# Real-time video with VP8/WebM

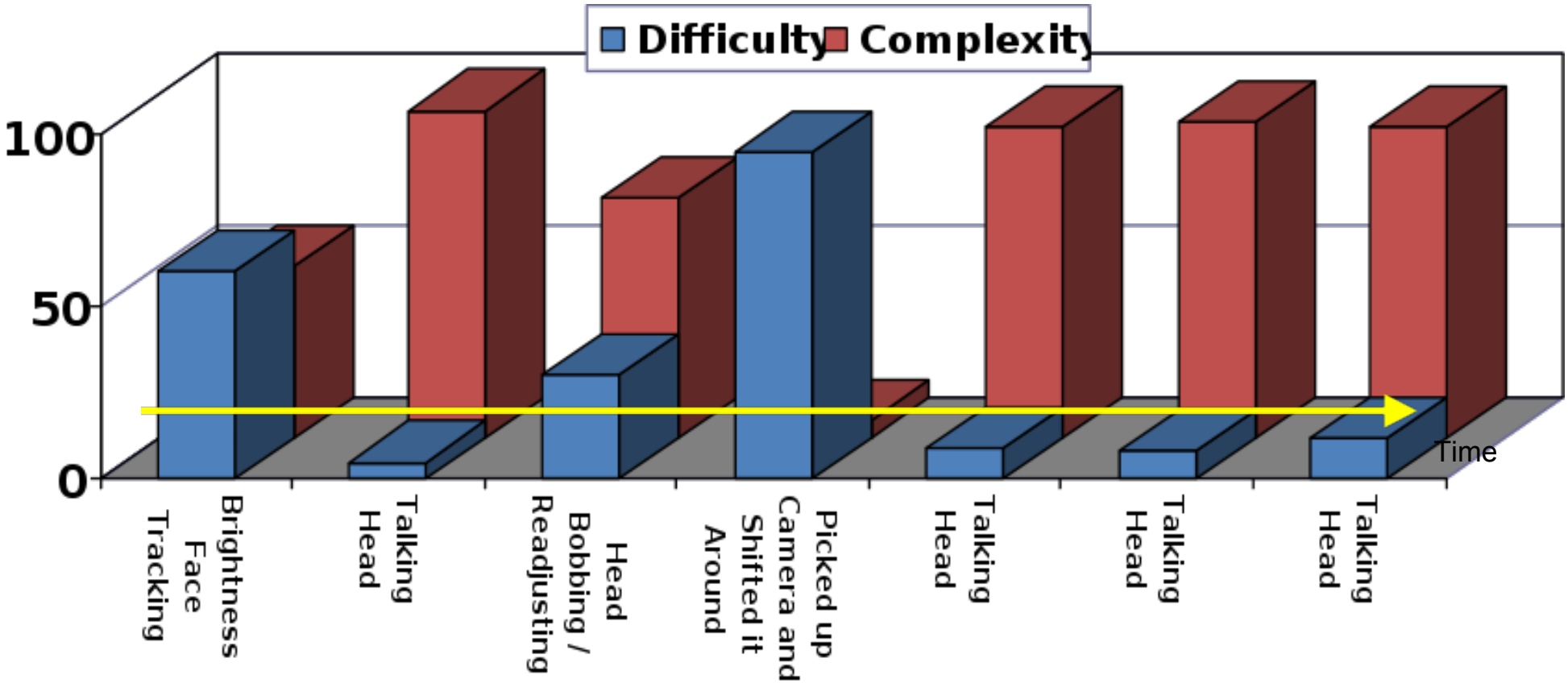
**Paul Wilkins**
Senior Software Engineer, WebM Project

# Key VP8 features

webm

- Multiple intra prediction modes for 4x4 and 16x16 pixel blocks
- Motion vectors for 16x16, 16x8, 8x16, 8x8 or 4x4 regions
- Single step ½ and ¼ pixel motion filters
- 4x4 DCT transform for coding the residual error signal
- Adaptive loop filtering (two complexity levels)
- Context predictive binary arithmetic coding (contexts defined at the frame level)
- Golden and constructed (alternate) reference frames

