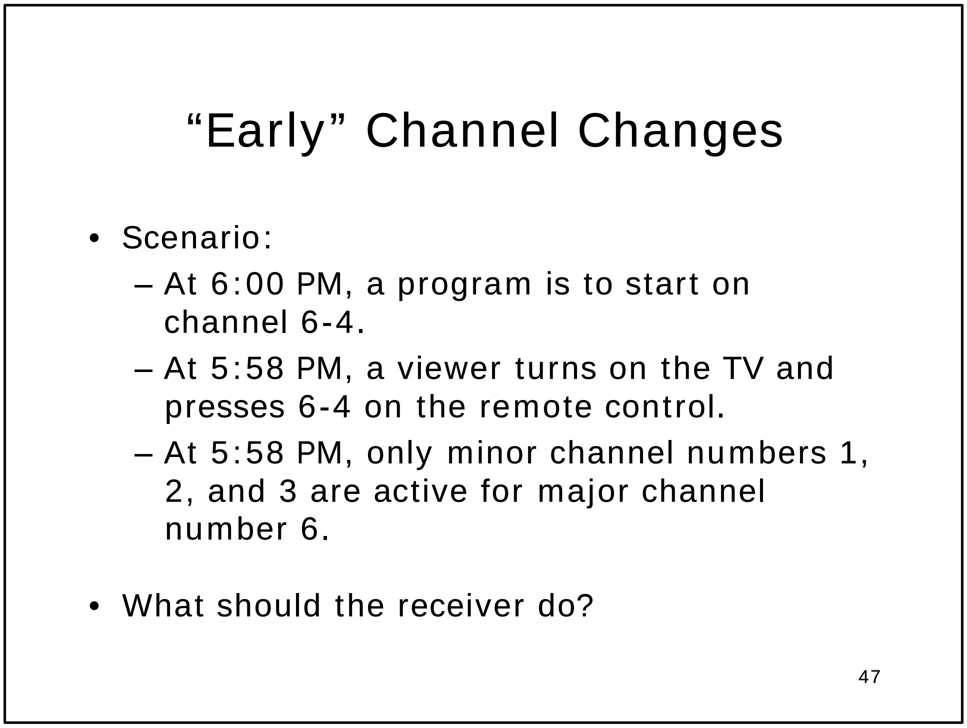
44

Fast Channel Switching

- Goal: Validate channel map as fast as possible when changing to a new frequency.
- Try to use MGT, since it is broadcast most frequently.
- Can’t reliably make decision number based solely on VCT version number.
- Possible Solution: Use VCT version number in combination with MGT’s CRC.

