Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)

Submission Title: Intel OCC Proposal

Date Submitted: 8 January 2016

Source: Rick Roberts, Intel Corporation
Location: Hillsboro, Oregon
Contact Info: 503-929-5624, richard.d.roberts@intel.com

Re: CFP Response

Abstract: CFP Response

Purpose: CFP Response

Intel 802.15.7r1 OCC Proposal

Rick Roberts, Javier Perez-Ramirez

Introduction

Intel is pleased to present this proposal to IEEE802.15.7r1 that contains technical details to facilitate the inclusion of optical camera communication modes into the IEEE802 family of standards in a manner that we believe is fully compliant with the IEEE802 reference model, while still offering full backwards coexistence with IEEE802.15.7-2011.

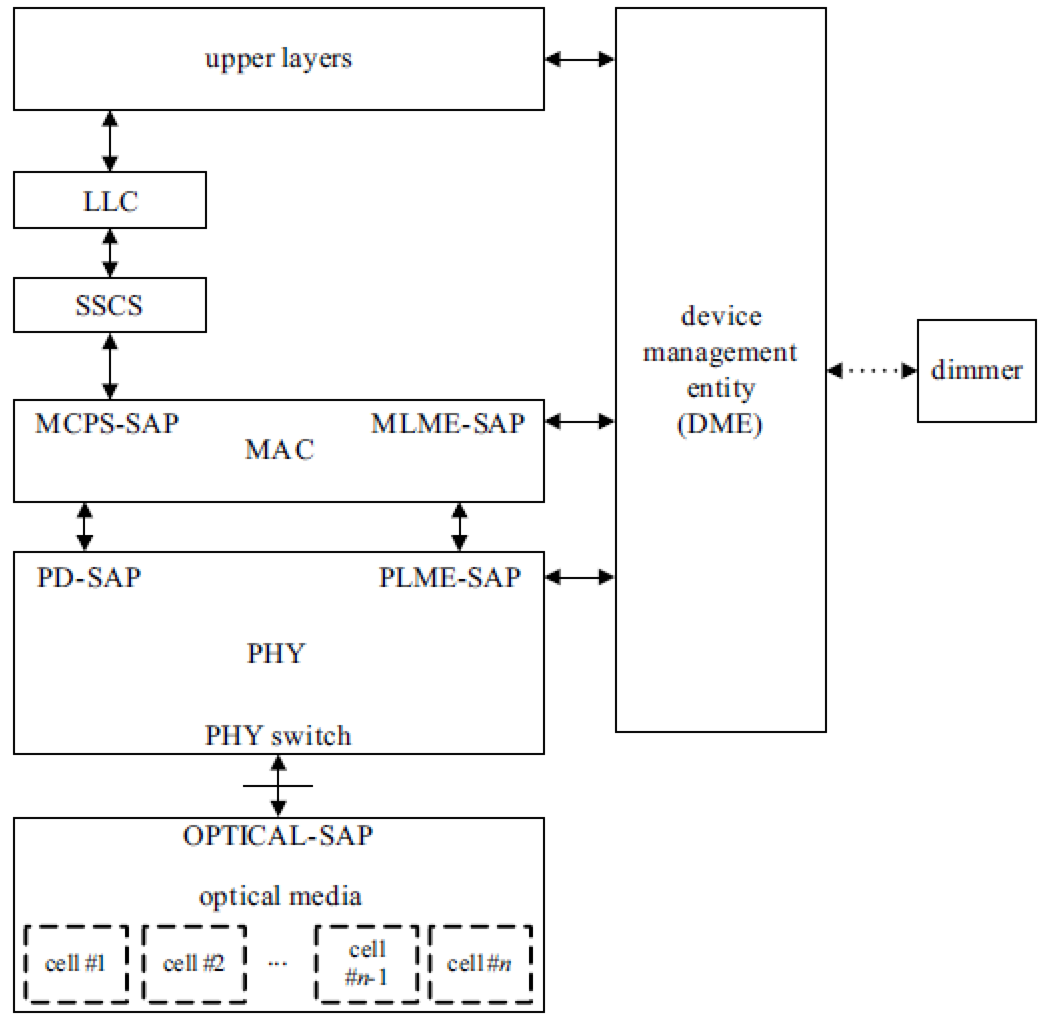


Figure 3—VPAN device architecture

Table of Contents

Proposing two OCC PHY types:

1. Low Rate OCC ... transmission is via serial (time domain) waveform from one or more LED lights

2. High Rate OCC … transmission is via two dimensional codes (i.e. QR codes)

Proposing two OCC MAC types:

1. High Rate OCC: Uses the 802.15.7-2011 broadcast mode MAC clauses with new MAC clauses that support the chosen two dimensional codes.

2. Static Message OCC: new MAC clauses that support one or more LED lights transmitting a message that seldom changes.

OCC Low Rate PHY

1. Undersampled Frequency Shift On-Off Keying
2. High Frame Rate Nyquist Region-of-Interest Sub-sampling

Undersampled Frequency Shift ON-OFF Keying

Undersampled Frequency Shift ON-OFF Keying

Transmitter Requirements

CamCom transmitters use "signals of opportunity"; that is, there may be little or no control over the light intensity and the associated applied level of dimming, which in turn means there is no guaranteed quality of service .

- modulation must provide acceptable performance with a wide dimming requirement (e.g. 10% to 90%) and exhibit no bit flicker or frame flicker
- average light intensity must be constant over any given interval not longer than 10 ms and not shorter than 1 ms during the transmission of data
- modulation technique has to be able to work with either a monochrome LED, a RGB LED or a white LED (blue LED with yellow phosphor)
- technique must be simple so as to minimize modification to signage (i.e. avoid wide bandwidth modulation redesign)

Receiver Requirements

- The receiver hardware must be an unmodified smartphone platform utilizing the embedded camera (either front facing, back facing or both) running at a viable consumer video frame rate (e.g. 30 fps)

- The signal processing is done on the host processor via an application software package

- The message latency is not critical but must not be annoying to the user (i.e. processing does not have to be real-time)

- The chosen technique must work satisfactorily with either a rolling shutter image sensor or a global shutter image sensor

- The chosen technique must work when the signal sources appear in the image as points (i.e. line-of-sight path).

Proposed solution … frequency modulate the PWM dimming waveform

- encode the bits using a form of DC balanced differential encoding called undersampled frequency shift ON-OFF keying (acronym UFSOOK)
- similar to frequency shift keying: defined mark and space ON-OFF keying frequencies for encoding bits
- frequencies are greater than 100 Hz to avoid flicker
- mark (logic 1) and space (logic 0) frequencies are selected such that when undersampled by a low frame rate camera the mark/space frequencies alias to low pass frequencies that can then be further processed to decode the bit values

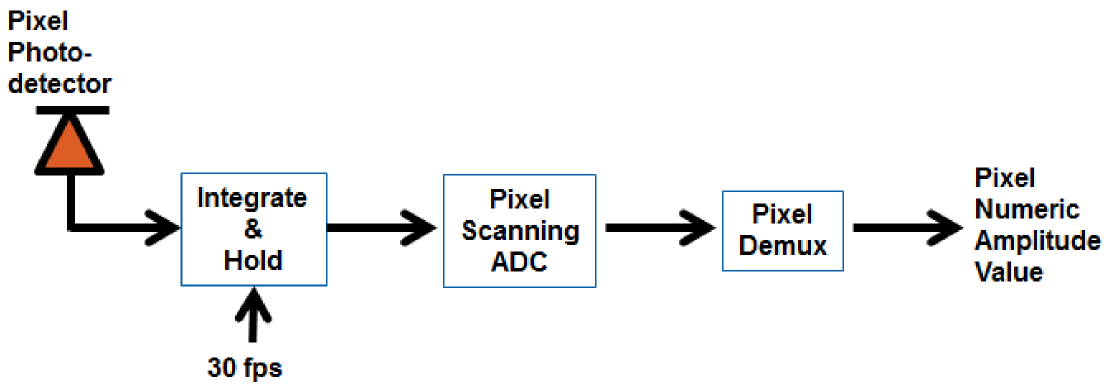


image sensor model: 2 dimensional lightwave-to-digital converter

- ✓ pixel photodetector produces a signal proportional to the incoming integrated light intensity
- ✓ signal held by the scanning ADC establishing the frame rate of the video camera

Encode bits as OOK frequencies (ON-OFF blinking LED) such that

- **logic zero:** represented by a OOK frequency that is an integer multiple of the camera frame rate.
- **logic one:** represented by an OOK frequency that is an integer multiple ± ½ of the camera frame rate.

When undersampled by the camera (appropriately fast exposure setting), the OOK frequencies alias to

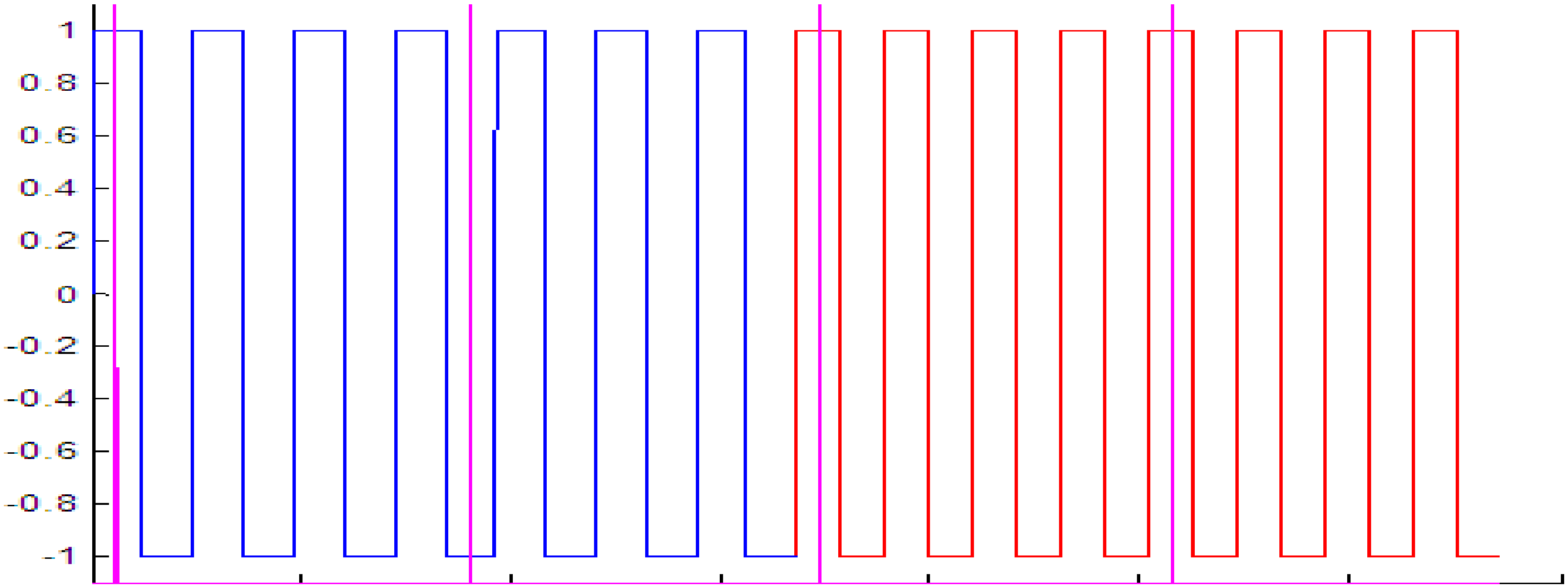
- **logic zero:** aliases to a steady state light (either ON or OFF)
- **logic one:** aliases to a blinking light at ½ the camera frame rate

The video frame-to-frame decoding rules are summarized below.

$$x(t,\omega_{\Delta}) = \begin{cases} unchanging & \text{"0"} \\ toggling & \text{"1"} \end{cases}$$

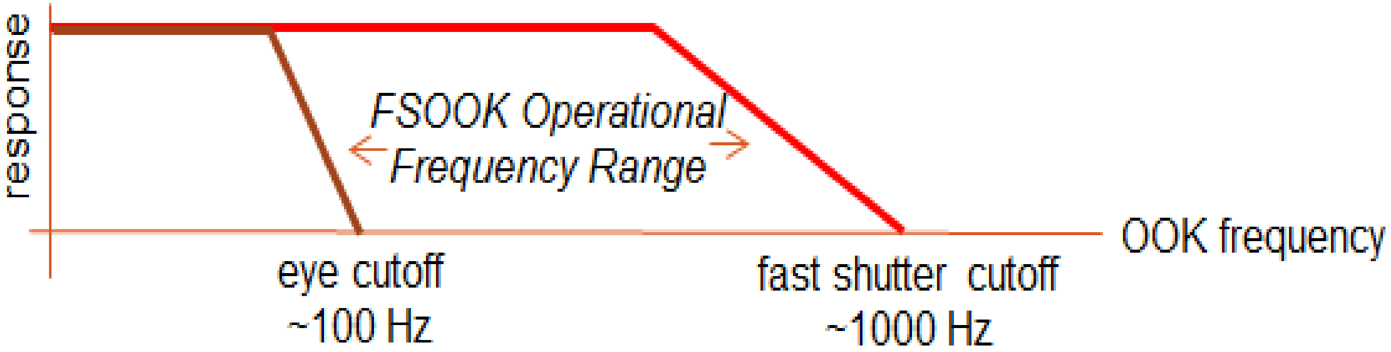
Adhering to the stated rules will always result in there being an even number of cycles of OOK per bit for a space frequency and an odd number of cycles for a mark frequency; hence, the “code” is always balanced.

Example of how bits are sent via the blinking lights …



- (figure left) Y-axis is read as a +1 turns the light ON and a -1 turns the light OFF
- logic one → 7 cycles of 105 Hz OOK (shown in blue)
- logic zero → 8 cycles of 120 Hz OOK (shown in red)
- composite waveform represents bit pattern “1 0”
- waveform sampled 30 times per sec by a camera (magenta sampling strobes)
- two samples per bit (bit rate half the sample rate)
- for logic 1 (blue) the two samples differ in value
- for logic 0 (red) the two samples have the same value

The UFOOK waveform transitions can be seen by a camera with the appropriate exposure setting, but not by the human eye … camera's exposure setting can be much faster than the eye

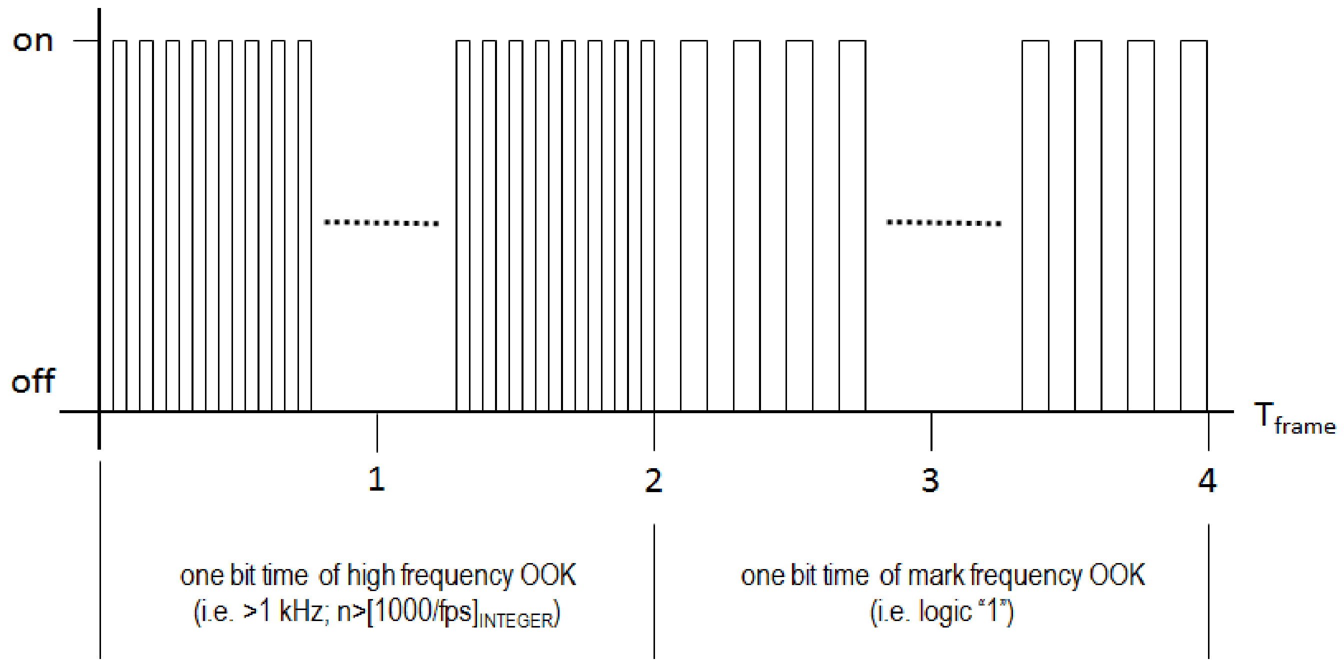


- human eye cutoff frequency is in the vicinity of 100 Hz
- camera's cutoff response can significantly exceed 100 Hz
- techniques shown in this presentation require a relatively intense light source (i.e. high SNR)

Start frame delimiter and frame structure

Start frame delimiter (SFD) - appended to the beginning of each frame of data. End of the data frame is indicated by the second appearance of the SFD (beginning of the next frame).

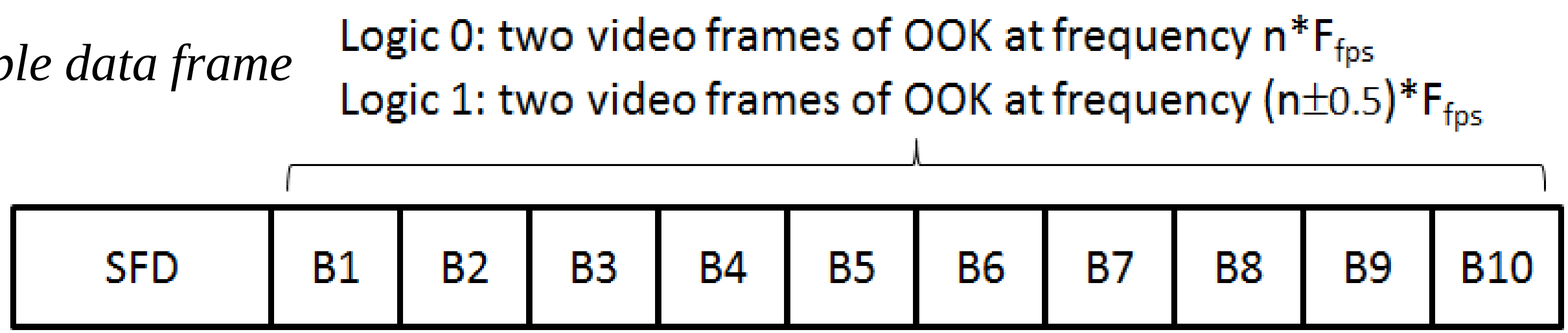
SFD Definition



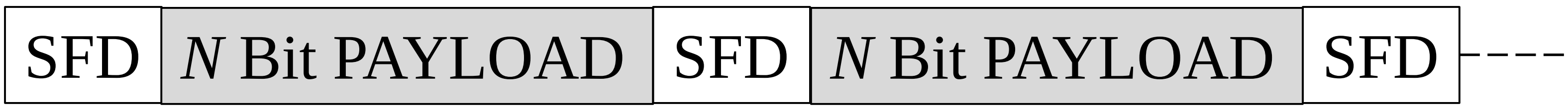
SFD, which is two bit times long (i.e. four video frames), is sent prior to a data frame.

- The first bit is sent at an OOK frequency that cannot be followed by a normal smartphone image sensor (i.e. > 1 KHz) … pixel integrator extracts the average light intensity such that the light appears half ON (assuming 50% duty cycle).
- The next bit of the SFD is the transmission of the logic "1" *mark* OOK frequency and can be used to verify the correct sampling phase (i.e. if a logic zero is read then discard the data frame).

Example data frame



The UFSOOK payload is a repetitive message that is seldom changed. The basic UFSOOK frame structure is shown below.



The *N* bit payload may be of any length and contain additional delimiters to support MIMO as discussed in a following section.

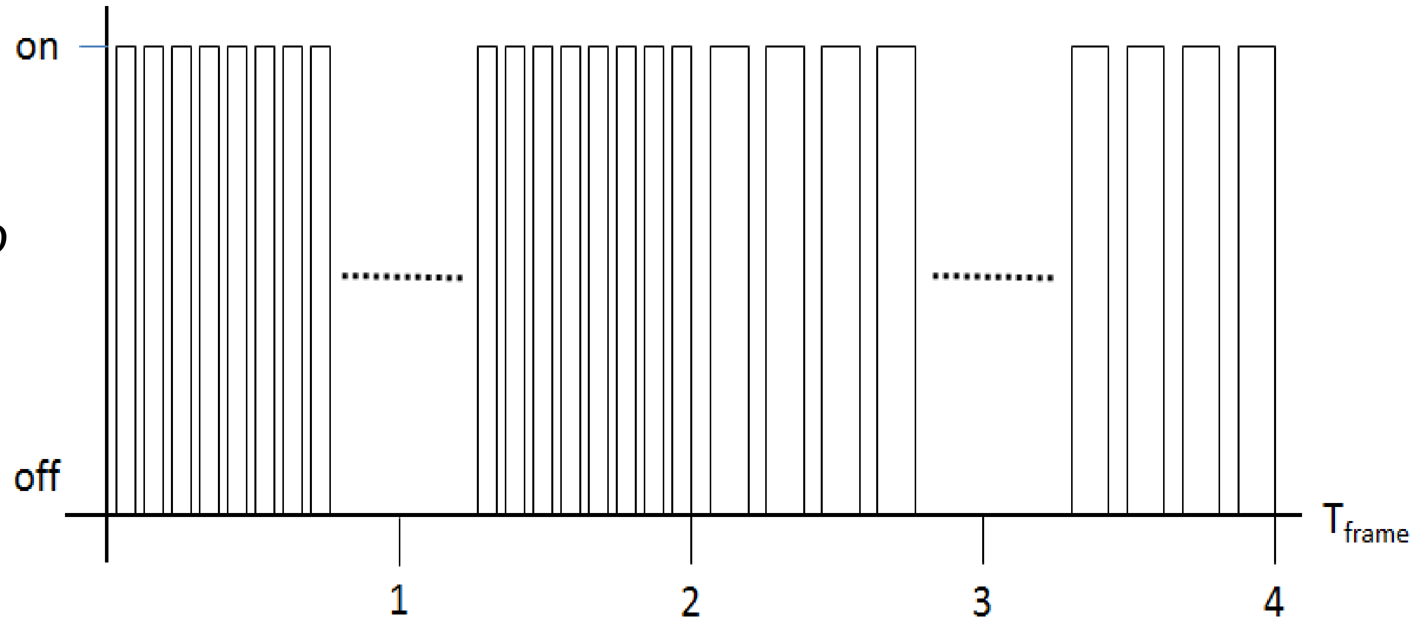
The end of a data frame is indicated by the reappearance of the SFD which also indicates the beginning of the next frame.