# Frame adaptive complexity for real-time

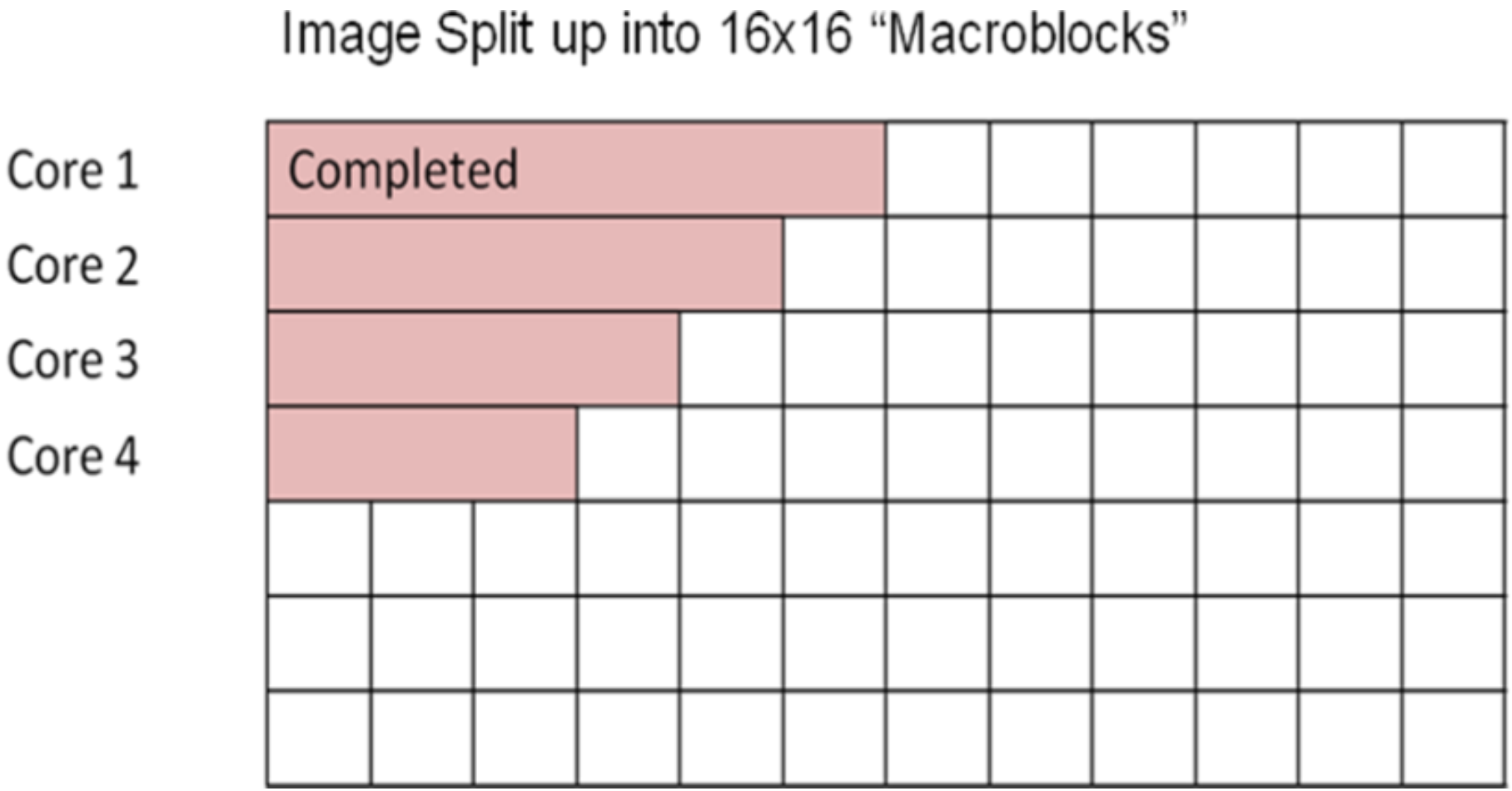



A VP8 encoder can adjust the encoder complexity dynamically to keep the encoding time per frame constant.