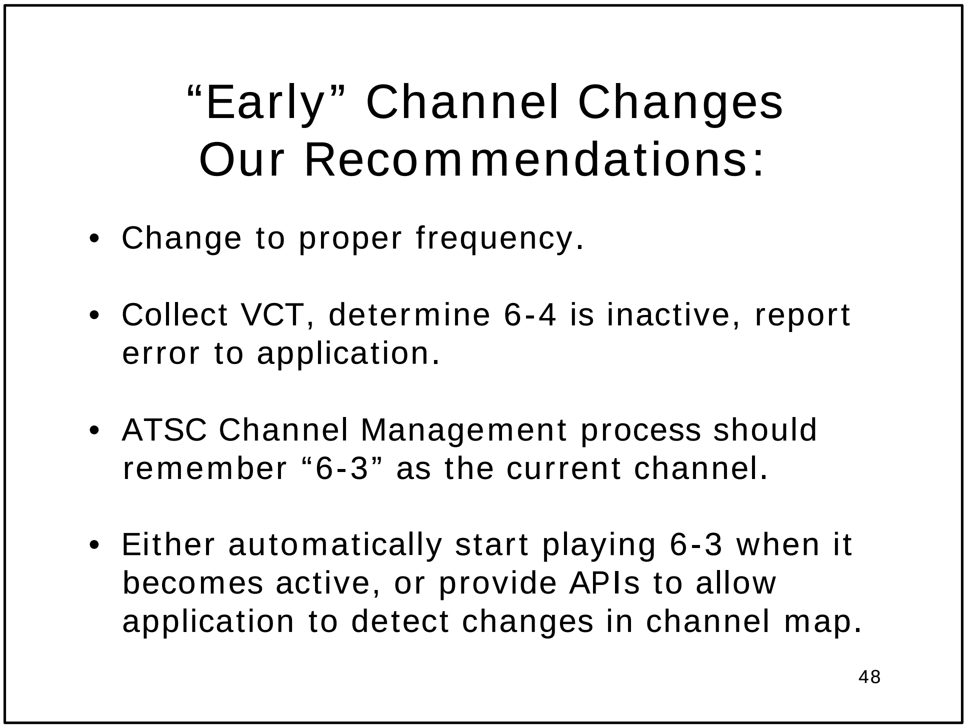
Supporting Favorite Channel Lists

- Many digital TV receivers (DBS, etc.) support favorite channel lists.
- Requires receiver to save the minor channel list for each frequency.
- One difficulty - some minor channels may not be permanent. Once they are in a favorite channel list, it is hard to know when to delete them.
- One possible solution - aging.



“Early” Channel Changes

- Scenario:
  - At 6:00 PM, a program is to start on channel 6-4.
  - At 5:58 PM, a viewer turns on the TV and presses 6-4 on the remote control.
  - At 5:58 PM, only minor channel numbers 1, 2, and 3 are active for major channel number 6.
- What should the receiver do?



“Early” Channel Changes Our Recommendations:

- Change to proper frequency.
- Collect VCT, determine 6-4 is inactive, report error to application.
- ATSC Channel Management process should remember “6-3” as the current channel.
- Either automatically start playing 6-3 when it becomes active, or provide APIs to allow application to detect changes in channel map.

The ATSC Electronic Program Guide Tables

The Event Information Tables

- Each table contains EPG data for a 3-hour period.
- PIDs can be determined from the Master Guide Table.
- Contains EPG event data.
- 4 tables mandatory, 128 allowed.
- Correlated to VCT with "source ID" field.
- Table ID Extension field is source ID.
- Uses Huffman compression for text.

The Extended Text Table

- Two types:
  - One for channels
  - One for events
- Channel ETT is correlated to channels via source ID field.
- Event ETT is correlated to events via source ID and event ID fields.
- Uses Huffman compression for text.

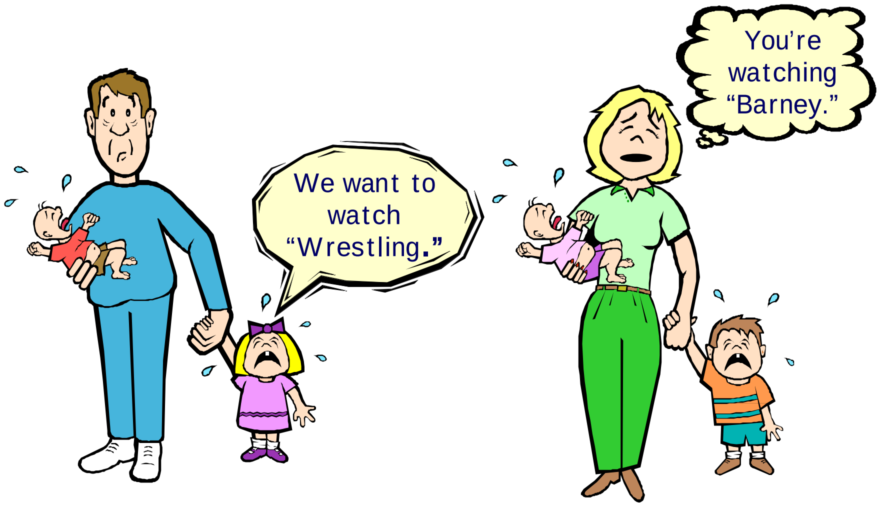
EPG API Issues

- Background caching of EPG information is difficult and not that useful.
  - Memory management issues make it hard
  - Information may become stale while it is cached
- Retrieving information event-by-event is very time consuming and cumbersome
- Our solution: Use an EPG “grid” cache.