LED dimming support with multiphase sampling

- previous discussion assumed 50% duty cycle for the UFSOOK modulation
- in reality the duty cycle varies from 50% (to change brightness)
- Duty cycle choice is up to the LED light/signage vendor

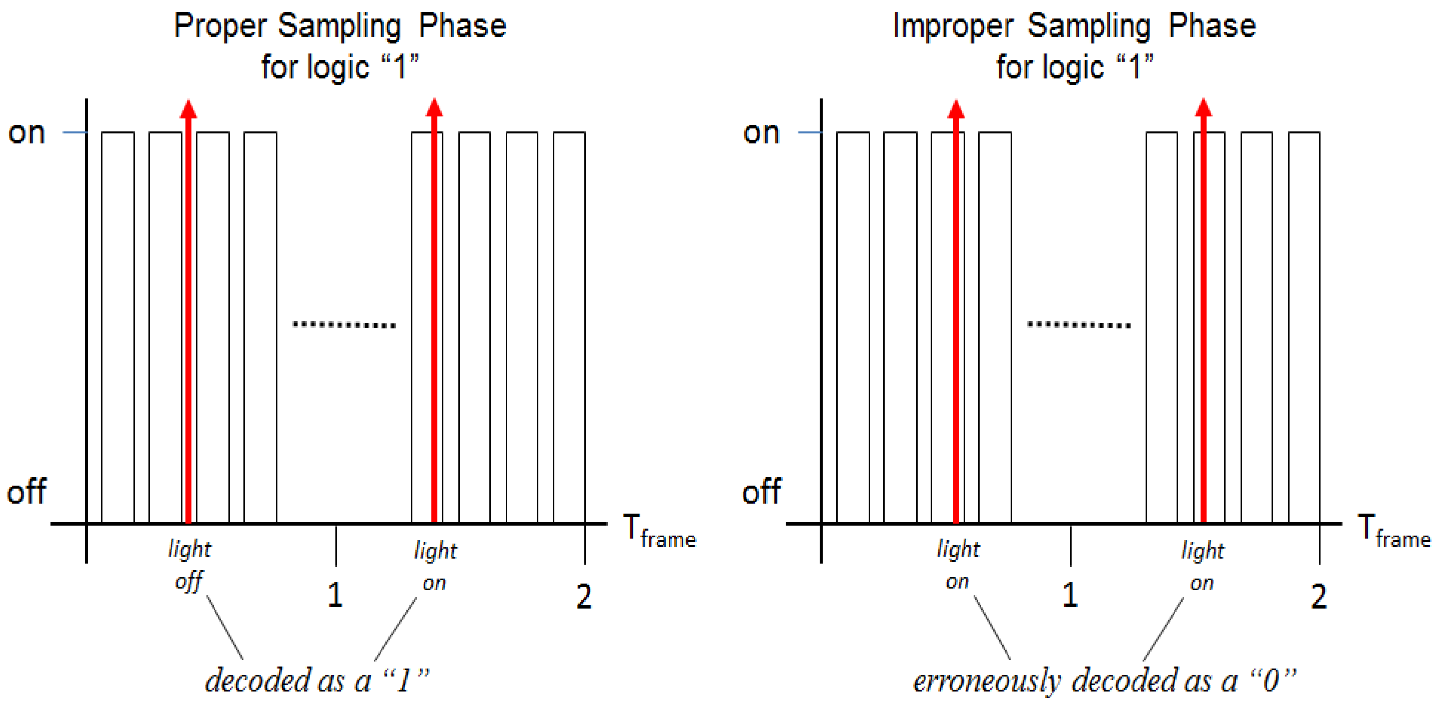
Impact of dimming: changes the duty cycle of the OOK frequency waveforms as shown below *(light is ON more than it is OFF)*

75% duty cycle SFD

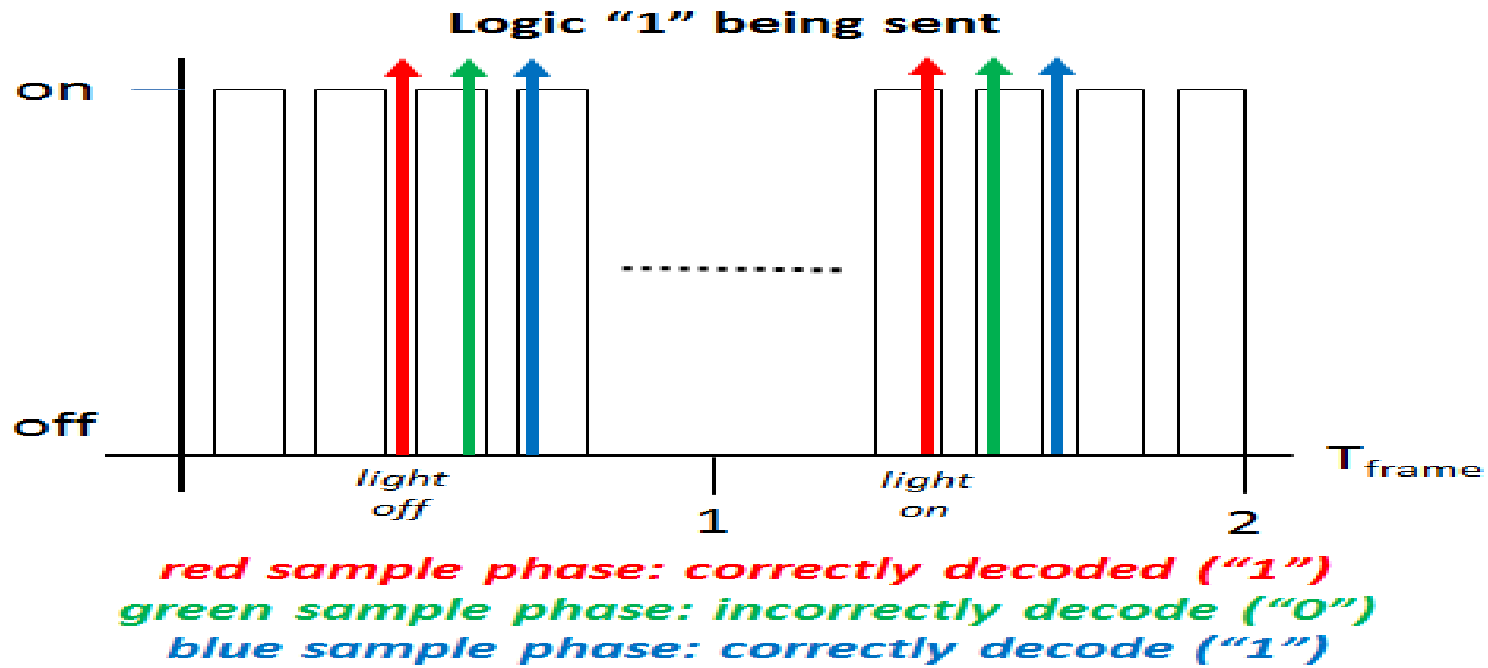


- challenge - find the correct sampling phase without synchronous detection (i.e. free running clocks)
- while 50% duty relatively immune to the random sampling phase …
- 75% duty cycle runs the danger of crippling the differential code by improper sampling

75% duty cycle mark frequency



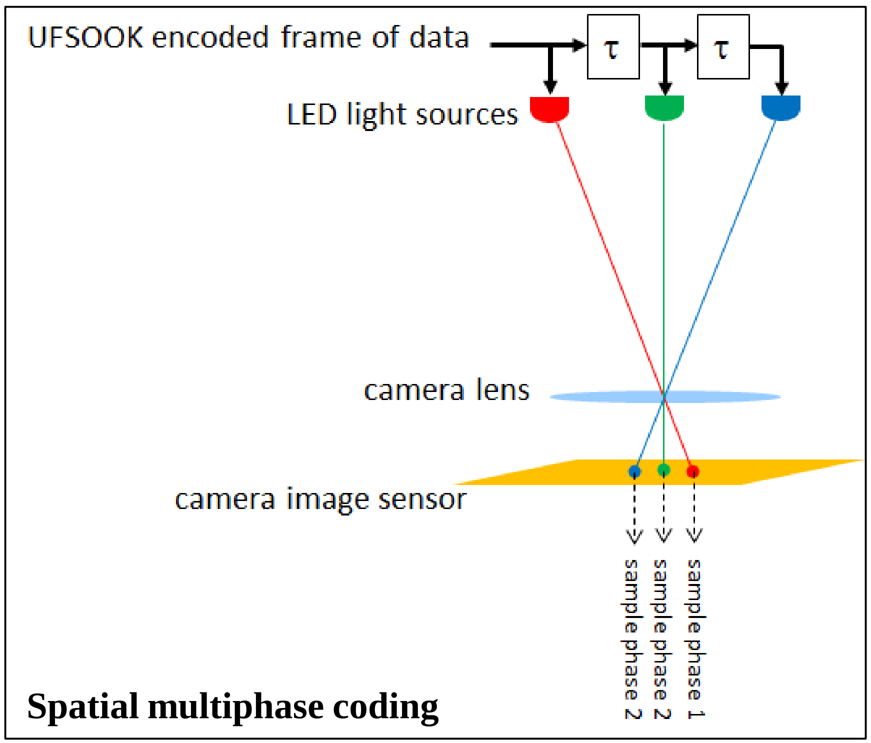
Cause of improper sampling: narrowing of the light "off" time as the duty cycle is increased
Solution: multi-phase sampling has to be used to correct the problem

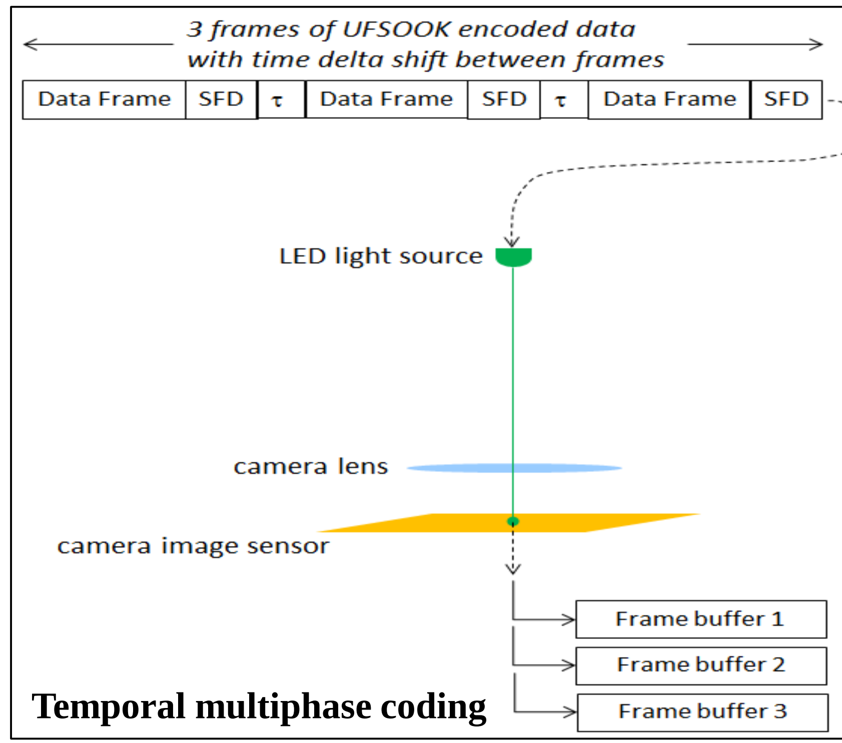


- example: UFSOOK encoded frame sampled at three different phases
- 2 out of the 3 phases results in a correct SFD
- incorrect phase can easily be identified during the SFD
- incorrect sample phase will decode the last two SFD frames as logic "0" (i.e. discard the related sampled data)
- repeat coding can be used on the data frame of the remaining "correct" phases to detect and/or correct frame bit errors

spatial multiphase coding (multi-source parallel transmission):

- same frame of UFSOOK encoded data is broadcast by each of three LEDs with slight time delay
- receiving camera lens provides spatial separation between each of the light sources
- time shifted data frames can be extracted from the appropriate pixels
- each pixel data stream is qualified based upon a valid SFD
- valid data frames are decoded into bit payloads
- repeat coding can be used to detect and/or correct errors
- ***equivalent temporal process can be done via a single LED by serial time shifted transmission***





Optional CRC

SFD	*N* Bit PAYLOAD	CRC	SFD	*N* Bit PAYLOAD	CRC	SFD

A CRC checksum can be added to each MIMO payload to further assist in qualifying error free packets (even when doing spatial redundancy).

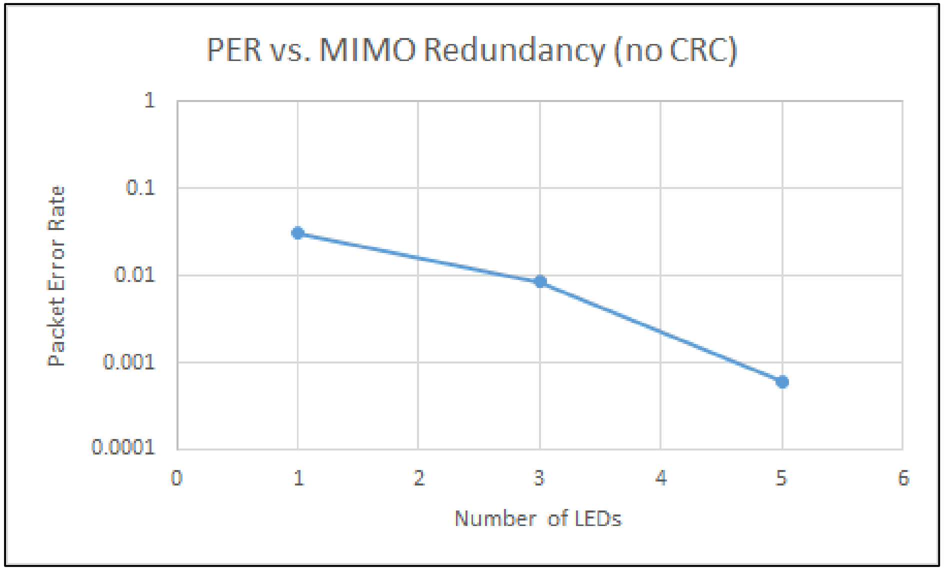
For 8-bit payloads (LED ID) we use a three bit CRC-3 checksum based upon the polynomial

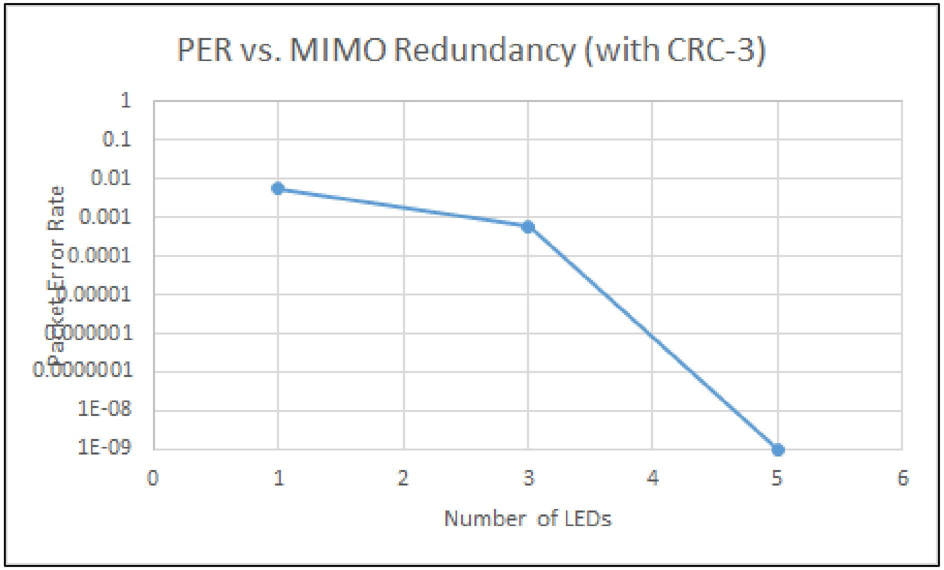
$$X^{3} + X + 1.$$

It was found using CRC improves performance but increases overhead (8 bits data, 5 bits overhead).

Other payload lengths will require other CRC polynomials.

UFSOOK Performance Data

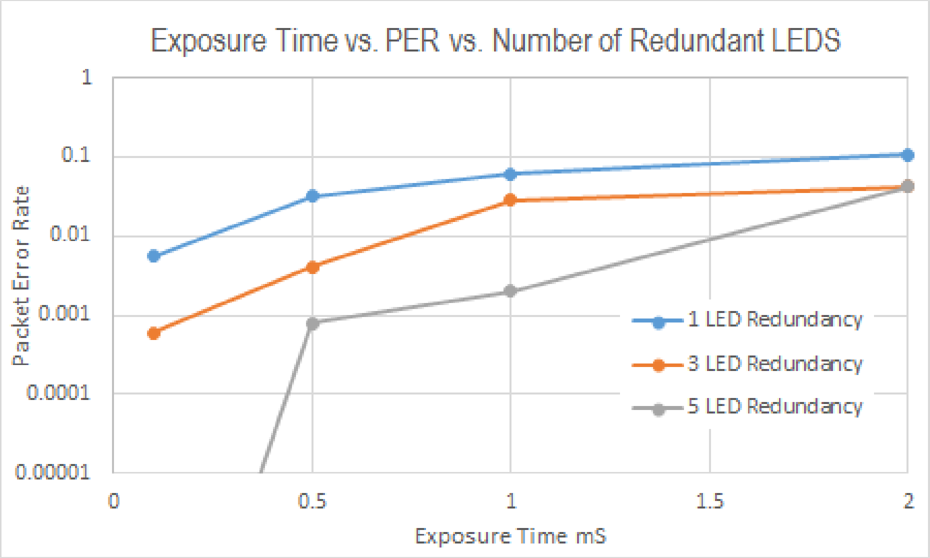




Name	PER vs. MIMO Redundancy
Exposure mS	0.1
SNR dB	100
Mark/Space Hz	105 & 120
PPM Offset	0
CRC-3	NO
Num MIMO LEDs	1, 3, 5
Duty Cycle	50%

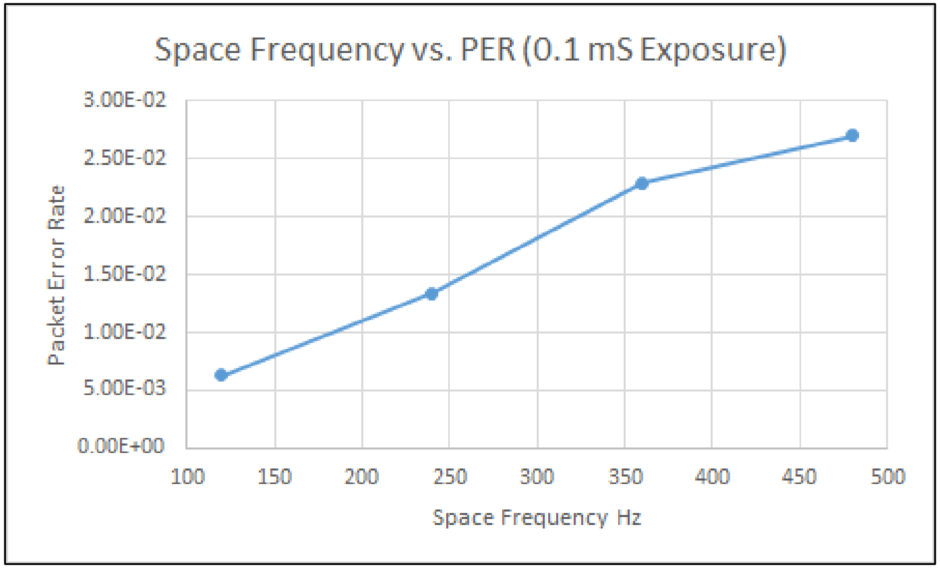
Name	PER vs. MIMO w/ CRC-3
Exposure mS	0.1
SNR dB	100
Mark/Space Hz	105 & 120
PPM Offset	0
CRC-3	YES
Num MIMO LEDs	1, 3, 5
Duty Cycle	50%

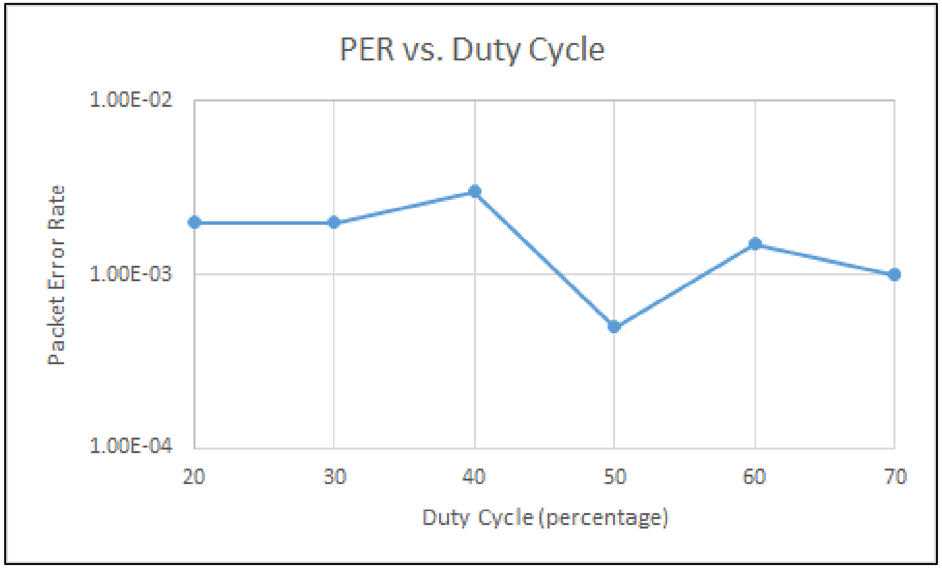
UFSOOK Performance Data



Name	PER vs. Exposure vs. MIMO
Exposure mS	0.1, 0.5, 1.0, 2.0
SNR dB	100
Mark/Space Hz	105 & 120
PPM Offset	0
CRC-3	YES
Num MIMO LEDs	1, 3, 5
Duty Cycle	50%

UFSOOK Performance Data

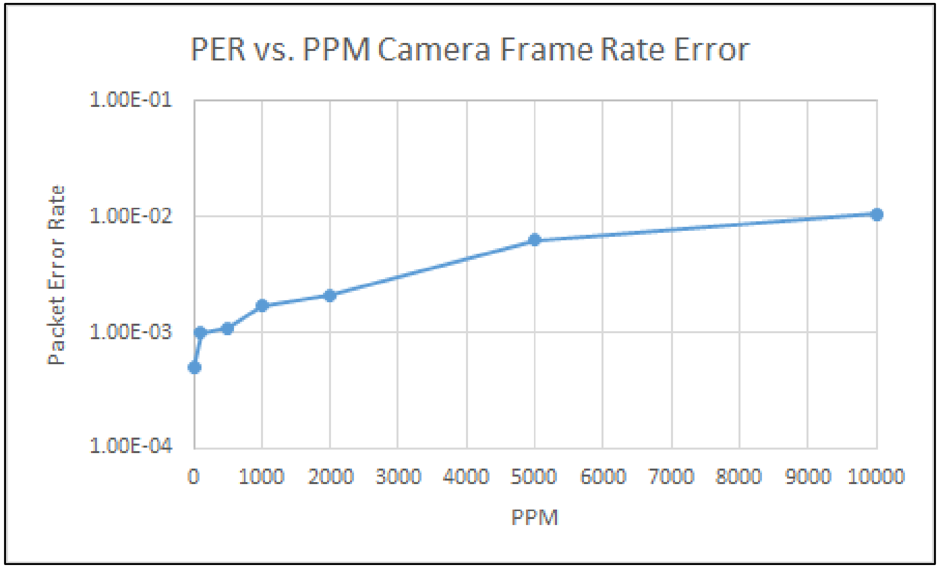


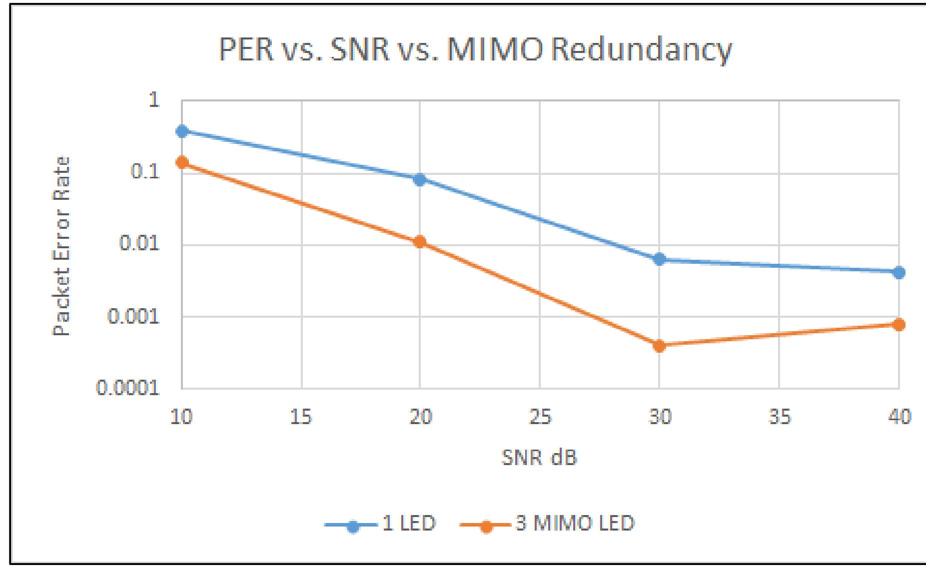


Name	PER vs. Space Freq Hz
Exposure mS	0.1
SNR dB	100
Space Hz	120, 240, 360, 480
PPM Offset	0
CRC-3	YES
Num MIMO LEDs	1
Duty Cycle	50%

Name	PER vs. Duty Cycle
Exposure mS	0.1
SNR dB	100
Mark/Space Hz	105 & 120
PPM Offset	0
CRC-3	YES
Num MIMO LEDs	3
Duty Cycle	20%, 30%, 40%, 50%, 60%, 70%

UFSOOK Performance Data





Name	PER vs. PPM @ 30 FPS
Exposure mS	0.1
SNR dB	100
Mark/Space Hz	105 & 120
PPM Offset	0,1e2, 2e2, 5e2,1e3, 2e3, 5e3,1e4
CRC-3	YES
Num MIMO LEDs	3
Duty Cycle	50%

Name	PER vs. SNR vs. MIMO
Exposure mS	0.1
SNR dB	10, 20, 30, 40
Mark/Space Hz	105 & 120
PPM Offset	0
CRC-3	YES
Num MIMO LEDs	1, 3
Duty Cycle	50%

Management of **Spatial Multiphase Coding** and **Temporal Multiphase Coding** …

1. **Spatial Multiphase Coding** is managed by a MIMO protocol presented in the next section.