# Frame adaptive complexity - continued

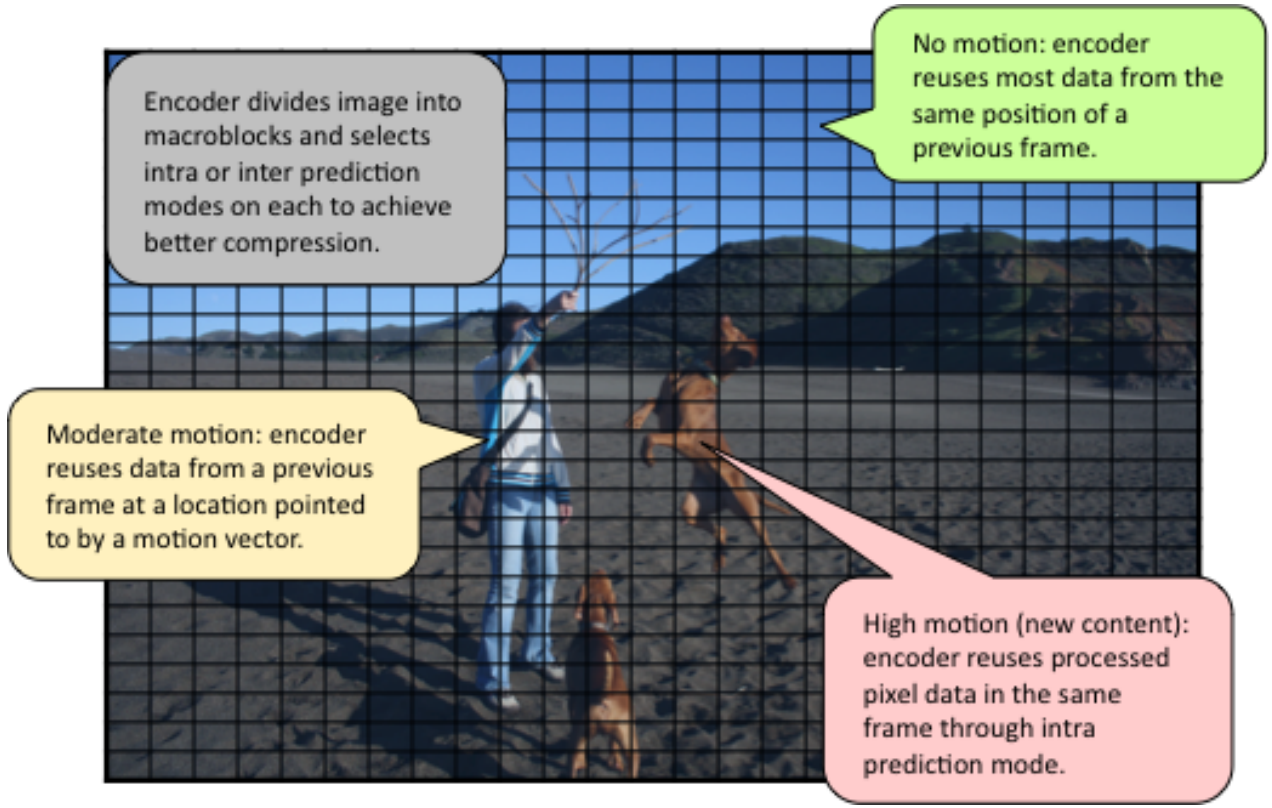
- Mode Real-time
  - Can assign the encoder a target % of the CPU through a "speed" parameter
- 16 encoder complexity levels which:
  - Select encoder features
  - Enable/disable rate distortion
  - Determine motion search strategy
  - ½ and ¼ pel sub pixel motion (on/off, filter type)
  - Set loop filter options (type, strength selection)
  - DCT and quantizer optimizations
  - Disable or thresholding expensive modes (4x4 intra, split macroblock)
- Complexity level selection can be set manually by the user or automatically by the encoder

# Using multiple cores to encode or decode




Each core can work on a separate row of 16x16 pixels (macroblocks) both to encode and decode.

# Block adaptive complexity



# Block adaptive complexity - continued

- Adaptive thresholds
  - Allowed threshold range set at frame level. Adapts according to how often a mode is chosen. The modes most often giving benefit ripple to the top.
- Frequency sampling
  - Expensive modes or methods (e.g., longer or slower motion searches) are sampled every *X* macroblocks. If they give benefit the sample frequency increases, else it falls within bounds defined at the frame level.
- Encoder breakout
  - Skips just about everything if the base 0,0 prediction error is below a user-defined threshold and no significant changes in level (brightness) are detected.

# Background / foreground coding

Signal noise in the background can prevent the quantizer from dropping.