EPG API Issues (Continued)

- For grid based EPGs, scrolling through the EPG grid screen should be easy to do with a single API call.
- May wish to allow easy interface to Java TV API or DASE API for EPG retrieval.
- Determining valid minor channel numbers is not always easy.
- May wish to allow two modes of operation:
  - Show all channels
  - Show only active channels

The ATSC Rating Region Table and Parental Control API



The Rating Region Table

- Defines one or more “parental control” rating scales (“rating dimensions”) for a rating region.
- Some rating dimensions may follow a graduated scale. Others may not.
- Content Advisory Descriptors in EIT or PMT will give an event’s actual rating value for one or more rating regions and rating dimensions.

Parental Control API Goals

- Possible ways of restricting viewing of content
  - by rating value(s)
  - by channel
  - by day of week / time of day
- Same methods can be used to enable viewing of content
- Access to parental control settings should be password protected

Typical Parental Control API Calls

- set_password();
- check_password();
- restrict_all_channels();
- allow_all_channels();
- restrict_channel();
- allow_channel();
- set_rating_mask();
- restrict_time_range();
- allow_time_range();

Some Table Processing Guidelines

The ATSC Master Guide Table

- Provides PIDs for EITs and ETTs.
- Provides size and version number information for all tables.
- Can be used to help validate channel map when changing to a new frequency.

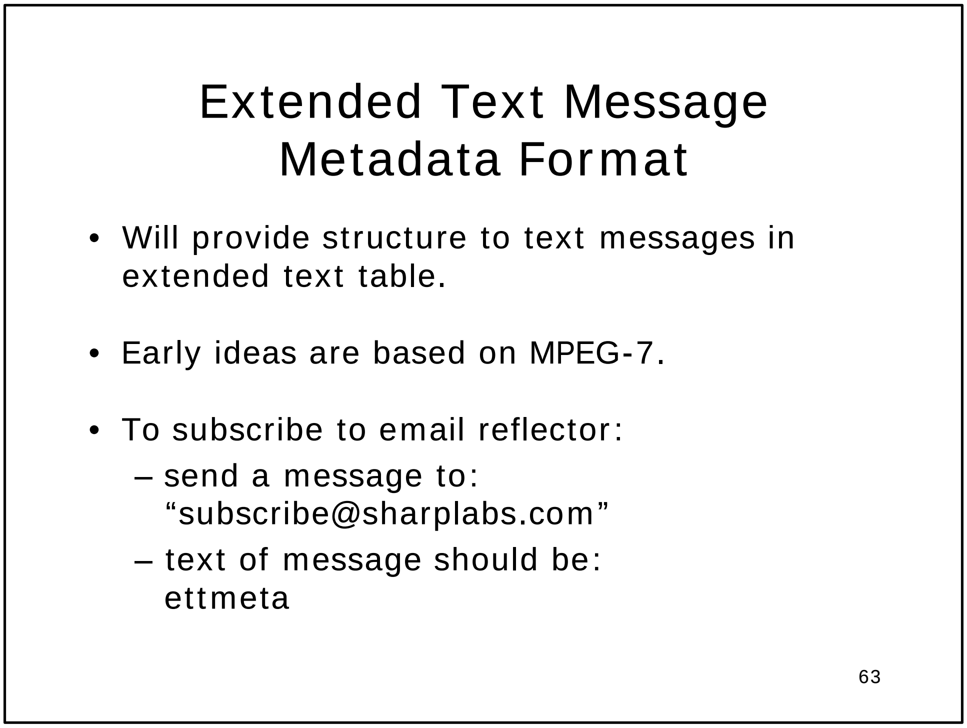
Digital VCR Issues

- Table version numbering may not be continuous
- System time will likely not be correct
- Be careful about updating channel map, EPG database, etc.