2. **Temporal Multiphase Coding** is usually associated with transmission from a signal light (or relatively few lights) is not managed. Since the payload is assumed to be repetitive, the temporal multiple coding is applied at the transmitter without informing the receiver. The receiver just assumes that temporal coding is being applied when it ingests, for FEC purposes, multiple time domain repetitions.

Multiphase Repeat Coding Bit Rates

Repeat coding, with decoding via majority bit voting, can be used to combat improper sampling phase error. The resulting frame bit rate is given as:

$$R_{BIT} = N_{LEDS} \cdot \left(\frac{L - OH}{2L} \right) \cdot \left(\frac{R_{FRAME}}{N_{REPEATS}} \right)$$

where R_{BIT} is the bit rate, N_{LEDS} is the number of LEDs, L is the data payload length in bits, OH is the overhead number of bits, R_{FRAME} is the video frame rate of the camera, and $N_{REPEATS}$ is the number of repeats for the repeat coding.

Example …

- a sign has 100 LEDs, the camera rate is 30 fps, the frame length is 10 bits (i.e. 8 bit payload with two bits of overhead for the SFD), 3x repeat coding
- aggregate bit rate is 400 bps
- video recording for post processing is 20 frames long *(2/3 of a second to ingest)*

UFSOOK MIMO Operation

A MIMO protocol for LED light arrays

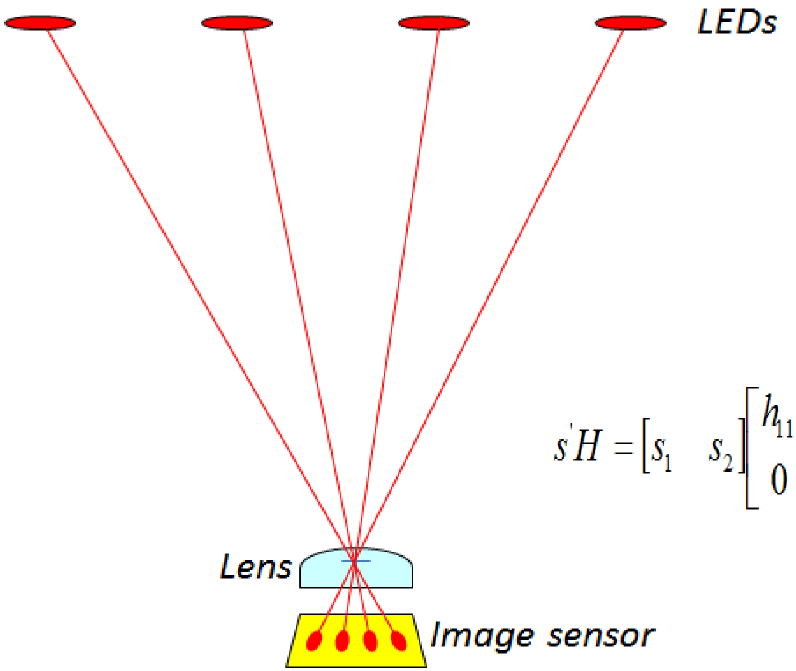
This proposals now shows a MIMO unidirectional data transmission protocol that can be used with an arbitrary LED array, including single LEDs (SIMO), transmitting to an image sensor array (camera) which typically has millions of “receiver” pixels. The protocol is constructed upon the use of efficient start-frame-delimiters (SFD) and data-delimiters (DD). Use is made of UFSOOK (undersampled frequency shift on-off keying) for the transmission of the delimiters and data bits.

One of the challenges of building a protocol based upon a cell phone camera is the sample rate of the camera is very low, typically 30 frames per second. This means that the maximum symbol rate that satisfies the Nyquist sampling criteria is half the frame rate, which in this case would be 15 bps. Such low symbol rates require a very efficient, non-elaborate protocol to keep transmission times within the limits of human endurance.

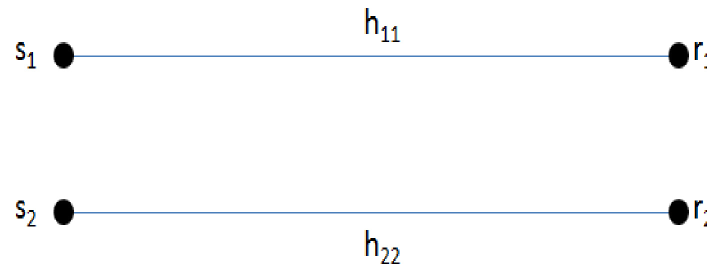
Space-time modulation of LEDs: problem statement

Below are three examples of an LED array: an LED bulb where the LEDs are relatively close together, a LED light panel where the LEDs are relatively far apart from each other, and a linear LED line bar. If the LEDs are doing spatial multiplexing then what is the protocol that allows us to de-multiplex and construct the original data message? This section of the proposal presents a protocol scheme to address these issues.

UFSOOK MIMO Operation



$$s'H = \begin{bmatrix} s_1 & s_2 \end{bmatrix} \begin{bmatrix} h_{11} & 0 \\ 0 & h_{22} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = r$$



The above figure shows the spatial resolution capability of a camera …

- multiple LEDs can be modeled as a degenerate case of the classic MIMO channel where there is little, if no, cross-coupling between the light sources
- each LED can transmit a parallel data stream that can be uniquely processed by the camera

Additional LEDs can be used for …

- spatial redundancy to increase transmission robustness
- spatial multiplexing to increase the throughput data rate
- a little of both to increase both the robustness and the throughput data rate

Four types of MIMO configurations …

1. Spatial Redundancy: transmit the same data on each LED in the array, perhaps with a different UFSOOK starting phase, to increase reliability and decrease sampling error.

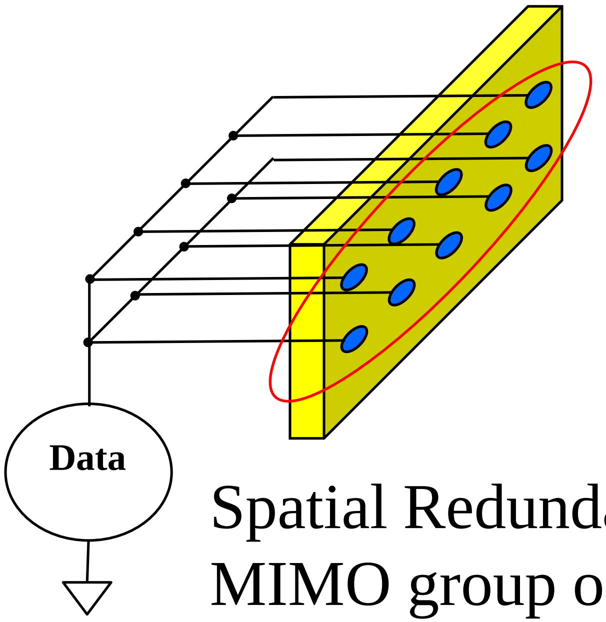
2. Spatial Multiplexing: de-multiplex a serial bit stream into multiple parallel bit streams which are parallel sent and then de-multiplexed back into a serial stream at the receiver.

3. Mixed Spatial Redundancy and Multiplexing: multiple sub-groups of LEDs are formed, each doing spatial redundancy, with spatial multiplexing being done across the sub-groups.

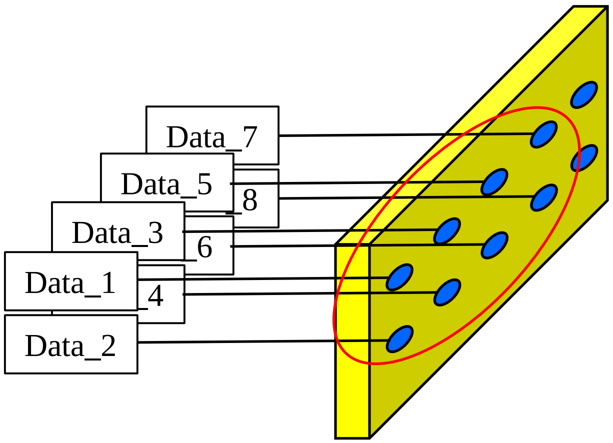
Four types of MIMO configurations (continued) …

4. Multiple User MIMO: multiple LEDs are viewed by the image sensor, each sending an independent data stream from an uncoordinated data source.

MIMO Group Definition

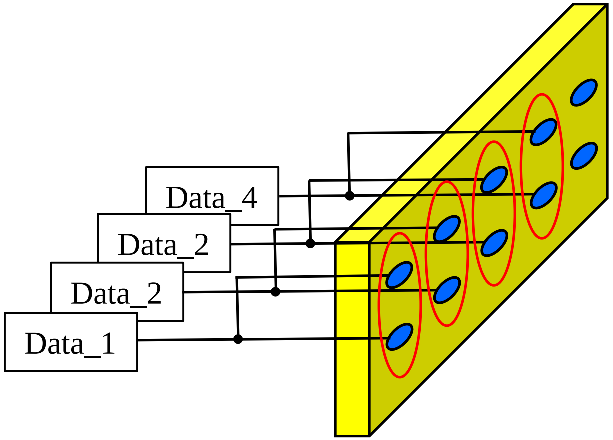


Spatial Redundancy
MIMO group of 10



Spatial Multiplexing
MIMO group of 8

MIMO Group Definition (cont.)



Mixed Spatial Redundancy and Spatial Multiplexing MIMO group

4 Spatial Multiplexing Sub-groups with 2 Spatial Redundant LEDs per Sub-group

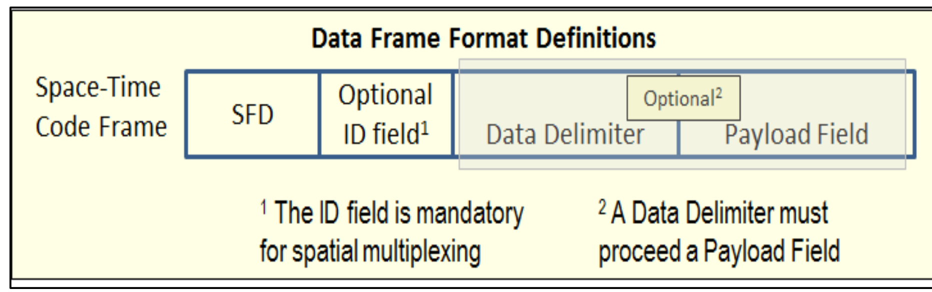
Space-Time Protocol for LED MIMO

Delimiter Definition

Define two types of delimiters to construct flexible data frames to accommodate space-time modulation.

SFD and DD Definitions

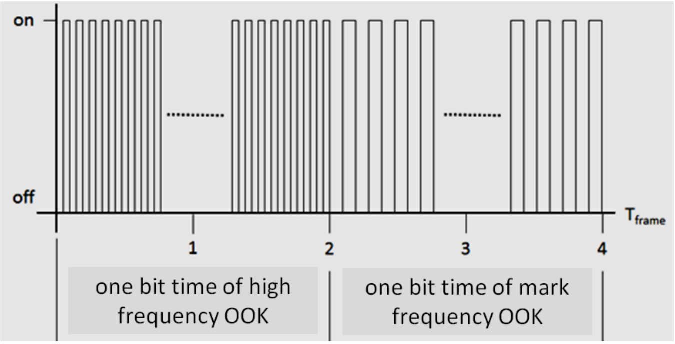
Start Frame Delimiter (SFD)	2 video frames HiRate Tone	2 video frames Mark Tone
Data Delimiter (DD)	2 video frames HiRate Tone	4 video frames Illegal Tone



The flowchart on the following page shows how these delimiters are used to realize an efficient LED MIMO protocol.

Start Frame Delimiter

- Two bit times long (i.e. four video frames)
- The first "bit" of the SFD is sent at an OOK frequency that cannot be followed by a normal smartphone grade image sensor.
- The pixel appears half ON (assuming 50% duty cycle)
- The next bit of the SFD is the transmission of the logic "1" mark OOK frequency $F_{mark} = \left(N \pm \frac{1}{2}\right) \cdot F_{FPS}$



Data Delimiter

- Six video frames long
- Two video frames of high frequency OOK
- Four video frames OOK transmitted at a frequency of $F_{tone} = \left(N \pm \frac{1}{4}\right) \cdot F_{FPS}$
- The last four video frames alias to $\frac{F_{FPS}}{4}$

Protocol Process Explanations

Identify all the LEDs in a MIMO group

All LEDs in a MIMO group simultaneously transmit the normal SFD. This identifies all the visible MIMO group members. The camera does a frame by frame scan of the lights looking for an occurrence when a group of lights are simultaneously flashing the SFD.

Decode ID fields

If the ID field is present, it will be between the SFD and DD. LEDs that are doing spatial redundancy all use the same ID number, which includes the null ID code (no ID field). For spatial multiplexing, there must be exactly 2^N LEDs in a MIMO group and each LED shall fill the ID field with its group sequence number. The anchor LED is assigned the value zero and the last LED in the group is assigned the value $2^N - 1$. The group sequence number indicates the order in which to concatenate the payloads so as to reconstruct the original message. If mixed spatial multiplexing with sub-group redundancy is being done then all the LEDs in a sub-group use the same ID number.

Protocol Process Explanations (cont.)

Spatial redundancy decoding of payloads and FEC across payloads

For spatial redundancy the payload sent by each light is identical and is decoded using the UFSOOK demodulation algorithm. Once the payload has been decoded, FEC can be accomplished by bit voting across all the received payloads in the MIMO group. There is no constraint on the number of LEDs in a spatial redundancy MIMO group.

Spatial multiplexing failure due to missing LEDs

As previously mentioned, the number of LEDs in a spatial multiplexing MIMO group must be 2^N (i.e. 1,2,4,8,16, etc.). For spatial multiplexing there should be 2^N sequentially decoded sequence ID numbers with no missing numbers. If there are not 2^N then some LEDs are missing from view and spatially multiplexed communications can not take place.

Protocol Process Explanations (continued)

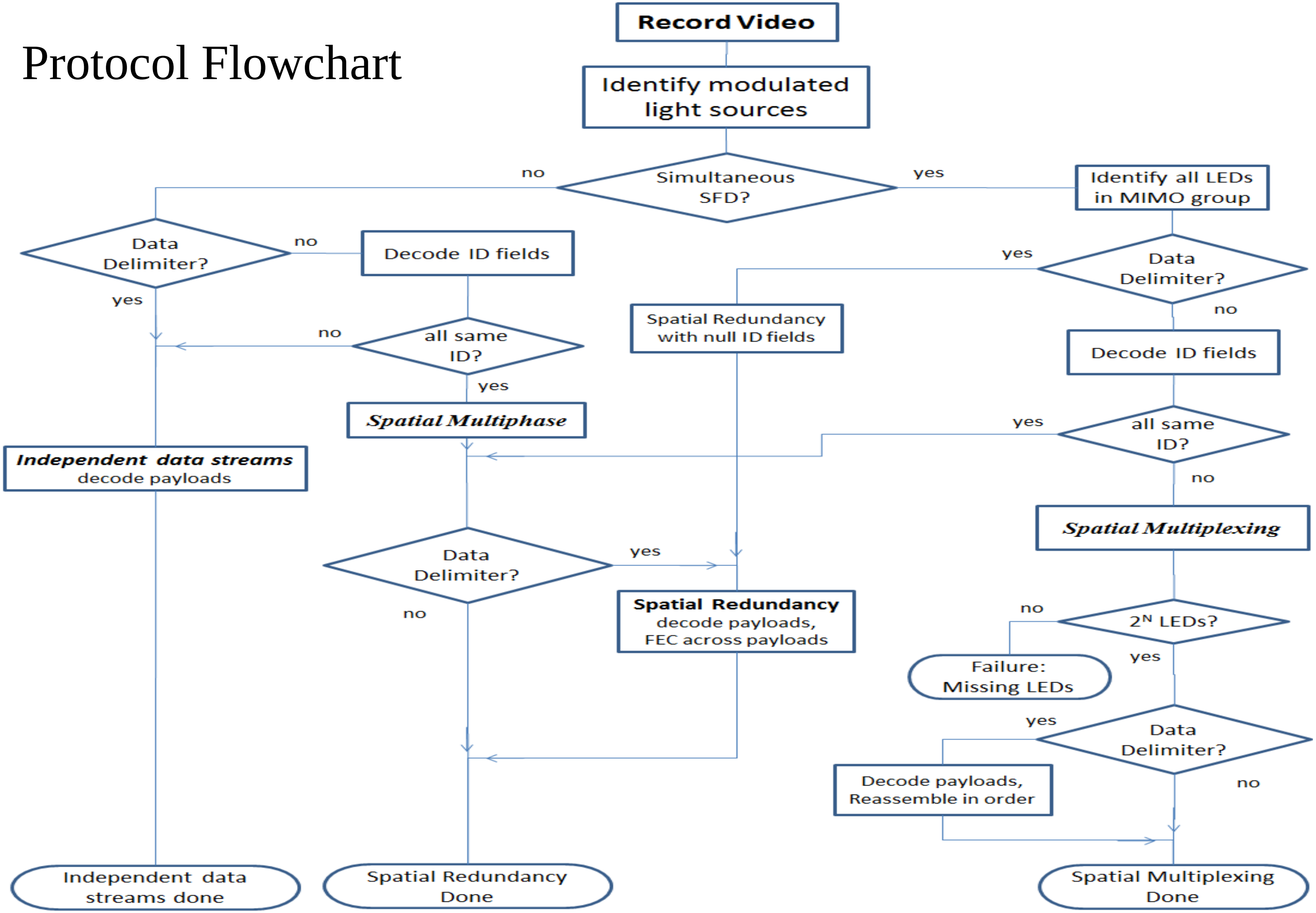
Spatial multiplexing decoding of payloads and reassembly of ordered payload

During spatial multiplexing, after the optional DD flash, each LED sends its optional payload. If there is a payload, it shall be preceded with a DD. All the individual payloads shall be the same length and shall be repetitive (cyclic); that is, the payload is repetitively and unchangingly sent each packet. Bit stuffing shall be used for any payloads requiring additional length.

Significance of anchor ID number

In a physically dispersed array of LEDs doing MIMO, the anchor ID represents the assigned physical location for the overall array; that is, it is the anchor location. All other LEDs in the array are not at this physical location. This concept is most useful for spatial multiplexing MIMO that is transmitting the anchor LED physical coordinates.

Protocol Flowchart



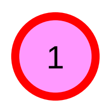
Example of a Single LED with only an ID

LED ID (8 bits): 4f

This would be sent from a single LED as …

LED: SFD,4f

The total bits sent would be: 2+8=10. At 15 bps the ID could be sent in 0.67 seconds. Unless standardized by industry agreement, the number of bits in the ID is variable and is determined by counting the number of bits between the SFDs.



Example of a Single LED with an ID and a payload

LED ID (hex): a7
Payload (hex): b

This would be sent from a single LED as …

LED: SFD,a7,DD,b

The total bits sent would be: 2+8+3+4=17. At 15 bps the ID with payload could be sent in 1.13 seconds.

Example of a Single LED with no ID and a payload

LED ID (hex): none
Payload (hex): b

This would be sent from a single LED as …

LED: SFD,DD,b

The total bits sent would be: 2+3+4=9. At 15 bps this would take 0.6 seconds.

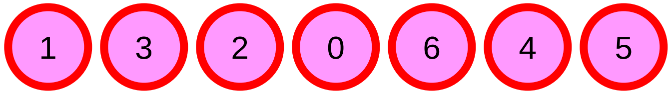
MIMO example of an IPv6 Address

Example IPv6 address: [3ffe:1900:4545:0003:0200:f8ff:fe21:67cf]

This could be sent with 8 LED, each with a 16 bit payload

LED$_0$: SFD,0,DD,67cf
LED$_1$: SFD,1,DD,fe21
LED$_2$: SFD,2,DD,f8ff
LED$_3$: SFD,3,DD,0200
LED$_4$: SFD,4,DD,0003
LED$_5$: SFD,5,DD,4545
LED$_6$: SFD,6,DD,1900
LED$_7$: SFD,7,DD,3ffe

The totals bits sent would be: 2+3+3+16=24. At 15 bps the whole 128 IPv6 address could be sent in 1.6 seconds.



order can be completely random

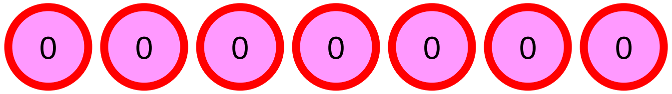
MIMO spatial redundancy

Payload (hex): 9f

This could be sent with 8 LED, each with a 16 bit payload

LED_0: SFD,0, DD,9f
LED_1: SFD,0, DD,9f
LED_2: SFD,0, DD,9f
LED_3: SFD,0, DD,9f
LED_4: SFD,0, DD,9f
LED_5: SFD,0, DD,9f
LED_6: SFD,0, DD,9f
LED_7: SFD,0, DD,9f

The totals bits sent would be: 2+3+3+8=16. At 15 bps the parallel payload sequence could be sent in 1.07 seconds.



Example of uncoordinated individual LEDs with ID and payload

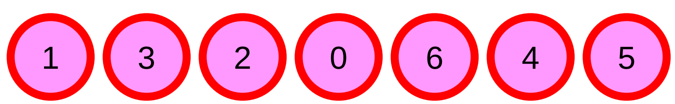
Payload (hex): e

Assume 8 un-coordinated LEDs, each with a 16 bit payload

LED$_0$: SFD,0, DD,e
LED$_1$: 1, DD,e,SFD
LED$_2$: DD,e,SFD,2
LED$_3$: e,SFD,3, DD
LED$_4$: DD,e,SFD,4
LED$_5$: SFD,5, DD,e
LED$_6$: DD,e,SFD,6
LED$_7$: 7, DD,e,SFD

For uncoordinated packets, the starting phases are random but the packets are cyclic, meaning for a minimum ingest time the full packet can be "un-wrapped" and recovered.

The totals bits sent would be: 2+3+3+4=12. At 15 bps the bit sequence, including ID and payload, could be sent in 0.8 seconds.



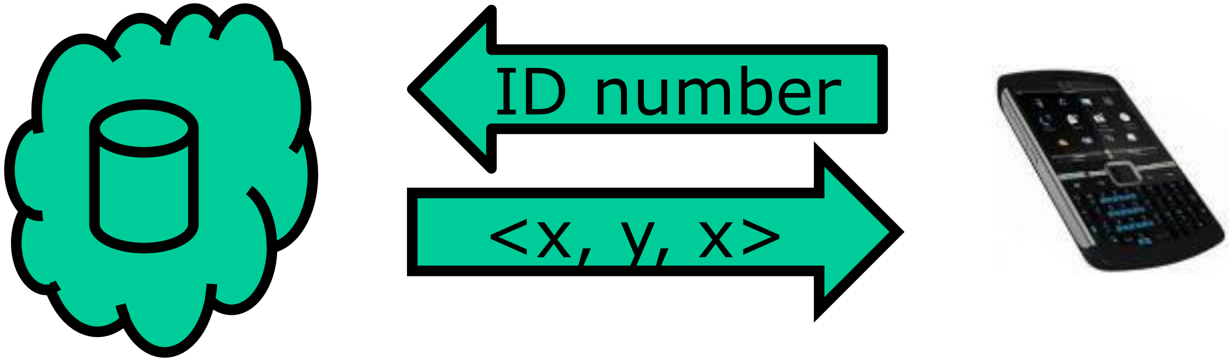
order can be completely random

Methods to Prevent Ambiguous LED IDs

ID code to absolute position lookup

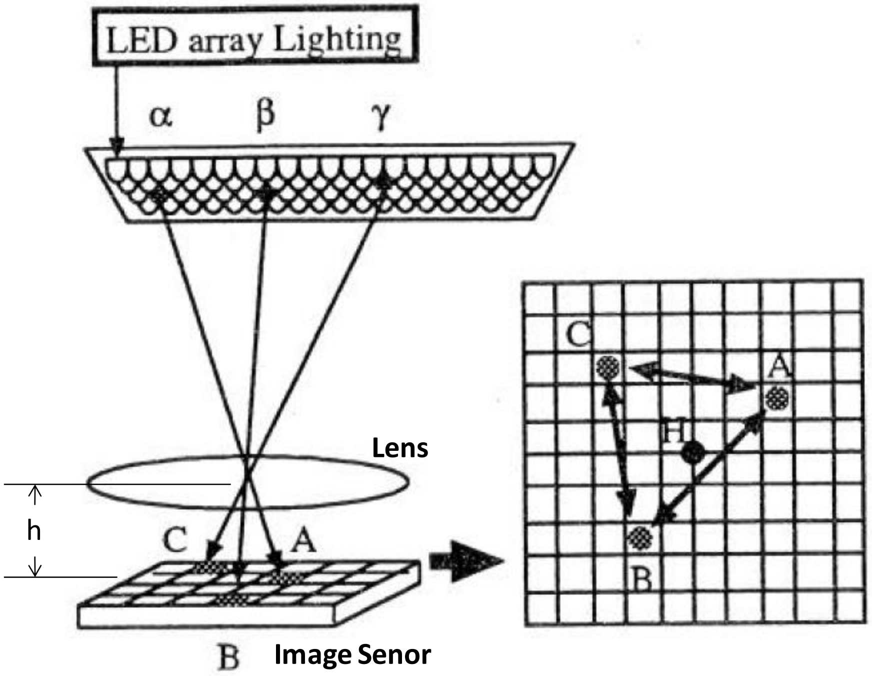
Once each lights’ ID code is obtained, a data base is accessed that provides the <x,y,z> room coordinates for each particular light.