- Golden frames
  - A stabilized, high-quality reference image (periodic or on demand update)
- Constructed reference frames
  - Noise-filtered, hidden reference frames if some encoding lag is acceptable
- Segmentation
  - Up to four segments with support for quantizer and loop filter adjustments
  - Background refresh

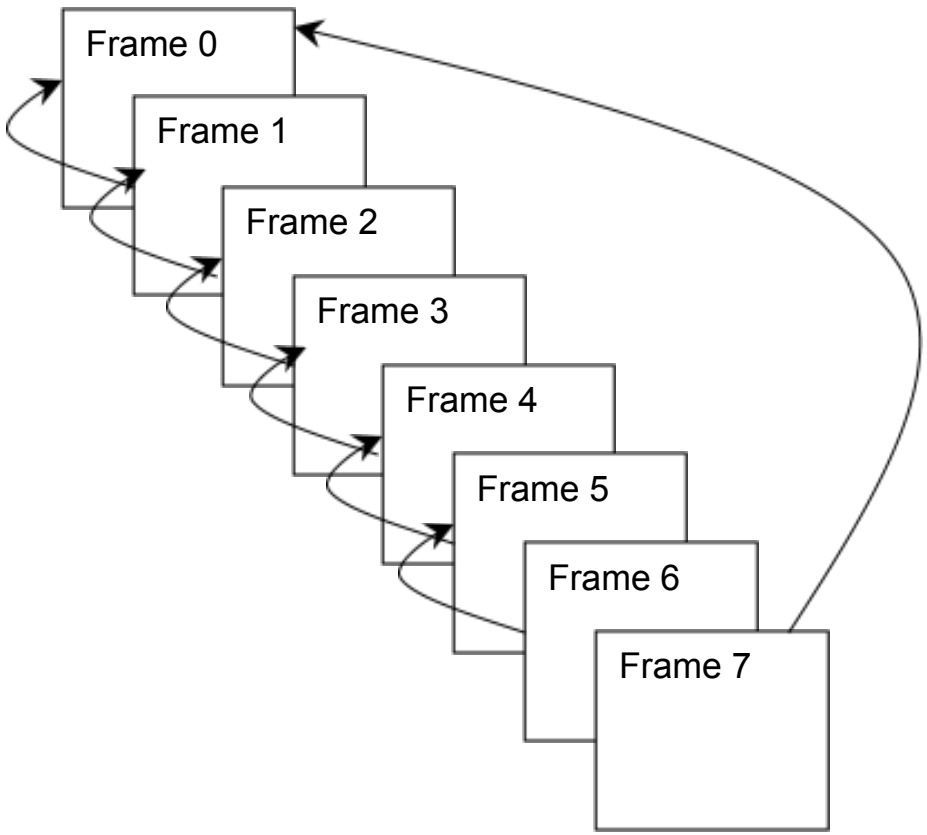
# Error recovery and packet loss

webm

## Error-resilient mode

- Ensures entropy contexts are defined even if frames are lost
- Allows the encoder to recover by predicting from a known “good” reference frame
- Support for up to five levels of temporal redundancy