Other suggestions

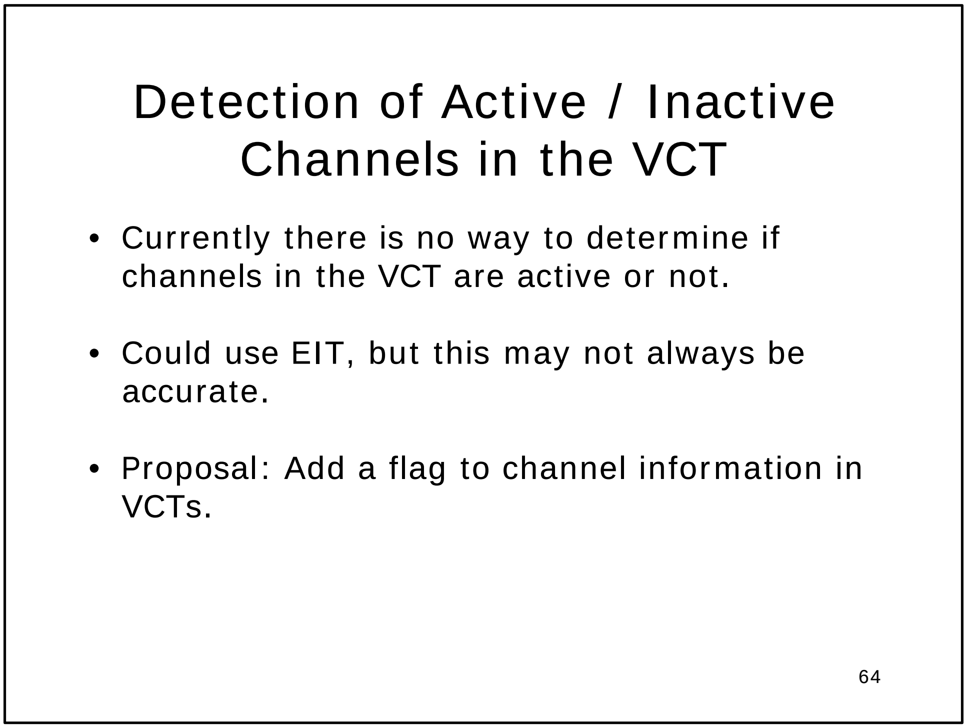
- Check protocol version number in all ATSC tables.
- When receiver is in “standby” mode, it can scan frequencies to keep tables updated.

ATSC PSIP Work In Progress



Extended Text Message Metadata Format

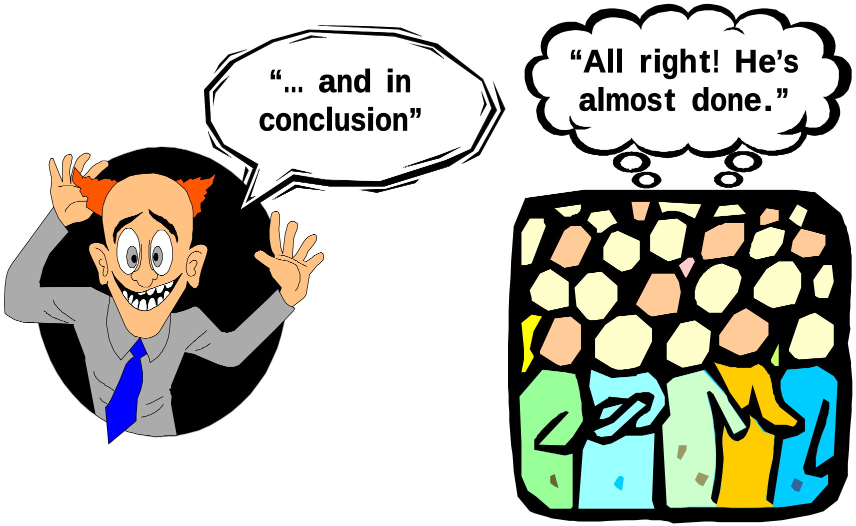
- Will provide structure to text messages in extended text table.
- Early ideas are based on MPEG-7.
- To subscribe to email reflector:
  - send a message to: "subscribe@sharplabs.com"
  - text of message should be: ettmeta



Detection of Active / Inactive Channels in the VCT

- Currently there is no way to determine if channels in the VCT are active or not.
- Could use EIT, but this may not always be accurate.
- Proposal: Add a flag to channel information in VCTs.

Conclusions



Suggestions

- Capabilities of the MPEG-2 transport demultiplexer IC are critical.
- Use debug trace buffer to track event sequence.
- Be careful of using I2C for data intensive peripherals.
- Involve the EPG application writers in the design of the EPG API.
- Cut and paste Huffman tables from A/65 document.

To Download This Presentation…

… Go to http://www.softproseinc.com