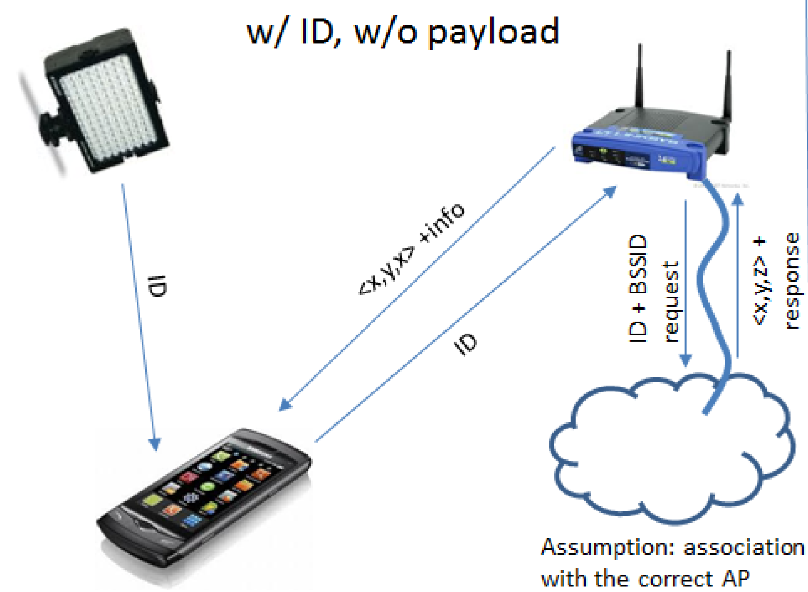
The database can be accessed over the internet, via a 3G or WiFi connection.

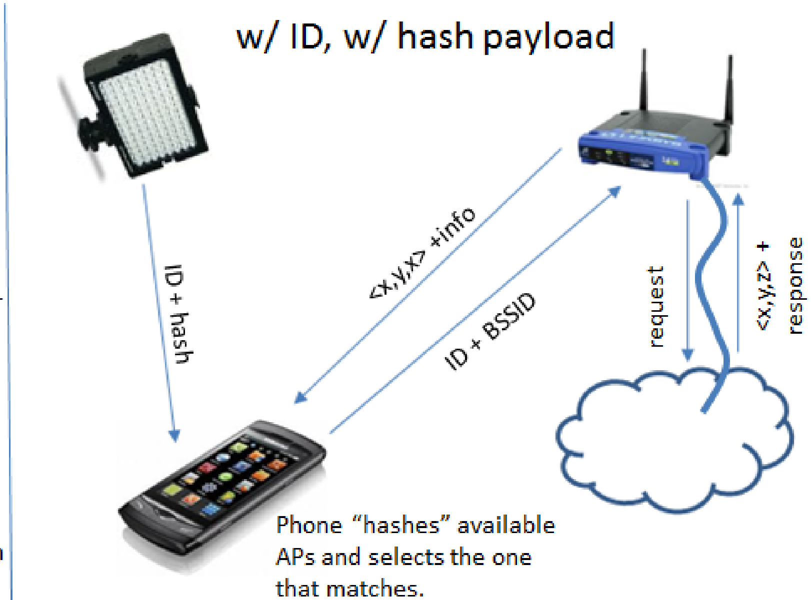


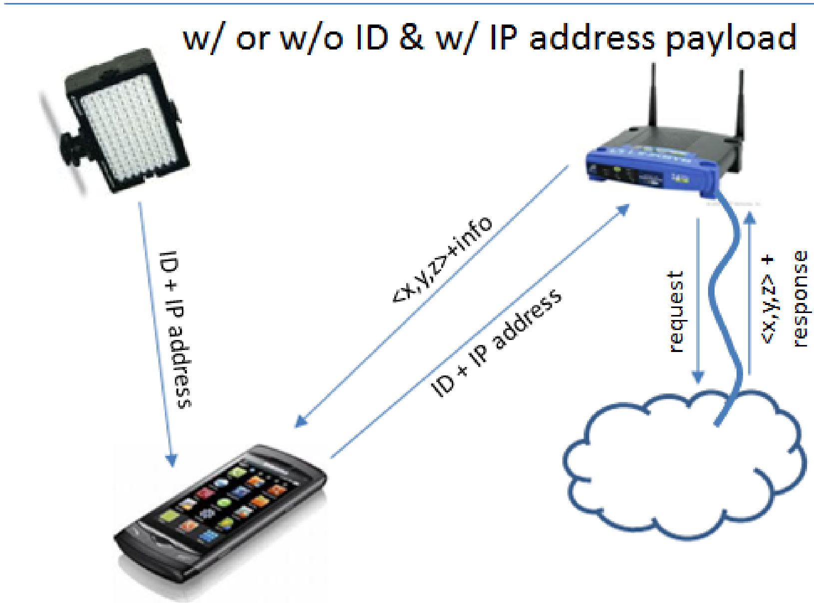
Once the LED light <x,y,z> positions are known, photogrammetric processing techniques can be used to solve the location problem

Geometrically solve the similar triangle problem posed by the light rays projecting through the lens upon the image sensor.









We can resolve short ID location ambiguity by either:

i) appending the ID to the access point MAC address (BSSID) ... the combination provides a globally unique identifier

ii) using an ID and BSSID hash to make sure we access through a specific AP ... this specific AP provides specific info to the user

iii) w/ or w/o an ID and sending an IP address payload ... great when we want additional info provided by a specific server

A Parenthetical Signal Processing Comment

In principle, for any of the described protocol techniques, the signal processing most likely will be non-real time; that is, a video of sufficient length is recorded and then post-processing identifies the location of all SFDs and isolates a single packet. (Keep in mind that the packet is "circular" so the recording start-time is arbitrary).

The problem with the above statement is that one does not know the packet length a priori, so we are forced to record an excessively long video to cover the worst case packet length.

There are three plausible solutions:

1. the length of the packet for a particular application is set by industry agreement (i.e. standard) … *this is the most efficient approach*
2. the SFDs are processed real time and the recording is stopped once it is recognized that a packet has been ingested … *most flexible but not the most efficient*
3. the length of the packet is provided a priori based upon the location. For example, you are in store XYZ and this store uses a certain packet configuration. This information could also be downloaded to the user over an RF network such as WiFi upon entering the premises.

This proposal supports the 3rd method via the PIB.

The sections of this proposal dealing with the MAC will show how the PIB is used to do the following:

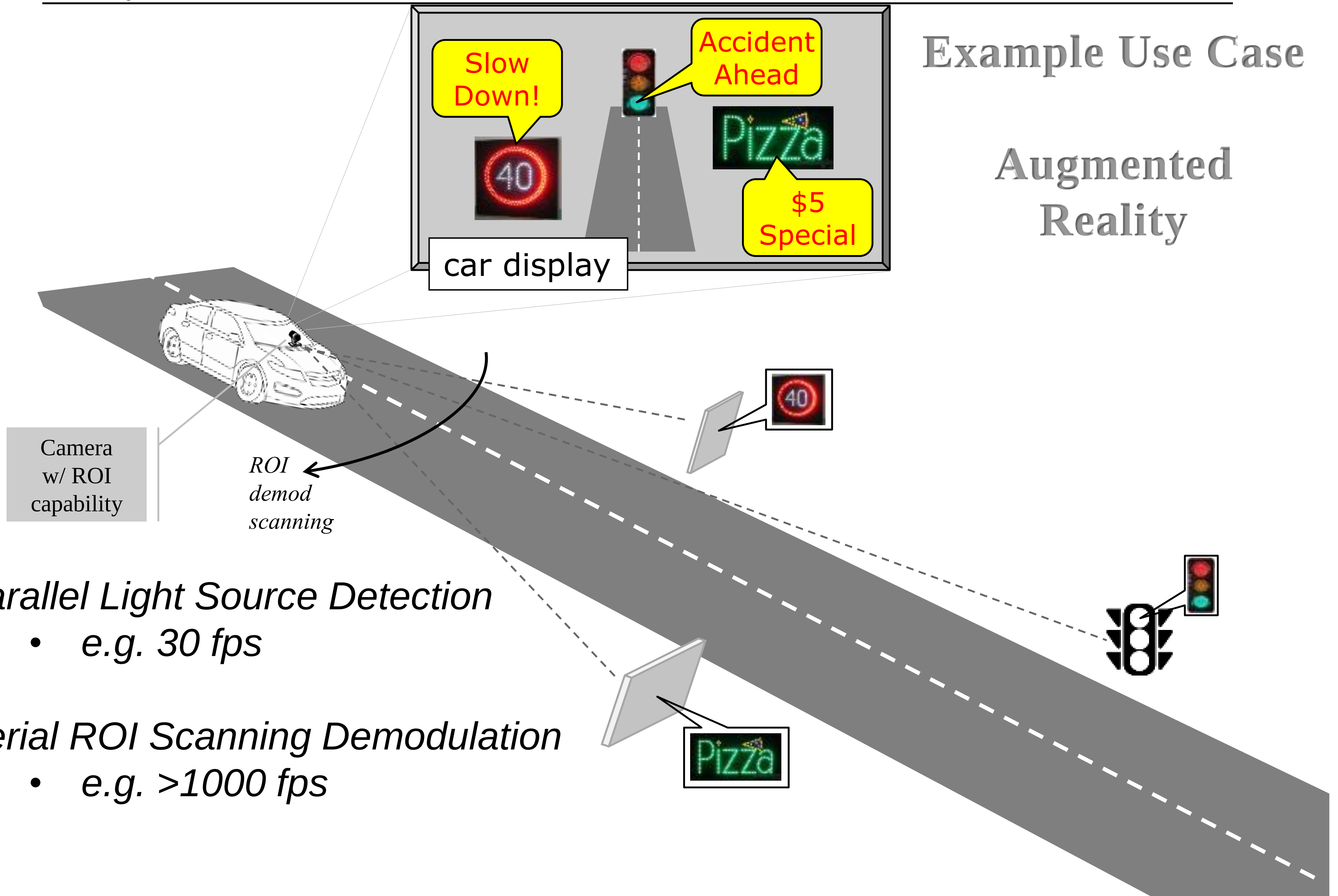
1. Append an LED ID to a WiFi SSID to eliminate location ambiguity.

2. Indicate APP specific payload length.

High Frame Rate Nyquist Region-of-Interest Sub-sampling

Example Use Case

Augmented Reality



Parallel Light Source Detection

- *e.g. 30 fps*

Serial ROI Scanning Demodulation

- *e.g. >1000 fps*

High Frame Rate Nyquist Region-of-Interest Sub-sampling

Problem to solve: given a scene with numerous LED lights, how does one know which lights are modulated with data and which are not? Hunting through all the possible light sources in a scene is time consuming. What is needed is a method for a modulated light to send a low rate indication beacon that can be processed by a low frame rate camera, viewing the whole scene, to ascertain which light sources actually are transmitting high rate data.

Objective: We want to ingest the simultaneous LED transmission of two data streams (i.e. a high rate data stream and a low rate data stream), both intended to be ingested by an image sensor. By adjusting the image sensor exposure time (integration time) the image sensor can choose which data stream to ingest and process. The high rate data stream is selected by setting the image sensor integration time to be short, and the low rate data stream is selected by setting the image sensor integration time to be relatively long.

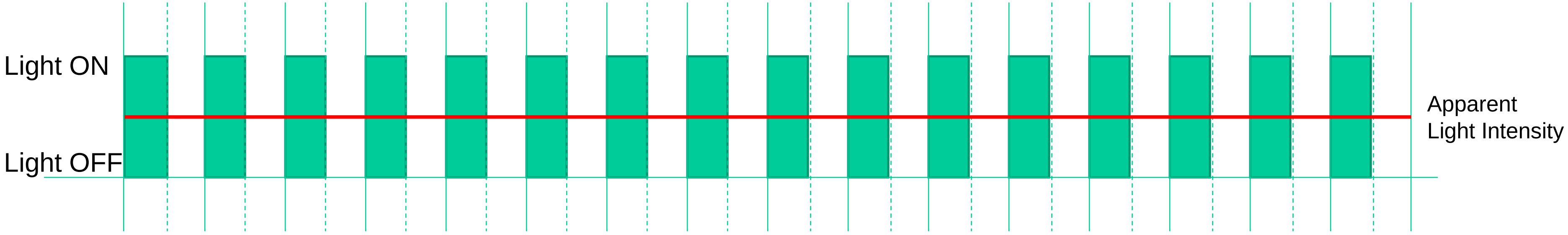
Assumption: assume an LED light is modulated with data (using ON-OFF keying) at a high bit rate frequency such that there is no noticeable flicker. In addition, assume that in the field of view of the camera’s image sensor (i.e. in the scene) there are multiple LED lights visible.

Solution: change the duty cycle of groups of high rate pulses (via pulse width modulation) over time such that a low frequency UFSOOK amplitude envelope is imparted on the high rate data which has the effect of making the LED light appear to change intensity in a low frequency manner while still transmitting high rate data. The low frequency amplitude envelope will appear to make the LED light flicker (twinkle) which can be used as a beacon to indicate which LED lights are actually sending high rate data.

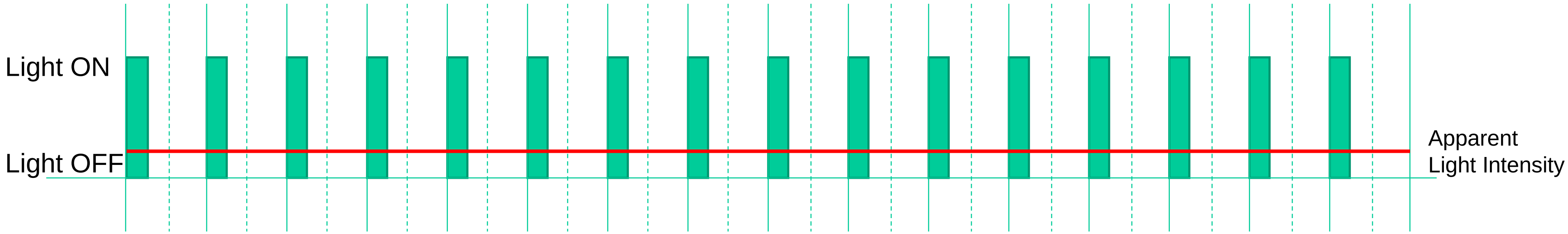
We Call This: AM by VPPM and the induced AM envelope we call “twinkle”

In visible light communications an LED light is pulsed ON and OFF to send data. The pulsing may be at a rate that can be seen or not be seen by a human being.

If the pulsation frequency of the light exceeds the flicker perception frequency of the human eye then a person sees only the average light intensity of the LED.

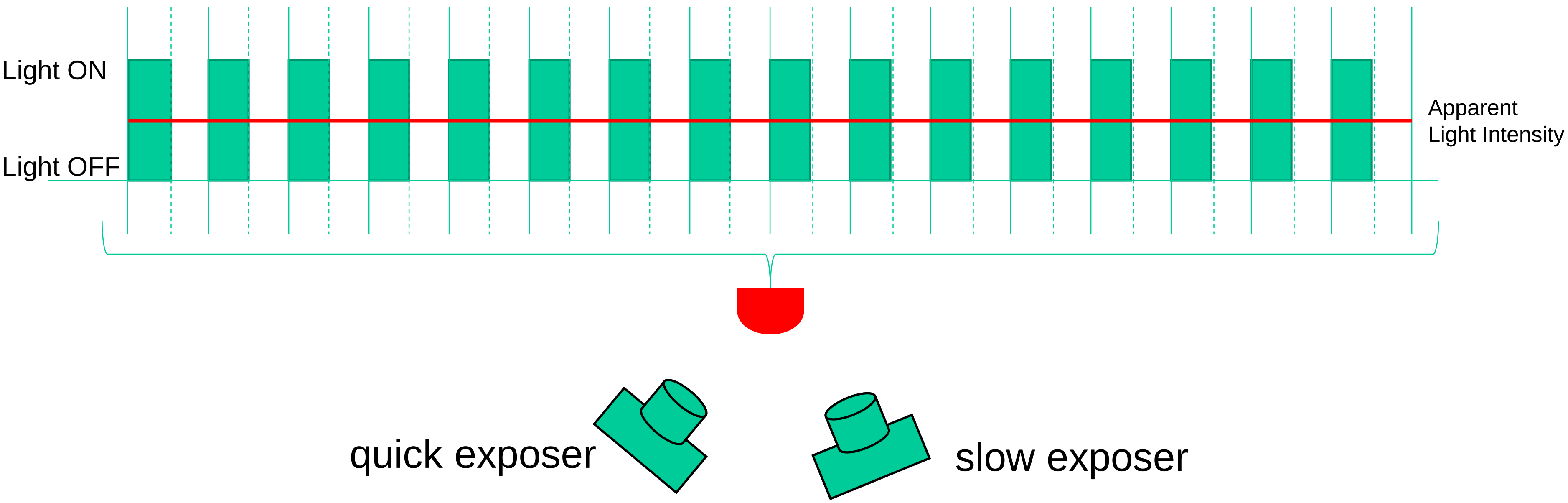


If the duty cycle of pulsation, at a frequency that exceeds the flicker perception frequency, is changed then the LED light appears to proportionally change average intensity.



When viewed by a camera instead of the human eye, the same principle applies and is controlled by the setting of the camera exposer integration time. If the exposure time is long then the camera can only respond to low frequency flicker. If the camera exposer time is sufficiently short then it can respond to the individual high rate pulses.

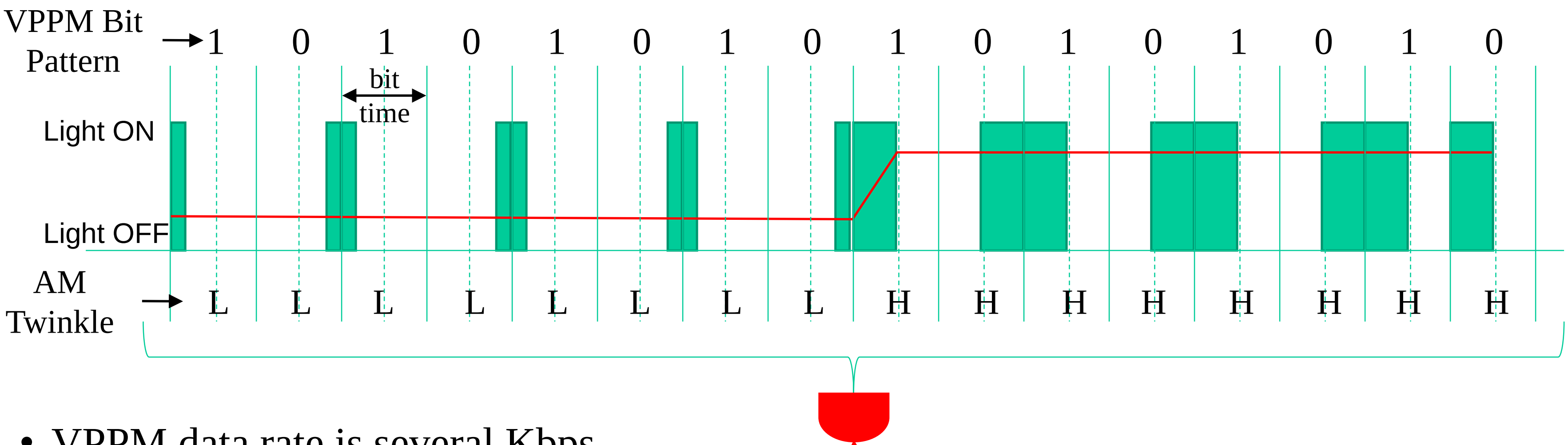
Generally a camera, when processing a complete frame of pixels, has a relatively low frame rate – typically 30 frames per second – which sets the Nyquist sampling rate for the camera operating in this mode. However, by only processing a subset of the pixels, called the region-of-interest (ROI), the camera can significantly increase the frame rate (albeit with less pixels) such that the resulting Nyquist sampling rate is high enough to be able to individually decode the high rate data pulses.



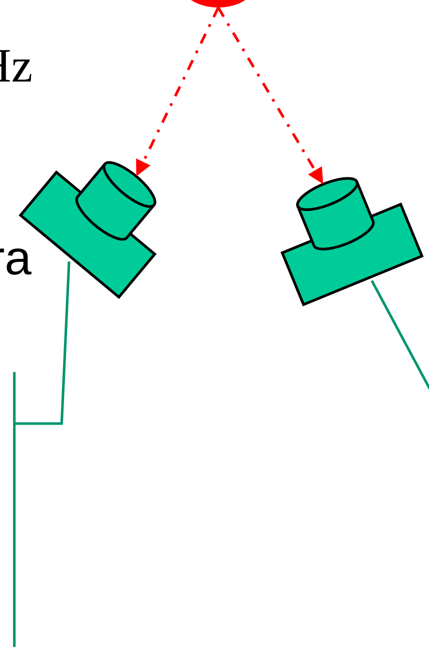
Two cameras observe the pulsating light. The first camera has a quick exposer such that it can decode the high rate pulsing. A second camera has a slow exposer time such that it only perceives a light with a constant intensity.

If the high rate pulse duty cycle is varied in a periodic manner, at a sufficiently low frequency, then the camera with a quick exposer still only sees the high rate pulses, while the camera with the slow exposer time perceives that the light has a time variant intensity at this lower frequency.

Composite ComCom Data Waveform with Twinkle



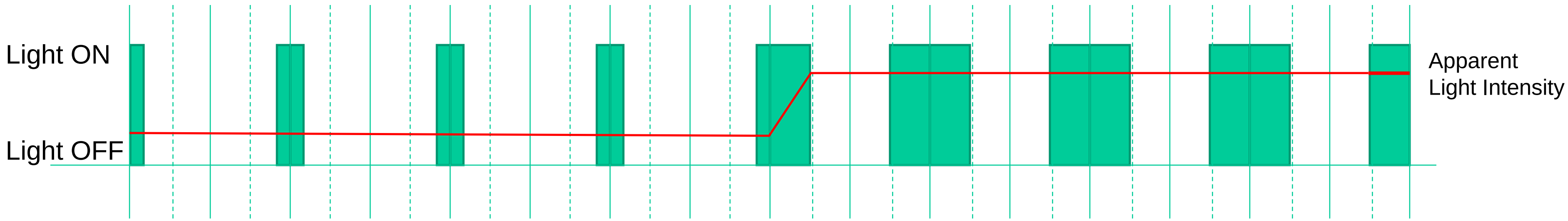
- VPPM data rate is several Kbps
- AM Twinkle Frequency is ~127.5 Hz



High speed (e.g. 10 kfps) camera detects the individual VPPM pulses.

Low speed (30 fps) camera detects the 127.5 Hz envelope due to pulse width modulation. Aliasing makes light twinkle at 7.5 Hz rate.

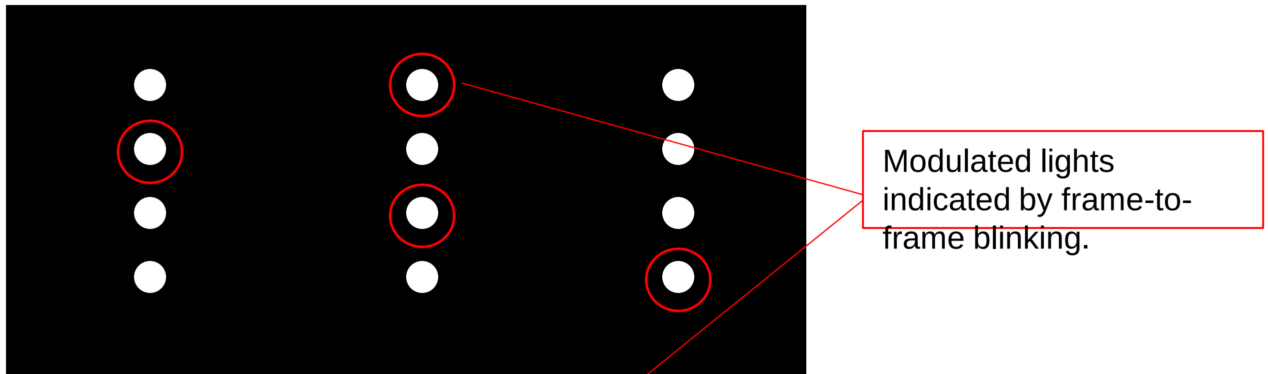
The high rate pulses can use a form of pulse position modulation called variable pulse width modulation (VPPM) to encode data while accommodating PWM (this can also be thought of as a form of Manchester encoding with dimming).