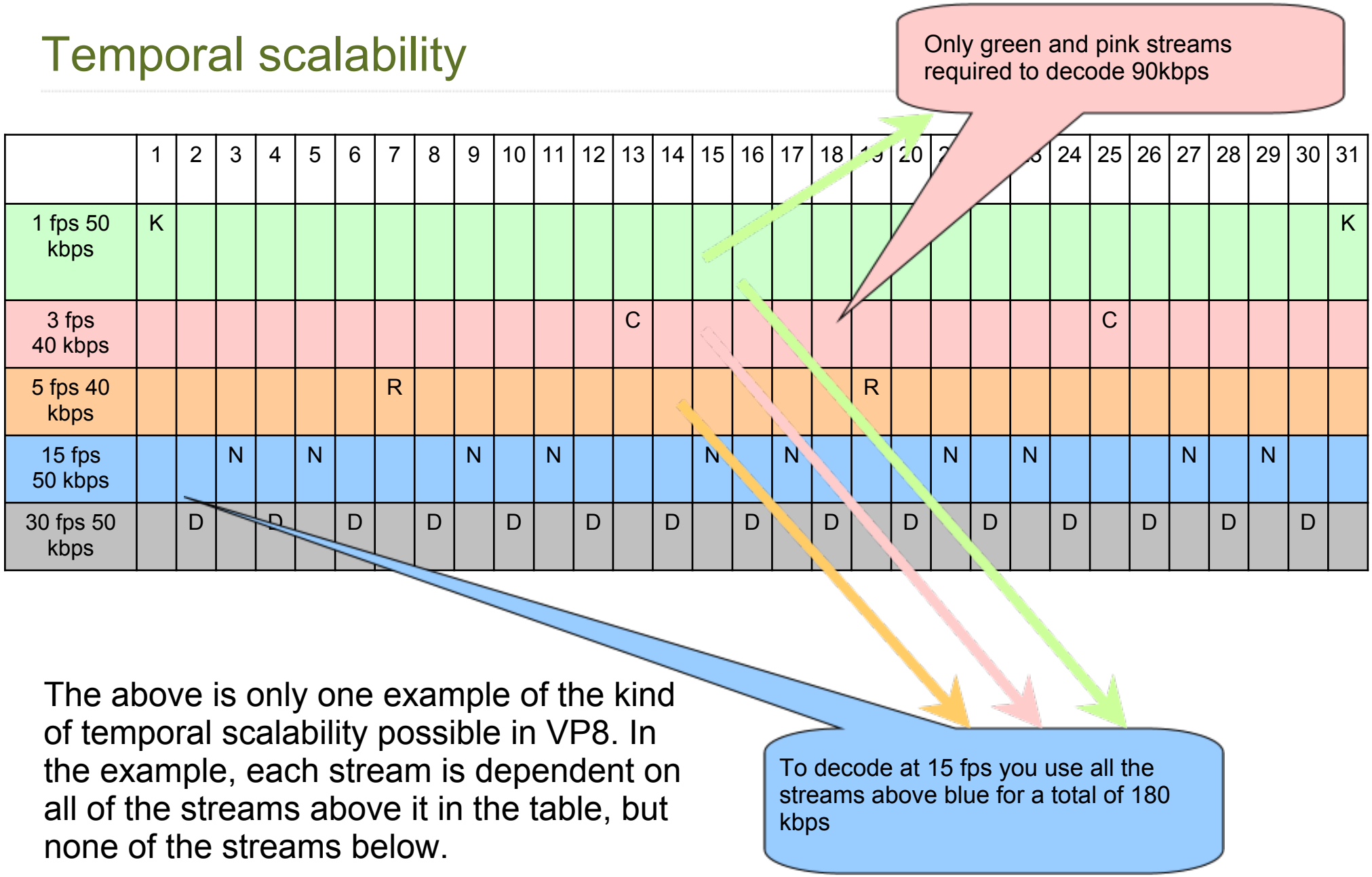
# Using golden frames for packet loss recovery




**Frame 0** is a key frame / gold frame
**Frame 1 through 6** build predictors using the prior frame
**Frame 7** uses only frame 0 as a reference.

If any frame between 1 and 6 is lost VP8 can still decode frame 7 as it references only to frame 0.

# Temporal scalability




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 fps 50 kbps | K | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | K |
| 3 fps 40 kbps | | | | | | | | | | | | | C | | | | | | | | | | | | C | | | | | | |
| 5 fps 40 kbps | | | | | | | R | | | | | | | | | | | | R | | | | | | | | | | | | |
| 15 fps 50 kbps | | | N | | N | | | | N | | N | | | | N | | N | | | | N | | N | | | | N | | N | | |
| 30 fps 50 kbps | | D | | D | | D | | D | | D | | D | | D | | D | | D | | D | | D | | D | | D | | D | | D | |

The above is only one example of the kind of temporal scalability possible in VP8. In the example, each stream is dependent on all of the streams above it in the table, but none of the streams below.

# Handling dropped packets with back channel

webm

- Packet retransmission
  - If a receiver doesn't get a packet in time it sends a message to the sender requesting the packet again.
- Forward error correction
  - Send duplicate (parity packets) so that lost packets can be reconstructed. E.g. Reed Solomon or XOR packets.
- Recovery frame requests
  - Request a frame that is not dependent on the last frame but on a known good frame from the more distant past.
- Key frame request
  - If all else fails request a new key frame. I.e. a frame not dependant on anything.

# Partition structure

Each VP8 frame comprises a minimum of 3 partitions:

- Uncompressed partition
  - Minimal header information
- Header and prediction signal partition
  - All other header information and all prediction modes and motion vectors.
- Error signal partitions (1, 2 ,4 or 8)
  - Signaled at the frame level, the error signal partitions allow for multiple instances of the arithmetic decoder to be instantiated at high data rates. However, there is some contextual dependency between partitions/macroblock rows.

# Questions?

Send to:
codec-discuss@webmproject.com

Full VP8 bitstream reference available at
https://datatracker.ietf.org/doc/draft-bankoski-vp8-bitstream/