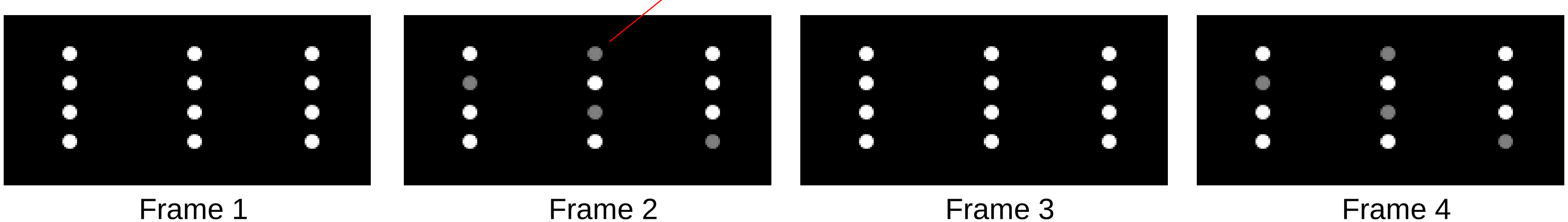
If a pulse occurs in the first half of a bit time (light ON) then that is encoded as one logic level. If the pulse occurs in the second half of a bit time then that is encoded as the other logic level. The pulse width is not relevant - just the pulse position within the bit time.

In this manner a camera with a sufficiently long exposure time (or human eye) will see a light that is flickering. Yet a camera with a sufficiently short exposure time, along with knowledge of bit timing, can ascertain the position of the pulse within the bit time slot and hence make a bit decision.

The ability to identify modulated lights in an image is crucial to the deployment of camera technology that leverages region-of-interest (ROI) sub-sampling for demodulating modulated lights because it is only after we've determined the region-of-interest can the ROI sub-sampling begin.



The techniques shown in this invention disclosure allows a full-frame camera to record a short video of a scene and then ascertain which lights are modulated by observing which lights appear to be blinking. These are the lights that need ROI sub-sampling.



ROI Nyquist Sampled VPPM

Modulation is a Modified Version of IEEE80.15.7-2011 PHY 1, VPPM Waveform

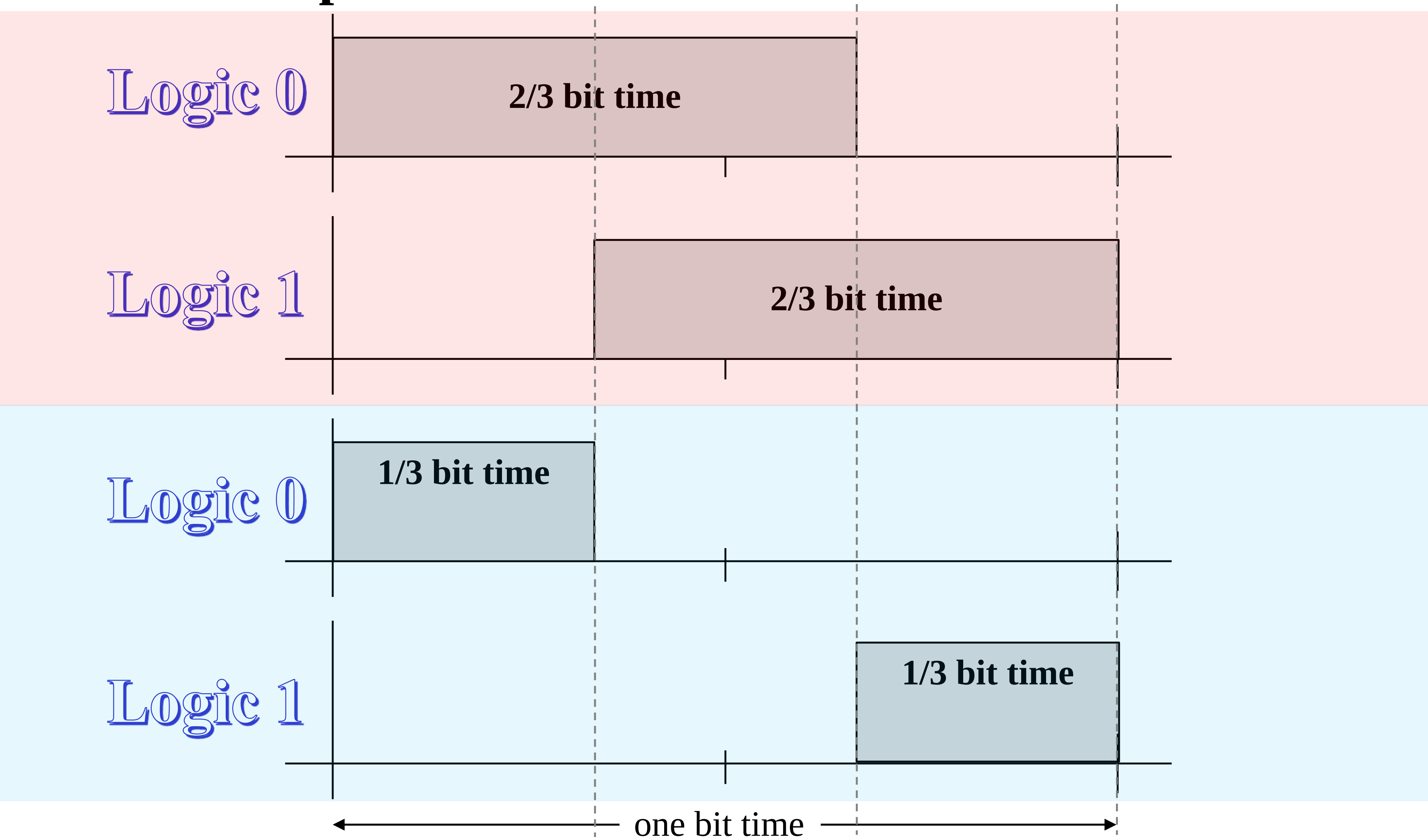
PPDU Format

Start Frame Delimiter	PHY Header	HCS	
SFD	PHR		PHY payload

This part of the proposal on Nyquist sampled communications considers the communications to be quasi-synchronous. No effort is made to actually synchronize the receiver timing to the transmitter timing; rather, the preamble is oversampled and a down-sampling phase is selected that offers the best performance for the given sample phases.

In order to maximized the throughput, it is required that we minimize the overhead; specifically, we want to keep the SFD and PHR to a minimum length.

Composite Waveform VPPM Bit Definition



How this generates the “twinkle” …

1. The desired “aliased twinkle frequency” is ¼ the camera frame rate. For a 30 fps camera this would be 7.5 Hz.
   - Why ¼ the camera frame rate? Because at 4 samples per cycle we are guaranteed to generate the twinkle frequency even if every other sample falls on a transition boundary.

2. The possible AM envelope frequencies are $F_{AM} = \left(n \pm \frac{1}{4}\right) \cdot F_{fps}$ where *n* is an integer.
   - For example, given a 24 fps camera and *n*=4 one possible amplitude modulation frequency would be 4.25*24=102 Hz

3. Using the VPPM bit definitions, alternately transmit 2/3 duty cycle bits for half an AM envelope cycle followed by 1/3 duty cycle bits for the second half AM envelope cycle. Round to an integer number of transmitted bits per half cycle.
   - For the above example, we’d alternate between duty cycles every 1/204 seconds.

Transmitter Power Distribution …

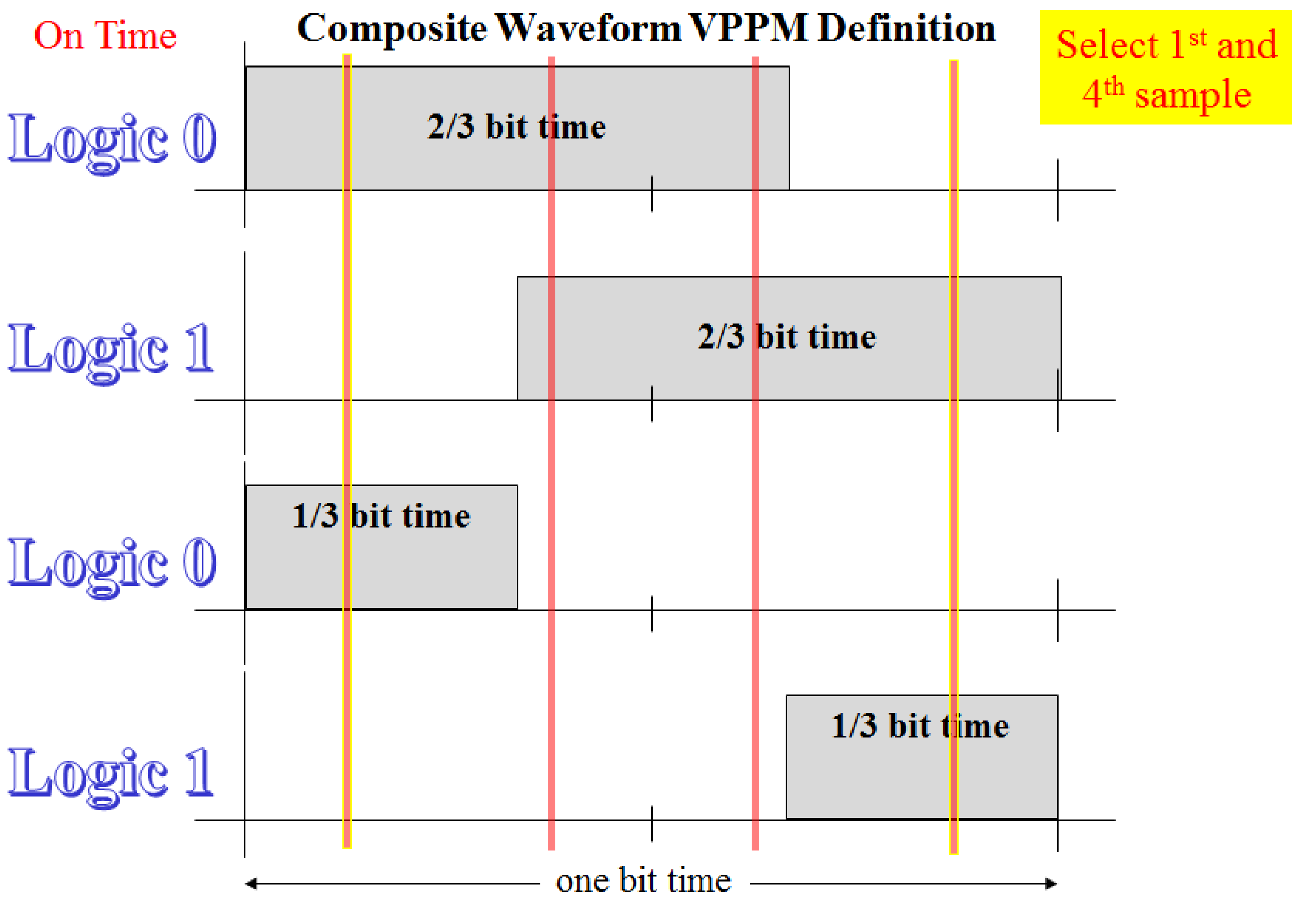
1. Light steady ON (non-modulated) … $P_{max} = V_{LED} \cdot I_{LED}$

2. Modulated Light … $P_{mod} = \frac{1}{2} \cdot \frac{2}{3} \cdot P_{max} + \frac{1}{2} \cdot \frac{1}{3} \cdot P_{max} = \frac{1}{2} \cdot P_{max}$

3. Extinction Ratio of AM Envelope … 50%

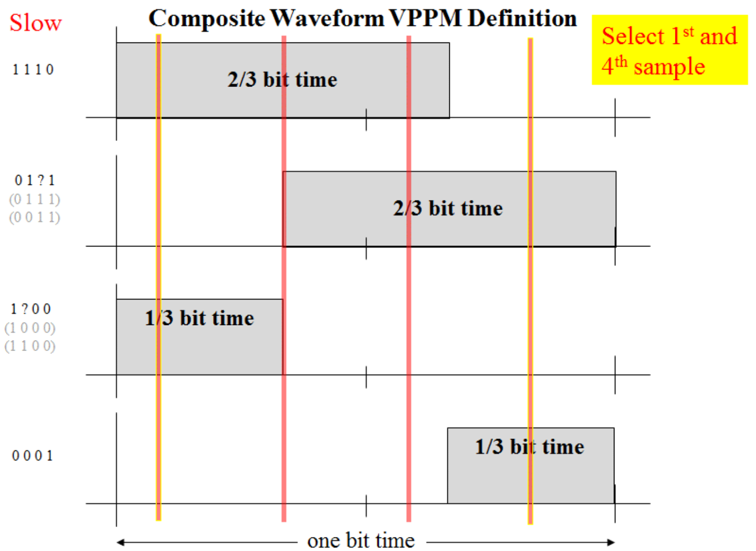
4. VPPM energy ingested by camera … dependent on camera exposure setting
    - For a camera exposure set to ¼ of a PPM bit time, with a down sample ratio of 2, $P_{camera} = \frac{P_{max}}{2 \cdot Loss_{path}}$

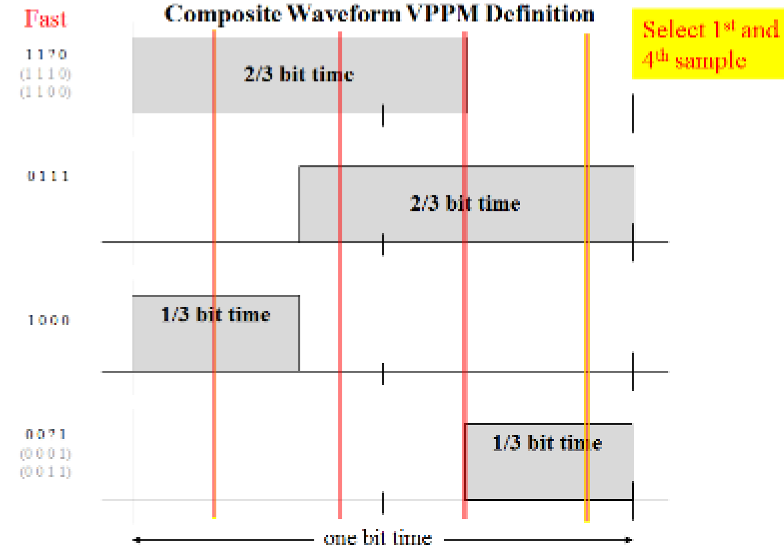
Question: how does one select the optimum camera sample phase given four asynchronous samples per bit?

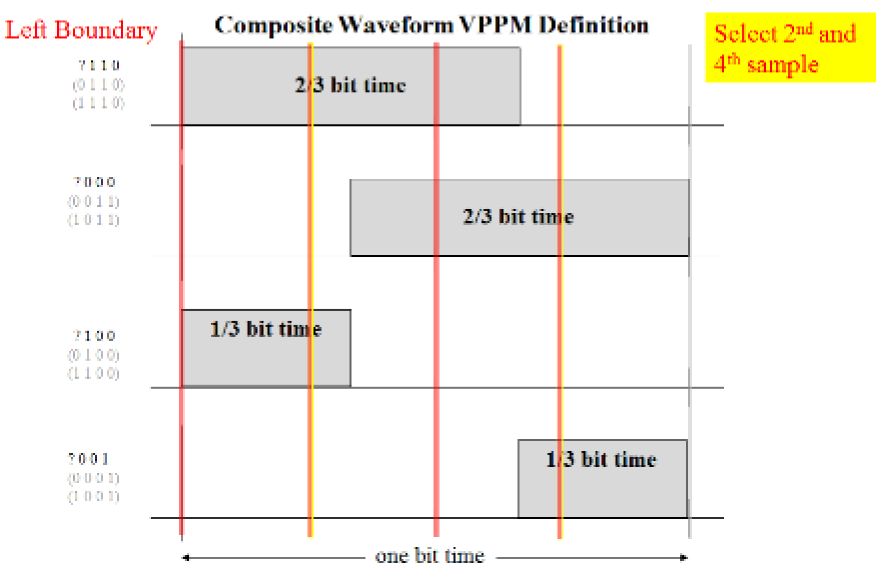


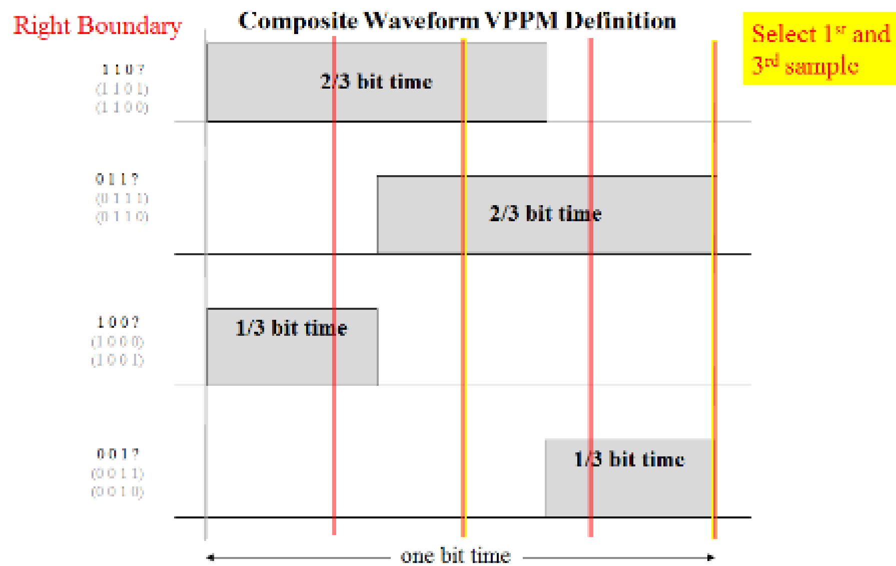
Answer: need to craft an SFD that will allow us to identify the beginning of a data frame and also allow us to optimize the camera sample phase.

Other possible sample phases

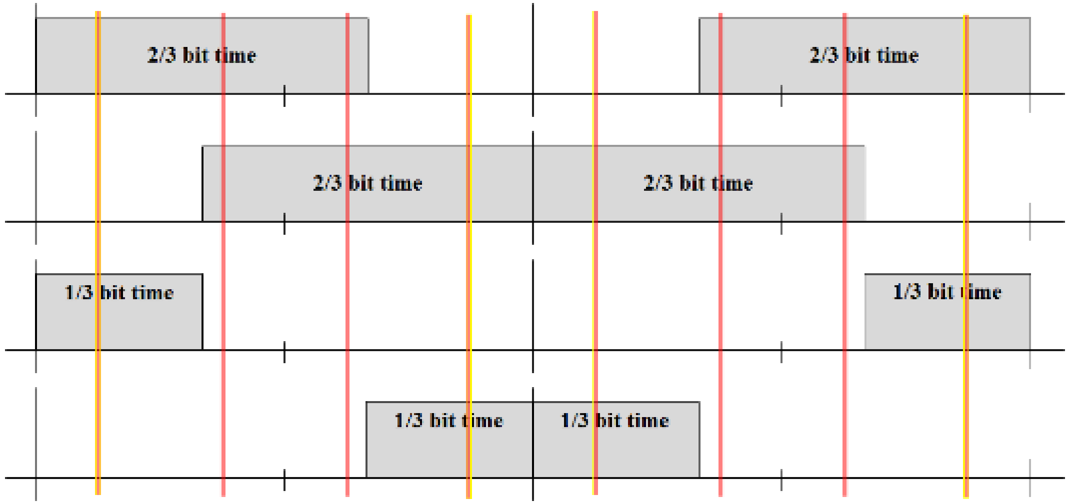








Determine Maximum Run Length

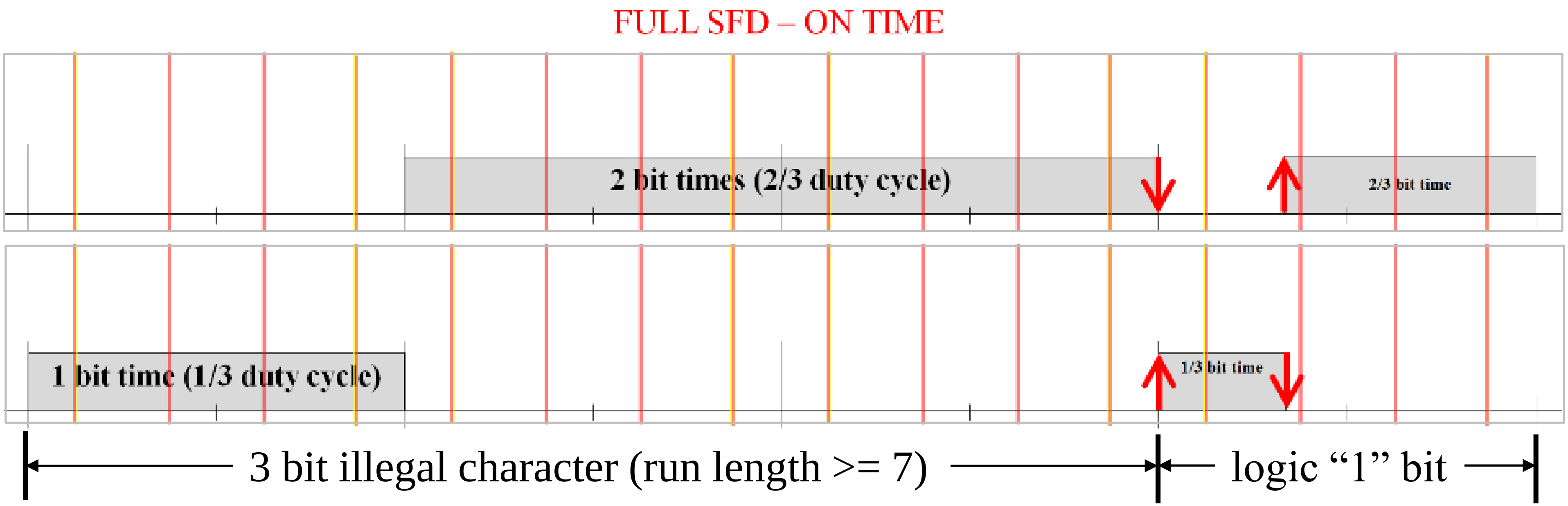


	70% Dimming					30% Dimming				
Logic String	On Time Chips	Slow Chips	Fast	Left Boundary	Right Boundary	On Time Chips	Slow Chips	Fast	Left Boundary	Right Boundary
00	11101110	11101110	11101110 11001100	01100110 11101110	11011101 11001100	10001000	10001000 11001100	10001000	01000100 11001100	10001000 10011001
01	11100111	11100111 11100011 Run: 3 zeros	11100111 11000111 Run: 3 zeros	01100011 11101011 Run: 3 zeros	11010111 11000110 Run: 3 zeros	10000001 Run: 6 zeros	10000001 11000001 Run: 6 zeros	10000001 10000011 Run: 6 zeros	01000001 11001001 Run: 5 zeros	10000011 10010010 Run: 5 zeros
10	01111110 Run: 6 ones	01111110 00111110 Run: 6 ones	01111110 01111100 Run: 6 ones	00110110 10111110 Run: 5 ones	01111101 01101100 Run: 5 ones	00011000 Run: 2 ones	00011000 00011100 Run: 3 ones	00011000 00111000 Run: 3 ones	00010100 10011100 Run: 3 ones	00111000 00101001 Run: 3 ones
11	00010001 Run: 3 zeros	01110111 00110011	01110111	00110011 10111011	01110111 01100110	00010001	00010001	00010001 00110011	00010001 10011001	00110011 00100010

Prior to selection of the decimation phase, a <u>legal</u> PPM code never has a run of more than 6 successive same polarity chips

Therefore, an <u>illegal</u> code would consist of 7 or more successive same polarity chips.

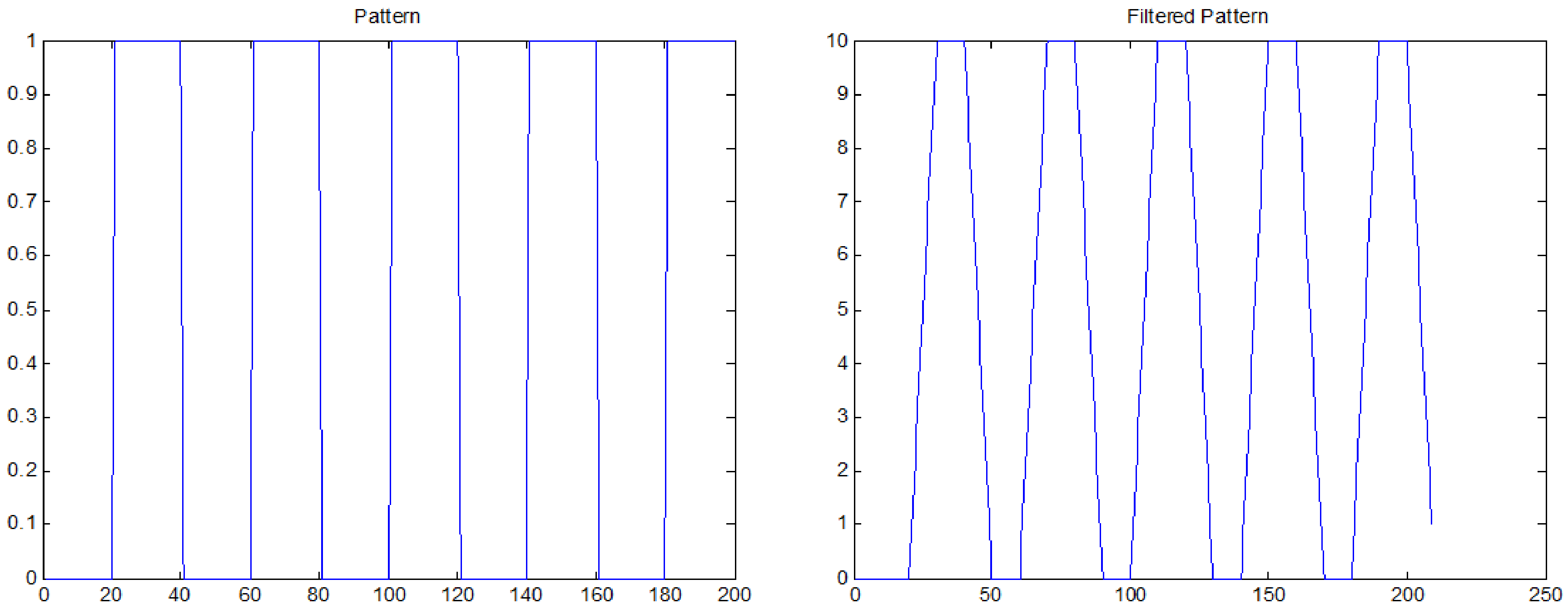
Proposed SFD



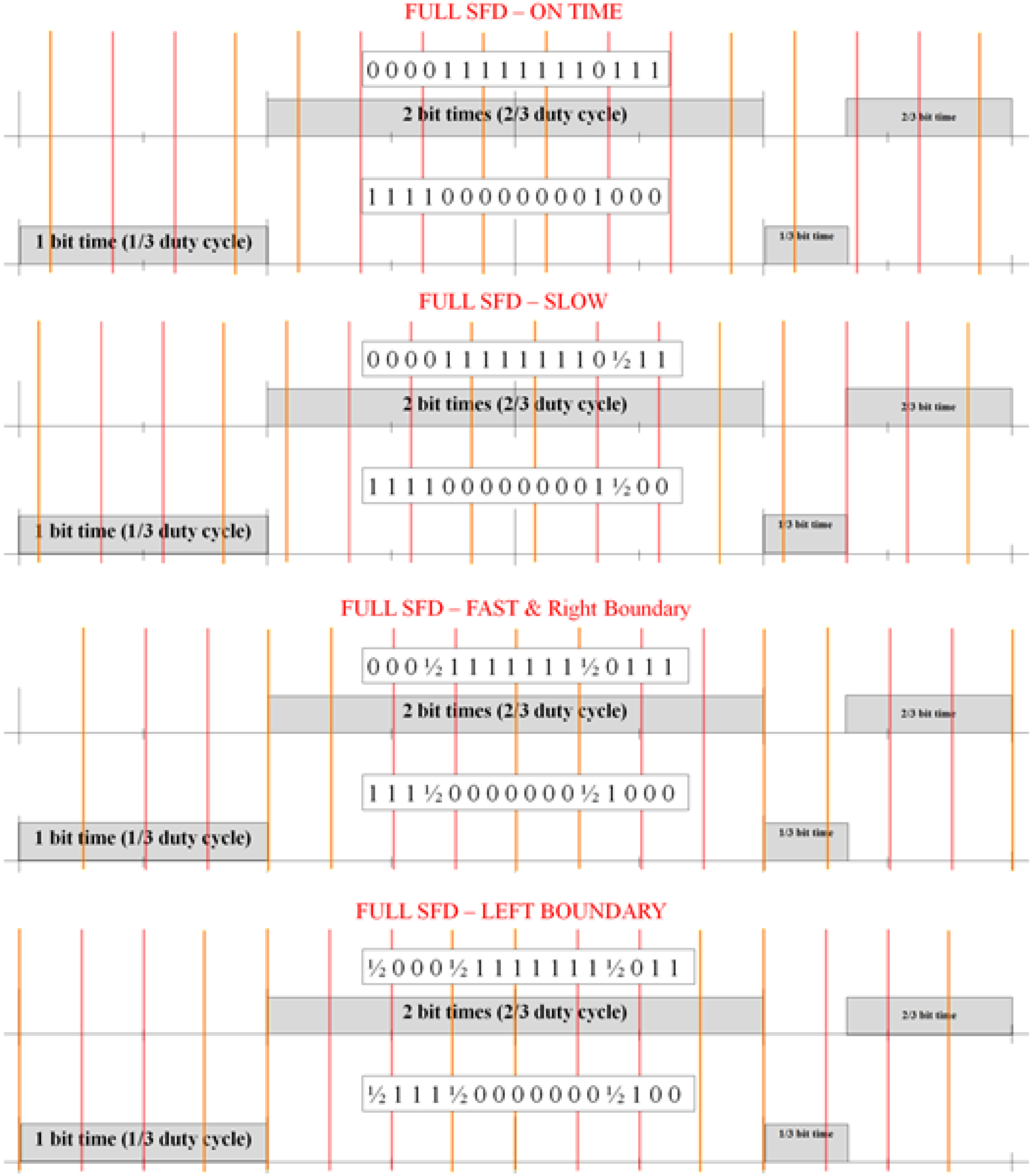
Characteristics:

1. It can be shown run length is never less than 7 (regardless of sample phase)
2. Minimizes SFD length at 4 bits
3. Allows us to set down-sample phase to avoid transition ISI
4. The TX oversample clock only needs to run at 3x bit rate

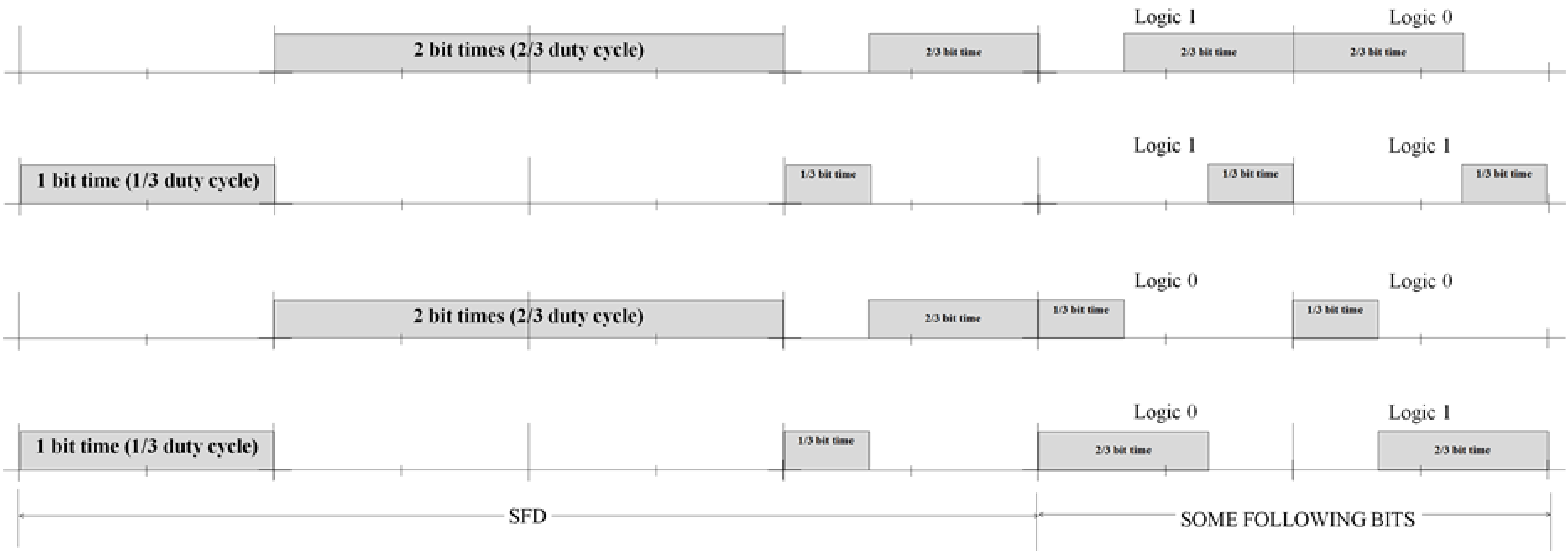
We need to be concerned about the transition edges because the finite camera exposure time causes inter-symbol interference which "smears" the transition edges. Sampling near the transition edges can cause unpredictable performance due to the quasi-stationary channel nature. We want to select sample phases that avoids sampling near the transition edge.



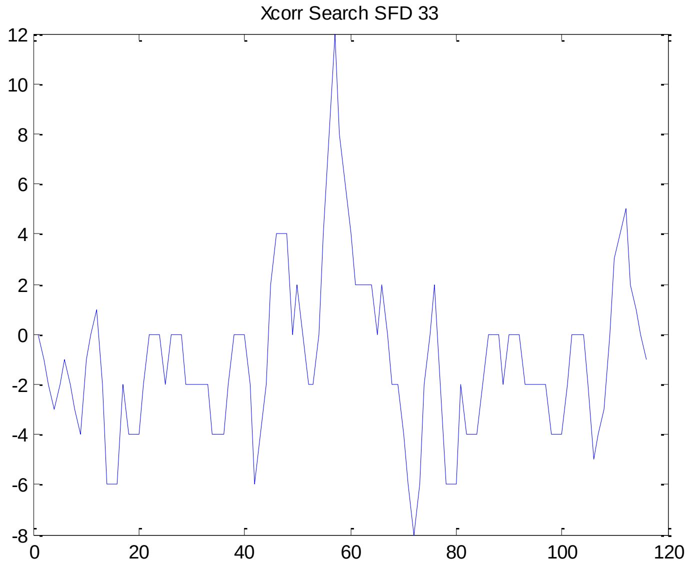
Examination of Various Sample Phases Reveals Run Length is Never Less Than Seven



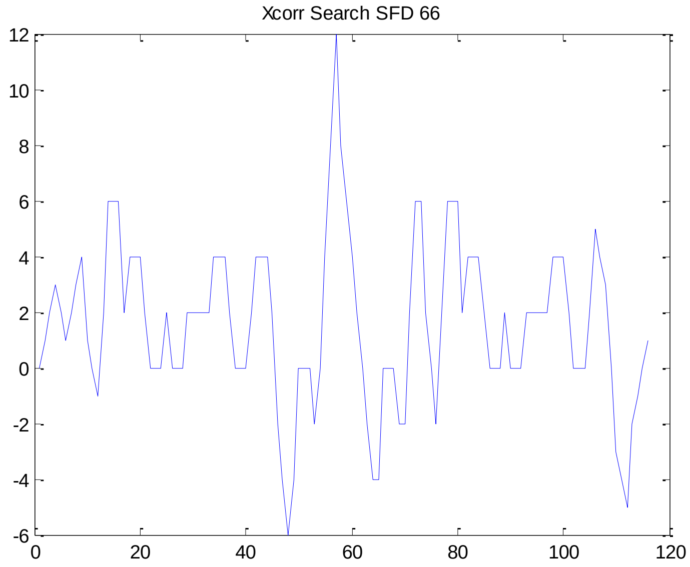
Some examples of the SFD followed by data bits.



Simulated cross-correlation looking for the SFD "buried" in random data.

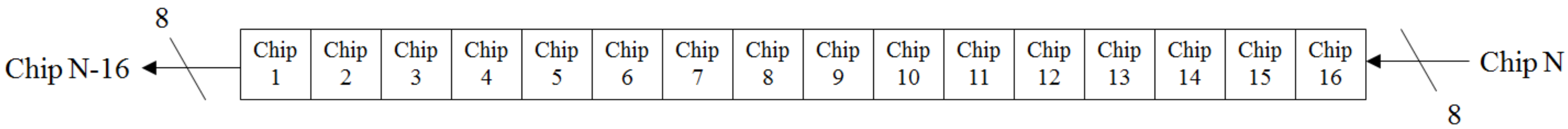


1/3 Duty Cycle



2/3 Duty Cycle

These results are based upon searching for the 3 bit illegal code.



A possible implementation of the SFD sliding correlator involves the use of a 16 chip shift register. The input is on the left and the output is on the right and the data samples represent image sensor grayscale readings taken from a pixel of interest via region-of-interesting sub-sampling. Taps are placed on the shift register so as to correlate for the 12 chip illegal code that forms the first 3 bits of the total SFD. Either soft decision or hard decision signal processing could be used.

Boolean Expression:

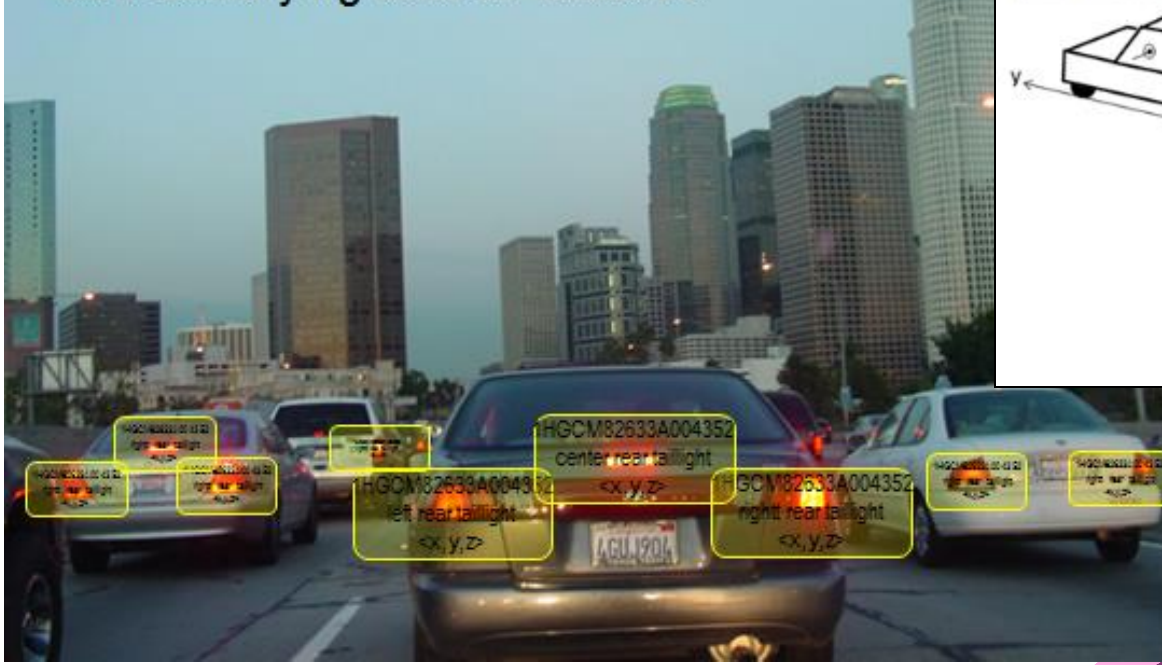
$$SFD_{DET} = (\overline{C1} \,\&\&\, \overline{C2} \,\&\&\, \overline{C3} \,\&\&\, \overline{C4} \,\&\&\, C5 \,\&\&\, C6 \,\&\&\, C7 \,\&\&\, C8 \,\&\&\, C9 \,\&\&\, C10 \,\&\&\, C11 \,\&\&\, C12) \,||\, (C1 \,\&\&\, C2 \,\&\&\, C3 \,\&\&\, C4 \,\&\&\, \overline{C5} \,\&\&\, \overline{C6} \,\&\&\, \overline{C7} \,\&\&\, \overline{C8} \,\&\&\, \overline{C9} \,\&\&\, \overline{C10} \,\&\&\, \overline{C11} \,\&\&\, \overline{C12})$$

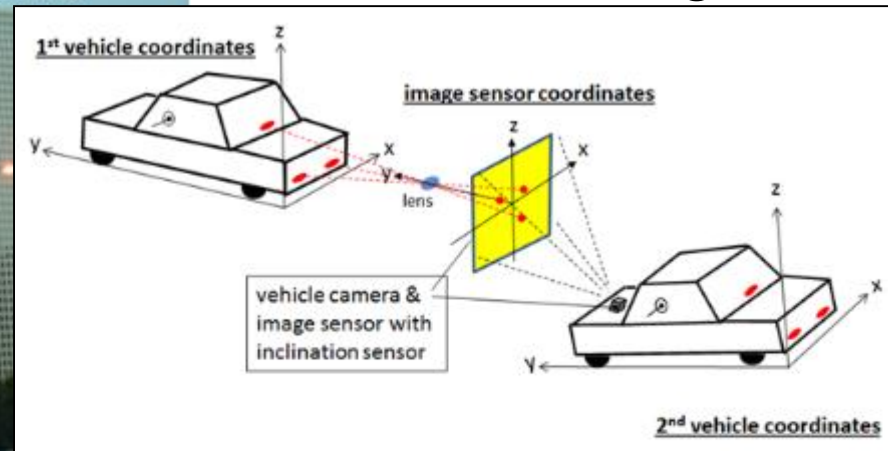
VPPM Bit Rates

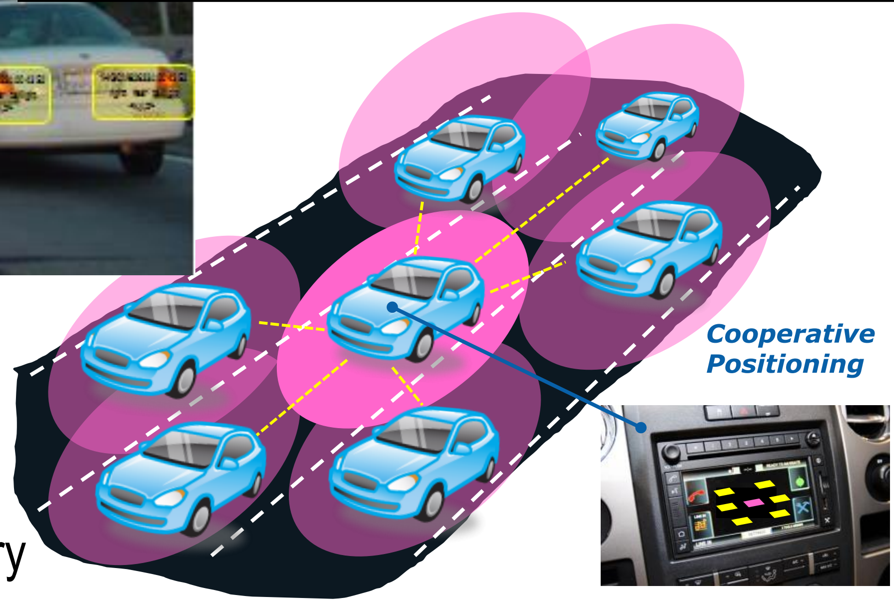
In this section we address the proposed VPPM bit rates needed to support various use cases. We start by considering the maximum bit rate related to the automotive comphotogrammetry use case.

Photogrammetry Positioning

Comphotogrammetry: the merger of CamCom and Photogrammetry such that light sources become self-identifying anchor features







- car has multiple lights (either visible or IR)
- each light broadcast an ID message
- ID message is processed by camera
- relative position determined by photogrammetry

Automobile Use Case - Estimated Bit Rate:

- VIN = 17 characters (23 capital letters and digits 0 to 9)
- Total number of ASCII bits to represent VIN = 7*17=119 bits

Coordinates max values

- Length: 10,000 cm ... 5 digits at 4 bits per 20 bits
- Width: 300 cm … 3 digits at 4 bits per digit = 12 bits
- Height: 1400 cm … 4 digits at 4 bits per digit = 16 bits
- Total number of bits: 167

Assume 100% overhead and FEC: 334 bits … round up to 400 bits.

- 400 bits sent in 100 mS yields bit rate of 4 kbps.
- 4x over sample yields frame rate = 16 kfps.

Compare to rolling shutter sample rate of 8 million pixel camera …

- OmniVision OV8858: 3264 x 2448 @ 30 fps (2448 rows)
- Row sampling = (1/30)/2448 = 13.6 uS or 73,440 rows per second

Achieving 16 kfps is achievable assuming one knows what region-of-interest row to sample.

Automobile Aug. Reality Case - Estimated Bit Rate:

- IPv6 address: 128 bits.

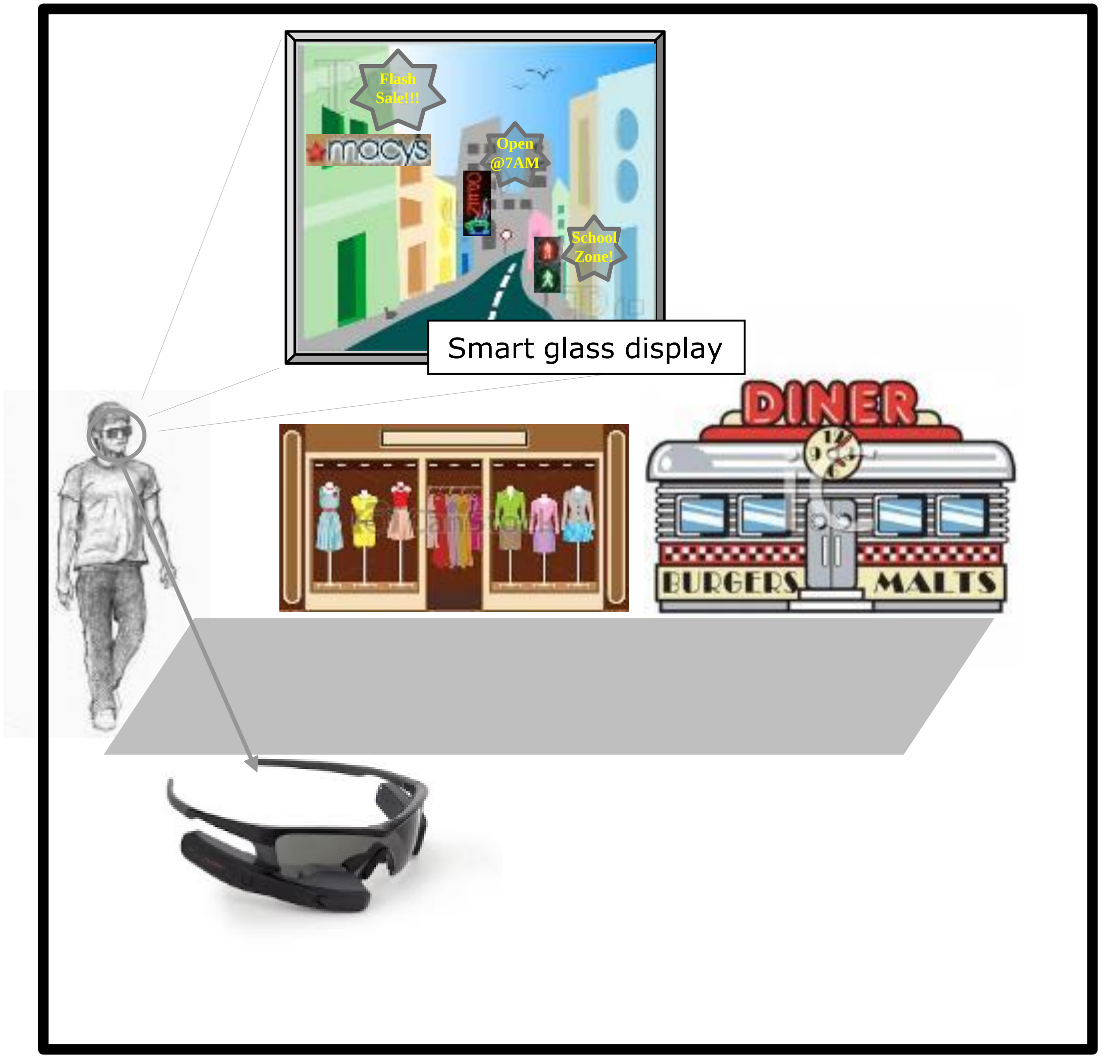
Assume 100% overhead and FEC: 128 bits … round up to 300 bits.

- Max. Vehicle speed: 80 mph = 37.76 m/s.
- Max. Range: 100 m.
- Min. Range: 3 m.

- Time to travel (Max. Range – Min. Range) : 97/37.76 ≈ 2.7 s.
- Assume modulated lights blockage 80 % of the observation time.
- Assume 0.4 s are used for modulated lights detection.
- If T is the time we have to extract data from the modulated lights, then: T = 2.7*0.2-0.4 ≈ 0.1 s.

- 300 bits sent in T s yields bit rate of 3 kbps.
- 4x over sample yields frame rate = 12 kbps.

Pedestrian Augmented Reality



Wearable Aug. Reality Case - Estimated Bit Rate:

- IPv6 address: 128 bits.

Assume 100% overhead and FEC: 128 bits … round up to 300 bits.

- Gaze time = 1 s.
- Assume modulated light blockage 20 % of the observation time.
- Assume 0.4 s are used for modulated light detection.
- If T is the time we have to extract data from the modulated lights, then: T = 1*0.8-0.4 = 0.4 s.

- 300 bits sent in T s yields bit rate of 750 bps.
- 4x over sample yields frame rate = 3 kbps.

Proposed VPPM Use Case Bit Rates and Frame Rates		
Use Case	**Bit Rate**	**ROI Min Frame Rate**
Auto Positioning	4 kbps	16 kbps
Auto Aug. Reality	3 kbps	12 kbps
Wearable Aug. Reality	750 bps	3 kbps
TBD	TBD	TBD
TBD	TBD	TBD
TBD	TBD	TBD

VPPM FEC

1. Reed-Solomon
2. (n,k) is TBD
3. Selected for robustness to both burst and random errors

VPPM with Twinkle Performance

The parameters are as follows:

1. Camera frame rate: 2040 chips/sec
2. Exposure time: 5e-5 (10% of chip duration)
3. SNR = 100 dB
4. Payload length: Variable (Figure 1), 200 bits (Figure 2).
5. Number of bits per AM envelope cycle: 20 bits/cycle

Figure 1 shows the maximum error free packet length vs the frequency offset between the camera frame rate and the data chip rate (no offset tracking – which could allow longer packets).

Figure 2

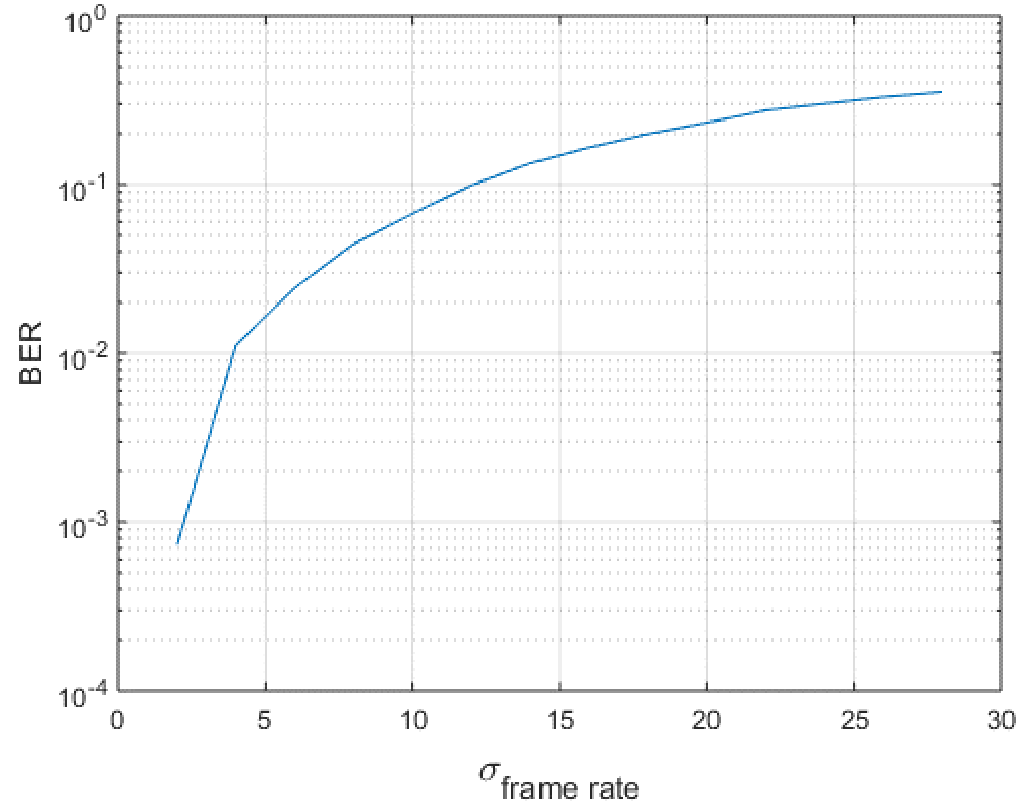


Figure 2 shows the impact of camera frame rate jitter on the overall BER.

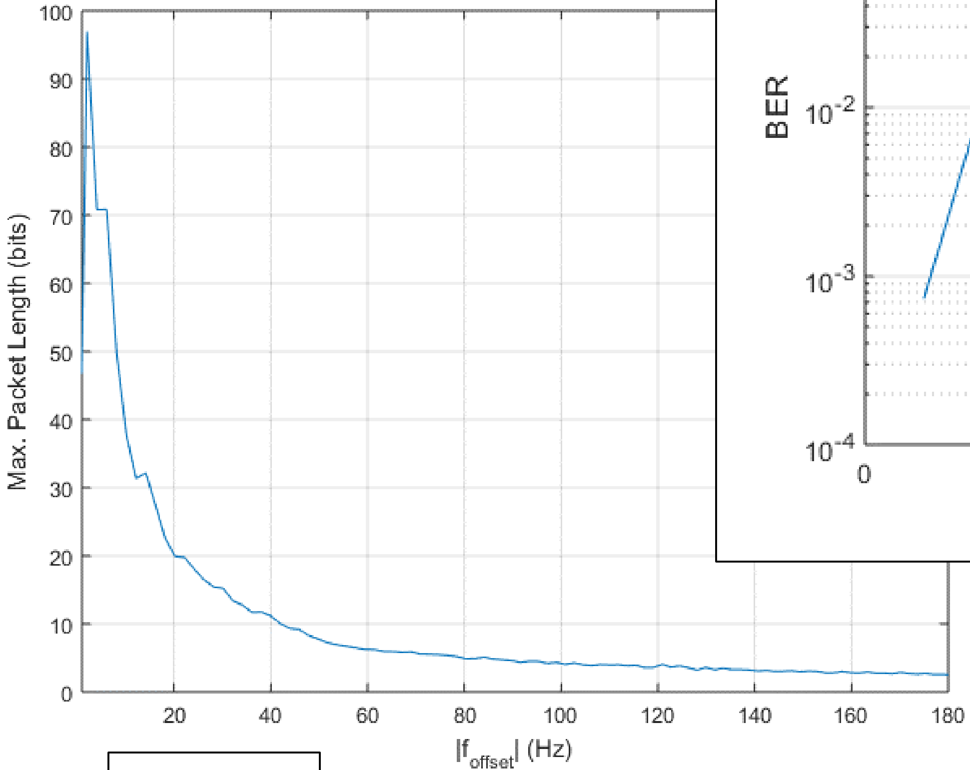


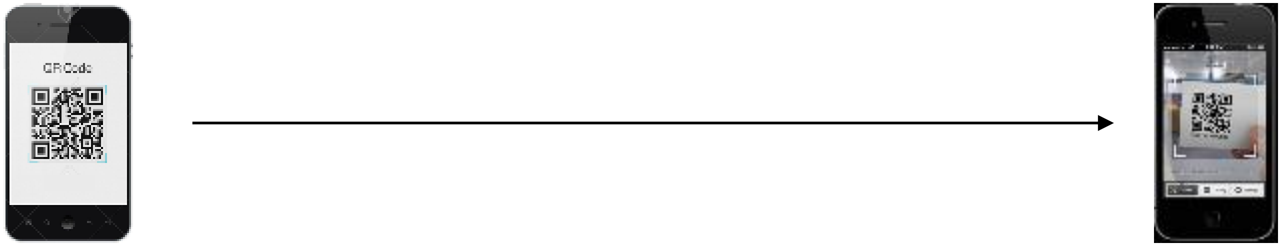
Figure 1

In all the figures, no FEC has been added.

OCC High Rate PHY

Sequential QR Code Transmission

Transmitter Receiver



- Transmissions are unidirectional
- Bi-direction communications is “synthesized” at a higher layer by concatenating two unidirectional transmissions
  - QR codes are encoded/decoded at higher layers (not PHY/MAC)
- QR codes are not processed real-time, rather a short video is short of the sequence (up to 48 frames) and then post-processed non-real-time.

Proposed High Rate OCC PHY Two Dimensional Code

We are proposing using QR codes specified by ISO/IEC 18004:2015 Information technology -- Automatic identification and data capture techniques -- QR Code bar code symbology specification.

INTERNATIONAL STANDARD

ISO/IEC 18004

Third edition
2015-02-01

Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification

Technologies de l'information — Technologie d'identification automatique et de capture des données — Spécification de la symbologie de code à barres Code QR

Reference number
ISO/IEC 18004:2015(E)

© ISO/IEC 2015

Summary of ISO/IEC 18004:2015 QR Code

- Encodable character set
  - Numeric Data
  - Alphanumeric Data
  - Byte Data
  - Kanji Characters

- Representation of data
  - Dark module is nominally a binary one; a light modulate is nominally a binary zero
  - Supports reflective reversal
  - Supports mirror imaging
  - Rotation independence

Summary of ISO/IEC 18004:2015 QR Code (cont.)

- Symbol size
  - 21 x 21 modules to 177 x 177 modules (version 1 to 40)
- Maximum data characters per symbol (version 40-L)
  - Numeric data: 7,089 characters
  - Alphanumeric data: 4,296 characters
  - Byte data: 2,953 characters
  - Kanji data: 1,817 characters
- Selectable Reed-Solomon error correction
  - Level “L” 7% errors
  - Level “M” 15% errors
  - Level “Q” 25% errors
  - Level “H” 30% errors

Summary of ISO/IEC 18004:2015 QR Code (cont.)

- Additional features
  - Structured append: allows 16 QR codes to be logically connected to allow large file transfers
  - Extended Channel Interpretations: allows use of other character sets like Arabic, Cyrillic, Greek, etc. and also supports compacted data transmission

- QR code “housekeeping” symbolic data
  - Version information
  - Format information
  - Alignment patterns
  - Timing patterns

The details of the OCC High Rate PHY mode are under management review – a decision on proceeding will be made prior to the March 2016 IEEE802.15.7 meeting.

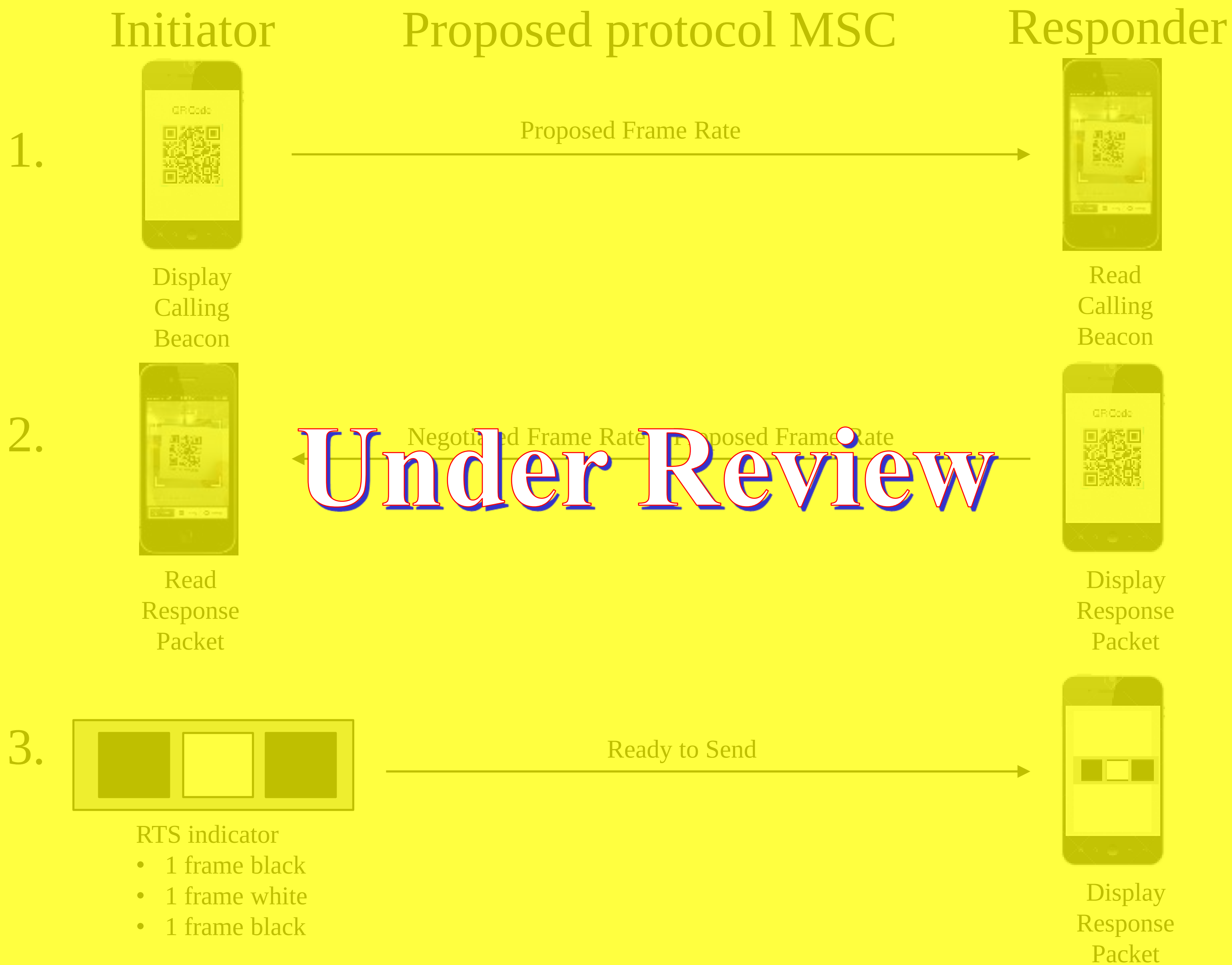
At the March 2016 meeting either the details for the OCC High Rate PHY will be presented or the OCC High Rate PHY mode will be withdrawn.

The basic problem is resource limitations have prevented us from making sufficient technical progress to present a creditable OCC High Rate PHY proposal at this time.

We are curious as to the level of IEEE802.15.7r1 interest in this type of OCC operational mode (i.e. using spatial codes).

Intel would be interested in collaboration with others in regards to the 2-D sequential code OCC PHY mode.

Proposed protocol MSC



Initiator Proposed protocol MSC (cont.) Responder

4.

Send Data →

Send each QR code for 3 video frames

Receiver discards invalid and duplicate video frames and keeps one frame per QR code for processing.

5.

Send Data →

Up to 16 QR codes may be send using the structured append feature.

Receiver records all video frames, keeping one frame of each triplet and recovering up to 16 QR codes for post processing.

Maximum bit rate at 30 frames per second:

- 16 QR codes in 16 x 3 = 48 frames
- 2953 x 8 = 23624 bits per code
- Total number of bits = 377984 bits
- Bit Rate … 16*23624/(48/30) = 236,240 bps

All data frames and required headers are contained within the QR code protocol itself.

One QR every two frames doesn’t work!

Camera and Display Running at same frame rate

Hold QR display for one refresh time interval	Write the new QR display for one refresh time interval	Hold QR display for one refresh time interval	Write the new QR display for one refresh time interval

Camera Reads QR code	Camera Reads QR code	Camera Reads QR code

Camera never gets to read the complete QR code

One QR every three frames does work!

Camera and Display Running at same frame rate

Hold QR display for one refresh time interval	Write the new QR display for one refresh time interval	Hold QR display for one refresh time interval	Hold QR display for one refresh time interval	Write the new QR display for one refresh time interval

Camera Reads QR code	Camera Reads QR code	Camera Reads QR code	Camera Reads QR code

By displaying the QR code for two intervals, camera gets a chance to see a correct QR code

Why display and read each QR code three times?

Reason is to accommodate QR code refresh on slowly updating screens.

T=0 First QR code | T=1/FPS First QR code | T=2/FPS Screen rewrite | T=3/FPS Second QR code | T=4/FPS Second QR code

OCC Superframe Format and Access Mechanism

OCC Operates Broadcast Mode Only

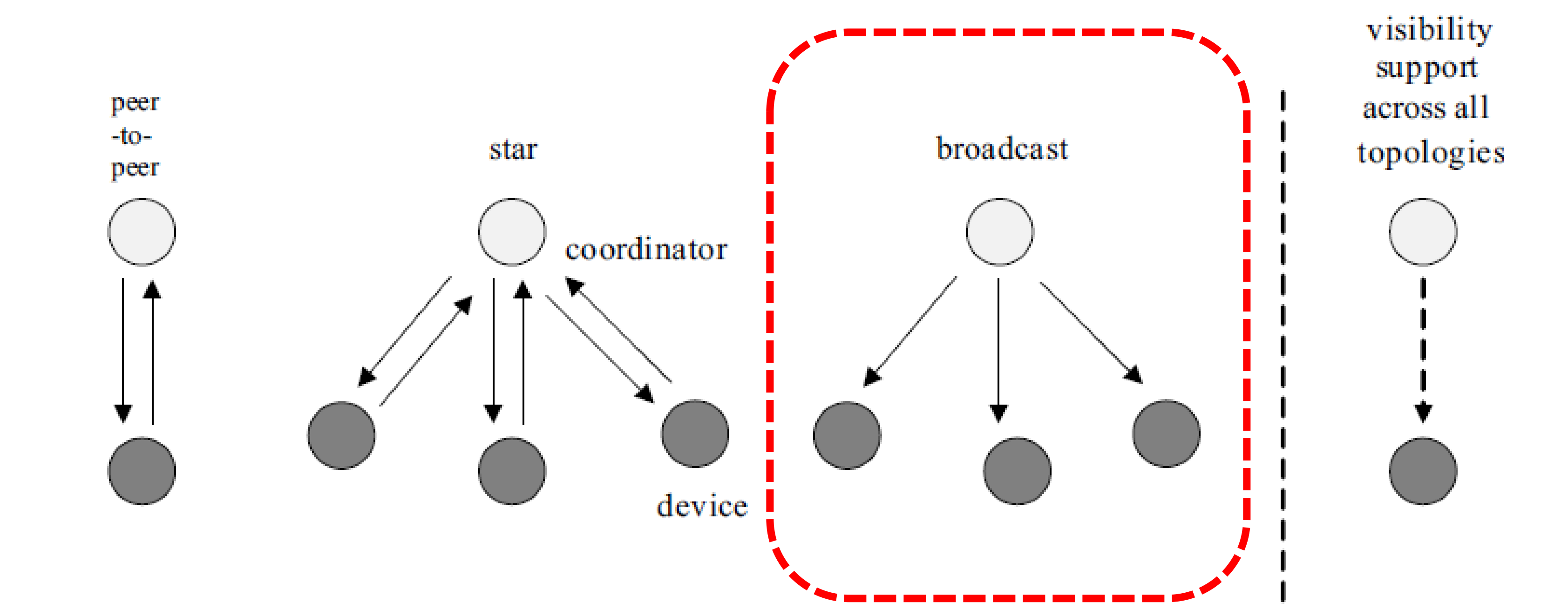


Figure 1—Supported MAC topologies

Intel views OCC as primarily operating in the broadcast mode; that is, the superframe would be setup to support broadcast mode operating in a NO BEACON mode.

4.2 Network topologies

… IEEE 802.15.7 devices are also allowed to operate in a broadcast only topology without being part of a network, i.e., without being associated to any device or having any devices associated to them.

4.2.3 Broadcast topology

The basic structure of a broadcast topology is illustrated in Figure 1. The device in a broadcast mode can transmit a signal to other devices without forming a network. The communication is uni-directional and the destination address is not required.

5.1.2.2 VPAN initiation

The broadcast mode does not have any requirements for starting a VPAN. Starting a VPAN is only applicable to bi-directional communication modes and not for broadcasting.

5.1.2.4 Device discovery

Device discovery requires bi-directional communication and is not applicable for broadcasting.

5.1.2.5 Guard and aggregation color channels

The bandplan provides support for seven logical channels in the MAC. However, in order to support association without knowledge of receiver capabilities and to support unidirectional broadcasting, the VLC receiver shall support reception on the entire visible light spectrum with any type of optical light source.

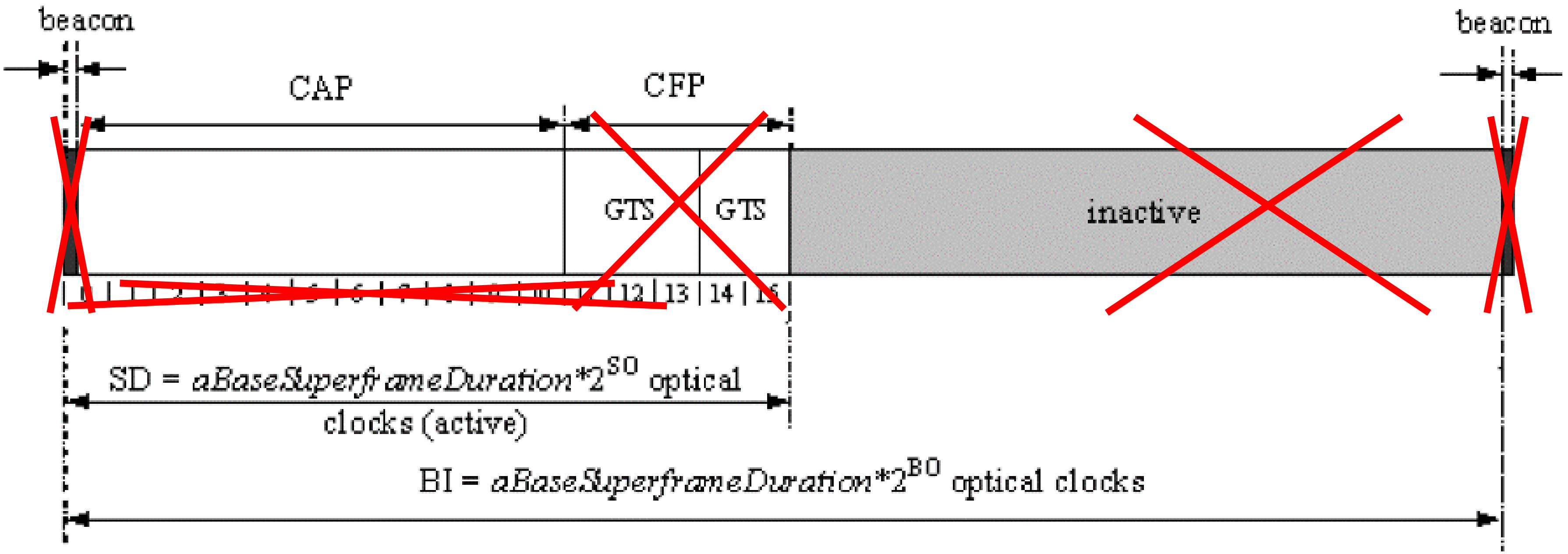
We are proposing that OCC PHYs are not confined to any of the 802.15.7-2011 bandplans as described in clause 8.3.1, “Wavelength band plan”; that is, the OCC emitter may operate on wavelengths from 300 nm (near UV) to 1000 nm (near IR). The OCC receiver needs to be responsive to all wavelengths from the longest wavelength to the shortest wavelength.

5.1.1.3 Random access algorithm

… if periodic beacons are not being used in the VPAN or if a beacon could not be located in a beacon-enabled VPAN, the MAC sublayer shall transmit using the unslotted version of the random access algorithm. ~~In both cases, the algorithm is implemented using units of time called backoff periods, where one backoff period shall be equal to aUnitBackoffPeriod optical clocks.~~

The OCC broadcast mode does not use a backoff period prior to transmitting due to the presence of optics that allows spatial coexistence between multiple users.

The OCC superframe is beaconless and uses un-slotted contention access. The broadcast transmitter does not do listen before talk rather relying on spatial re-use, provided by the optics, to accommodate multiple uses and insurance coexistence with other OCC like systems.



The definition of optical clocks for OCC is TBD.

Clause 5.1.1.1.3, visibility support during channel access, is not applicable to the OCC broadcast mode since data is continuously sent from a light source that is continuously lite.

Clause 5.1.1.2, interframe spacing (IFS), is not applicable for the OCC broadcast mode since the data transmissions are repetitive and continuous.

OCC MAC/PHY Frame Formats

MAC Frame Format

The native MPDU has too much overhead for OCC and most of the fields are not needed for a short, repetitive MSDU.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination VPAN Identifier	Destination Address	Source VPAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
		Addressing fields						
MHR							MSDU	MFR

Figure 44—General MAC frame format

The Field Check Sequence (FCS) can be optional.

The MAC text will need to be modified to reflect these changes.

PHY Frame Format

Modification needed for clause 8.2 “Operating modes” …

… there is no mandatory mode for the OCC PHY. All PHY modes are specified using a new PHY PIB attribute, managed by the DME, called *phyOccApplicationSpecficMode*. The details are presented in the next section.

For OCC the current PPDU structure has too much overhead – in many cases the overhead exceeds the payload length.

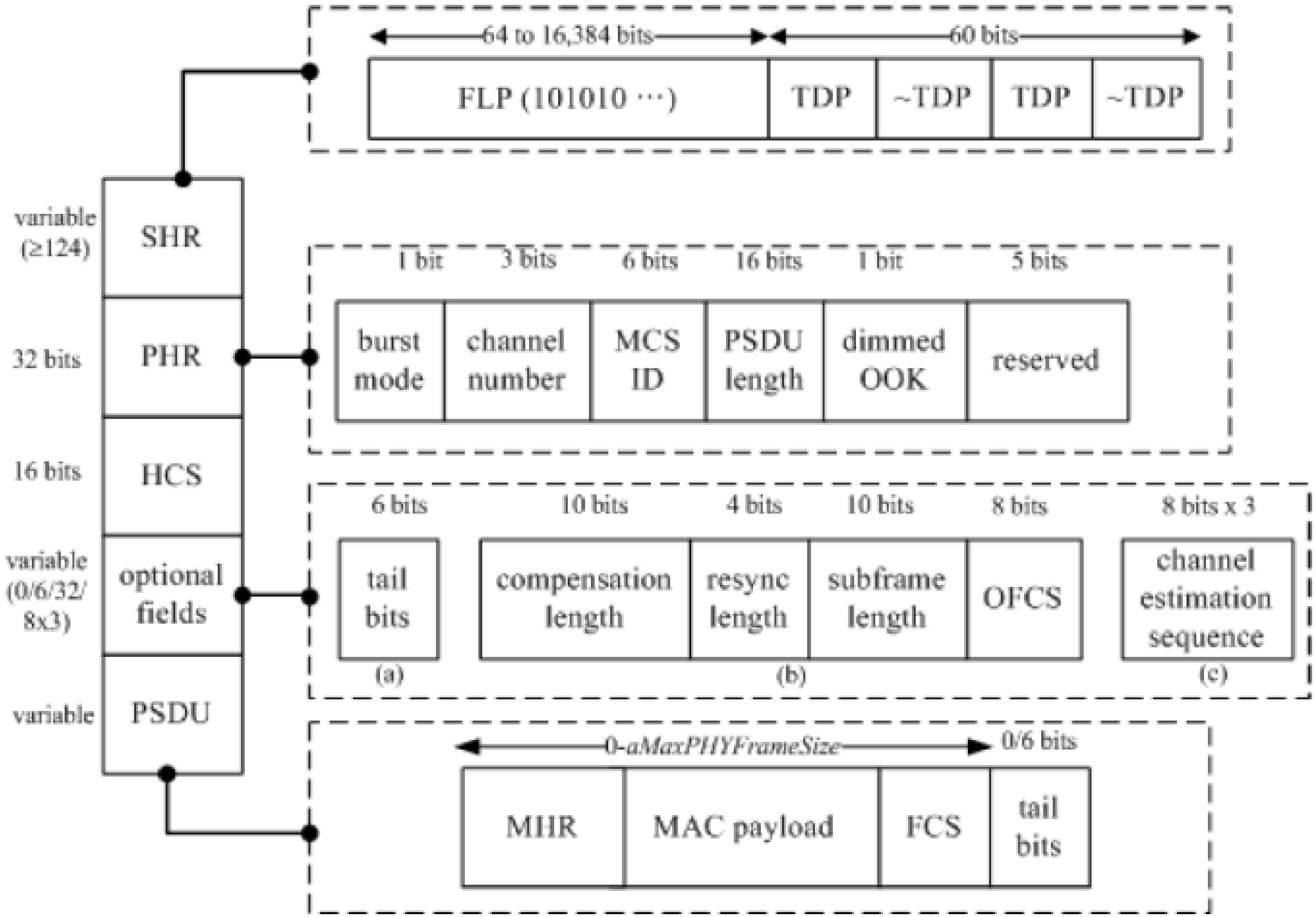


Figure 124—PPDU structure

Preamble (see 8.6.1)	PHY header (see 8.6.2)	HCS (see 8.6.3)	Optional fields (see 8.6.4)	PSDU (see 8.6.5)
SHR	PHR			PHY payload

Figure 118—Format of the PPDU

For OCC it is proposed that:

- the preamble is replaced with a Start Frame Delimiter
- optional OCC PHY headers can be defined
- HCS is optional (but probably not needed)

The PHY text will need to be modified to reflect these changes.

MAC/PHY Management Services

Both MAC and PHY layers conceptually include management entities, called the MAC sublayer management entity and PHY layer management entity (MLME and PLME, respectively). These entities provide the layer management service interfaces for the layer management functions.

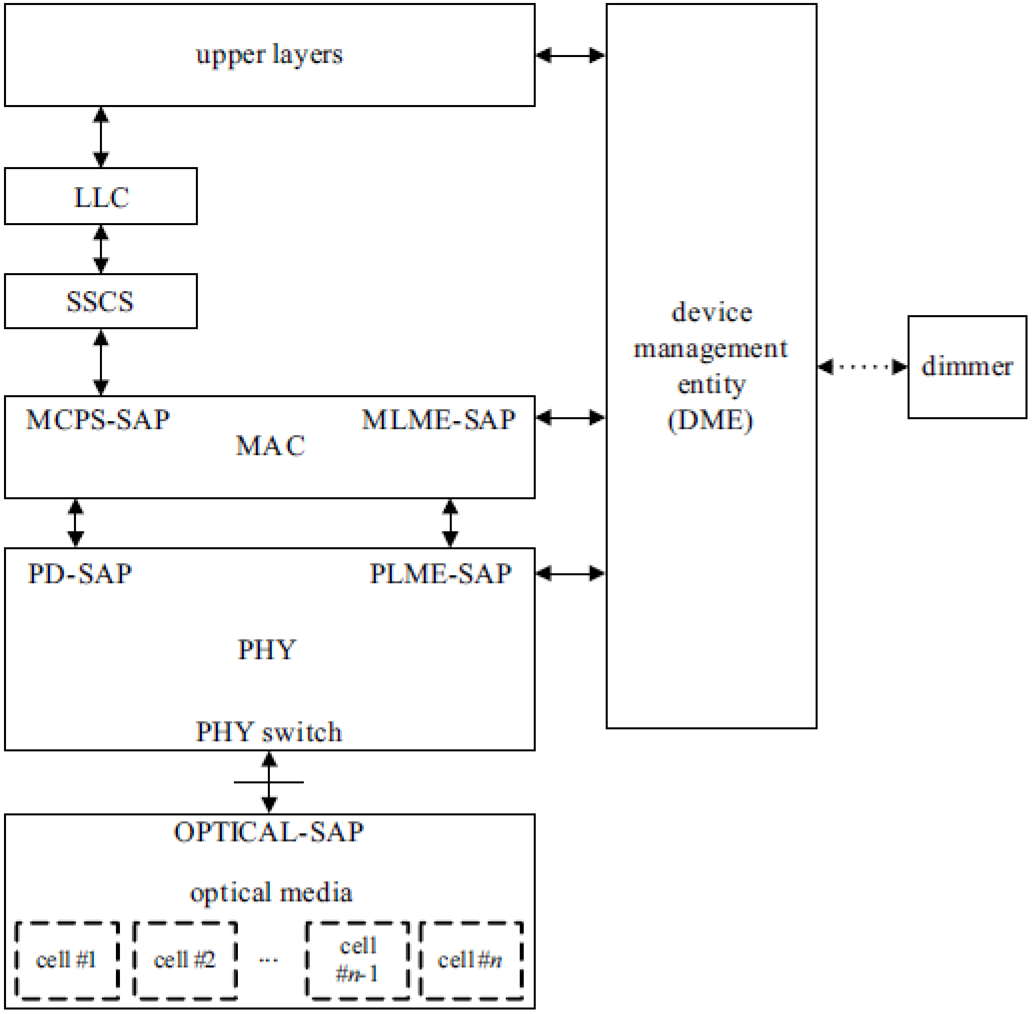


Figure 3—VPAN device architecture

Add 2 MAC PIB items …

ID Ambiguity Resolution: used to resolve the location ambiguity of a device with a very short ID code. A particular short ID code can be duplicated numerous times around the world; that is, it is not unique. But by appending the ID code to a unique MAC address then the combination becomes unique and globally identifies a particular device.

Application Specific Packet Length: used to specify the length of a data packet for a given geolocation such as the APP associated with a particular store, The information is provided by an out-of-band channel (i.e. WiFi) and is provided by the particular store prior usage (i.e. by downloading the stores OCC APP).

Add 1 PHY PIB item …

OCC Application Specific PHY Mode: used to specify the PHY mode for a given geolocation such as the APP associated with a particular store, The information is provided by an out-of-band channel (i.e. WiFi) and is provided by the particular store prior usage (i.e. by downloading the stores OCC APP).

6.4.2 MAC PIB attributes

The MAC PIB comprises the attributes required to manage the MAC sublayer of a device. The attributes contained in the MAC PIB are presented in Table 60 and Table 66. Attributes marked with a dagger (†) are read-only attributes (i.e., attribute can only be set by the MAC sublayer), which can be read by the next higher layer using the MLME-GET.request primitive. All other attributes can be read or written by the next higher layer using the MLME-GET.request or MLME-SET.request primitives, respectively. Higher layers may impose additional constraints on read/write operations, without making devices non-compliant. Attributes marked with a diamond (♦) are optional for a device (i.e., not operating as a coordinator).

Add the following items to the MAC PIB …

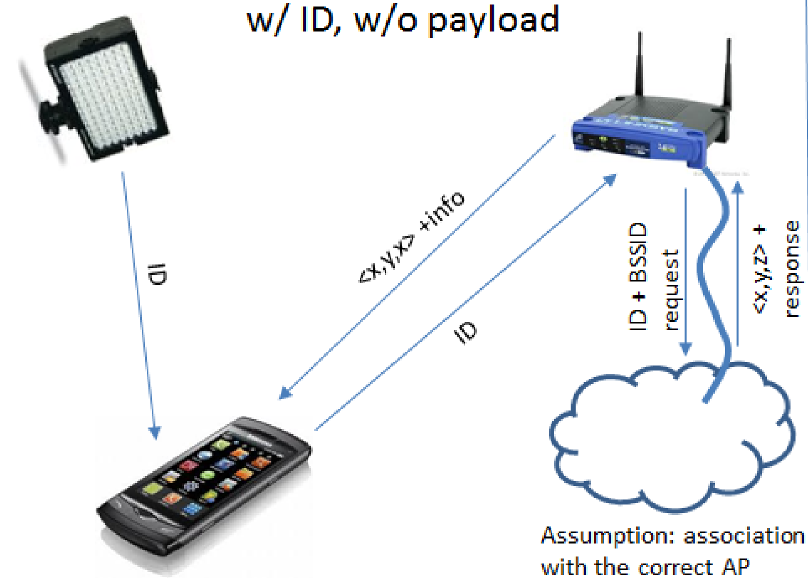
MAC PIB Table 60 Addition

Attribute	Identifier	Type	Range	Description	Default
mac_LED-ID_Ambiguity_Resolution	0x71	Unsigned	0-255	This attribute resolves the ambiguity of a short LED ID by appending it to a SSID as shown in the table below.	0

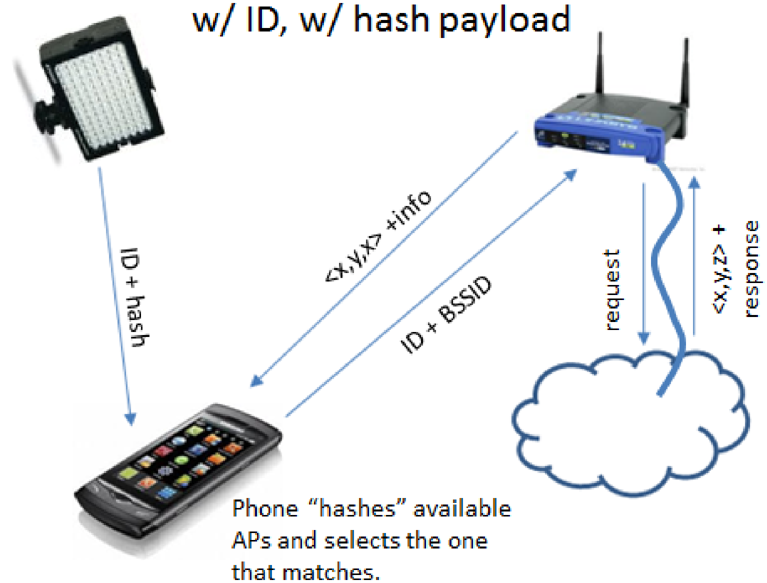
Table xx – Ambiguity Resolution Method

PIB Attribute Value	Method Name	Method
0	With ID without payload	Preappend ID to SSID
1	With ID and SSID hash	Preappend ID to hash resolved SSID
2	w/ or w/o ID and IP address	Preappend ID to provided IP address

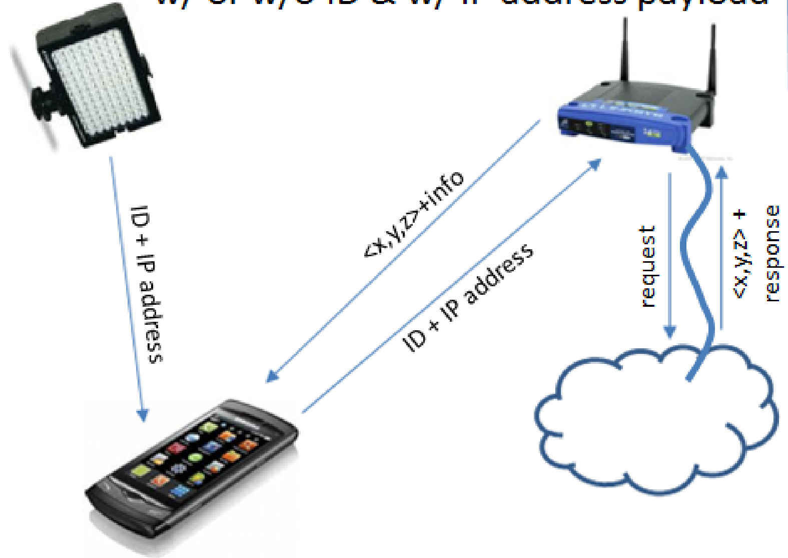
w/ ID, w/o payload



w/ ID, w/ hash payload



w/ or w/o ID & w/ IP address payload



We can resolve short ID location ambiguity by either:

i) appending the ID to the access point MAC address (BSSID) … the combination provides a globally unique identifier

ii) using an ID and BSSID hash to make sure we access through a specific AP … this specific AP provides specific info to the user

iii) w/ or w/o an ID and sending an IP address payload … great when we want additional info provided by a specific server

9.5.2 PHY PIB attributes

The PHY PIB comprises the attributes required to manage the PHY of a device. Each of these attributes can be read or written using the PLME-GET.request and PLME-SET.request primitives, respectively. The attributes contained in the PHY PIB are presented in Table 100.

PHY PIB Table 100 Addition				
Attribute	Identifier	Type	Range	Description
phyOccApplicationSpecficMode	0x06	Integer	0-N	This attribute specifics the application specific PHY mode as shown in the table below.

Table xx – OCC Application Specific PHY Mode		
PIB Attribute Value	Mode Name	Mode
0	Rolling Shutter Sampling #1	First Rolling Shutter Mode
1	Rolling Shutter Sampling #2	Second Rolling Shutter Mode
:	:	:
i	UFSOOK #1	First UFSOOK Mode
i+1	UFSOOK #2	Second UFSOOK Mode
:	:	:
j	LED ID Mode #1	First LED ID Mode
j+1	LED ID Mode #2	Second LED ID MOde
:	:	:
k	RS-FSK Mode	RS-FSK MOde
:	:	:
N	2-D Sequential Spatial Code	2-D Sequential Spatial Code

Backup Slides

Pixel Eb/No Computations

Unlike RF, where the sole purpose of the TX is the transmission of bits, in VLC the primary purpose of the light source TX is for illumination with data TX being secondary. One could argue that VLC system performance is solely based upon what the image sensor detector "sees" when the shutter is open and not on the "wasted" TX power that is sent when the camera shutter is closed. This is somewhat a philosophical issue. CamCom does not use matched filtering and the signal integration time is only a fraction of the bit interval. For VPPM the integration time is typically less than 25% of a bit interval. For UFSOOK the integration time is typically less than 1% of a bit interval. Therefore, assuming the light is either ON or OFF for the duration of the camera exposure time, the actual duty cycle of the light itself is irrelevant. This is particularly true given the nature of the proposed acquisition algorithms.

Pixel Eb/No computation: Noise model

Pixel noise in CCD/CMOS cameras can be approximately modeled as follows

$$n \sim N(0, \sigma(s)^2).$$

where s is the pixel value, and σ^2(s) = $s\alpha a + \beta$, with α, β obtained experimentally.

Consider the following parameters:

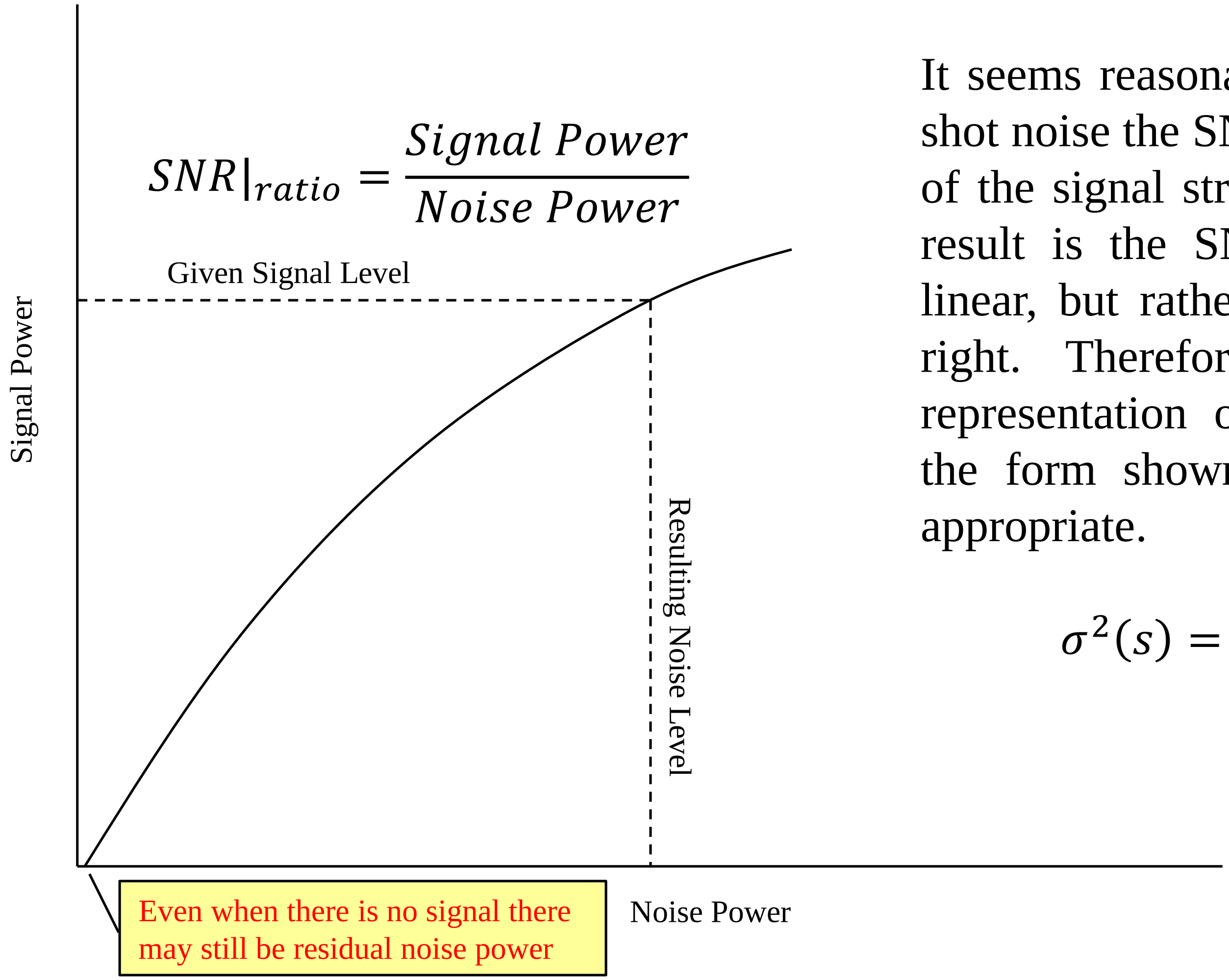
- s = Pixel value.
- a = Mark and space amplitude.
- Δ = Camera exposure duration as ratio of bit period $\left(\Delta = \frac{T_{exposure}}{T_{bit}}\right)$.
- n = Noise value $\sim N(0, \sigma(s)^2)$.
- σ^2(s) = $s\alpha + \beta$.
- α, β = Noise model fit parameters.

The pixel Eb/No, assuming one bit per symbol, is computed as follows:

$$\text{Pixel } \frac{\varepsilon_b}{N_0} = \frac{E[s^2]}{E[n^2]} \approx \frac{a^2 \cdot \Delta}{\alpha \cdot a \cdot \Delta + \beta}$$

Discussion on Eb/No equation

$$SNR|_{ratio} = \frac{Signal\ Power}{Noise\ Power}$$

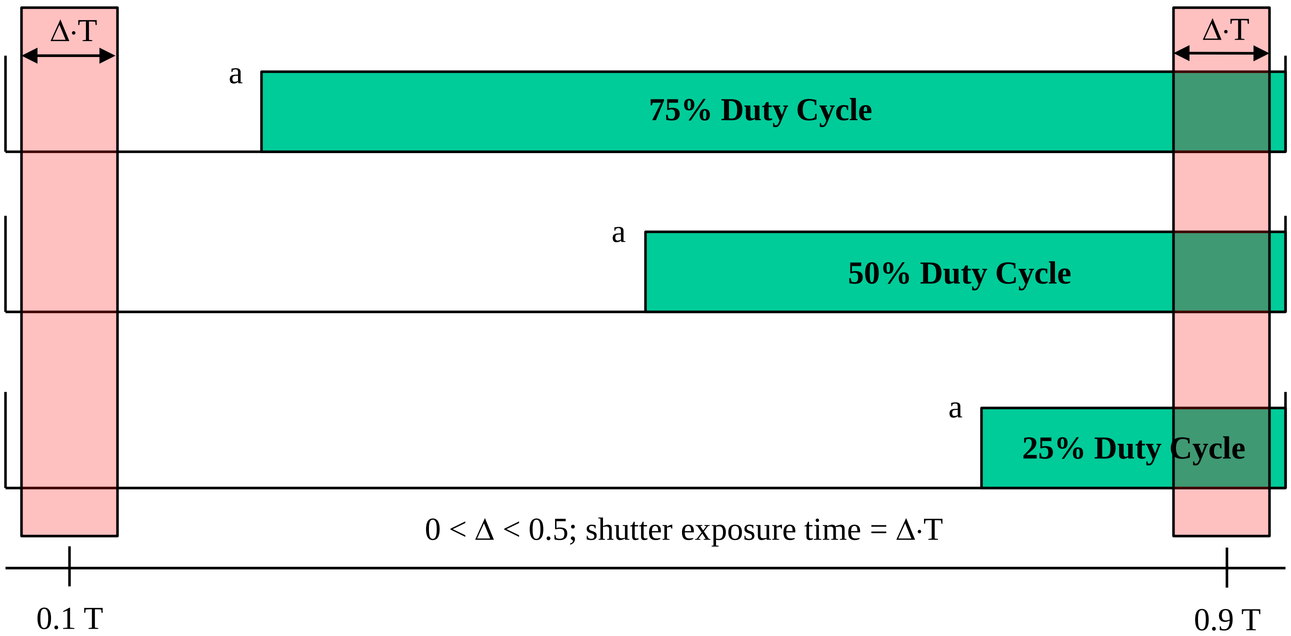


It seems reasonable that due to shot noise the SNR is a function of the signal strength. The net result is the SNR plot is not linear, but rather curves to the right. Therefore, a parametric representation of the noise of the form shown below seems appropriate.

$$\sigma^2(s) = s \cdot \alpha + \beta$$

Illustrative Example

VPPM Symbol sampled at 0.1 bit interval and 0.9 bit interval with fractional exposure time.



In the above example, regardless of which duty cycle is being used, when the camera shutter is "open" and the pixels are being exposed, the light is either OFF or FULL ON. The camera is not aware that the duty cycle is being changed; hence, the BER performance for all 3 duty cycles is identical. The following equation applies provided the exposure time does not exceed the time the light is actually ON.

$$\text{Pixel } \frac{\varepsilon_b}{N_0} = \frac{E[s^2]}{E[n^2]} \approx \frac{a^2 \cdot \Delta}{\alpha \cdot a \cdot \Delta + \beta} \qquad P_e \approx \frac{1}{2} e^{\left(-\frac{\varepsilon_b}{2 \cdot N_0}\right)}$$

Given a desired Eb/No, we need to determine the prerequisite amplitude "*a*".

$$\frac{\varepsilon_b}{N_0} \approx \frac{a^2 \cdot \Delta}{\alpha \cdot a \cdot \Delta + \beta}$$

Assumptions [1]

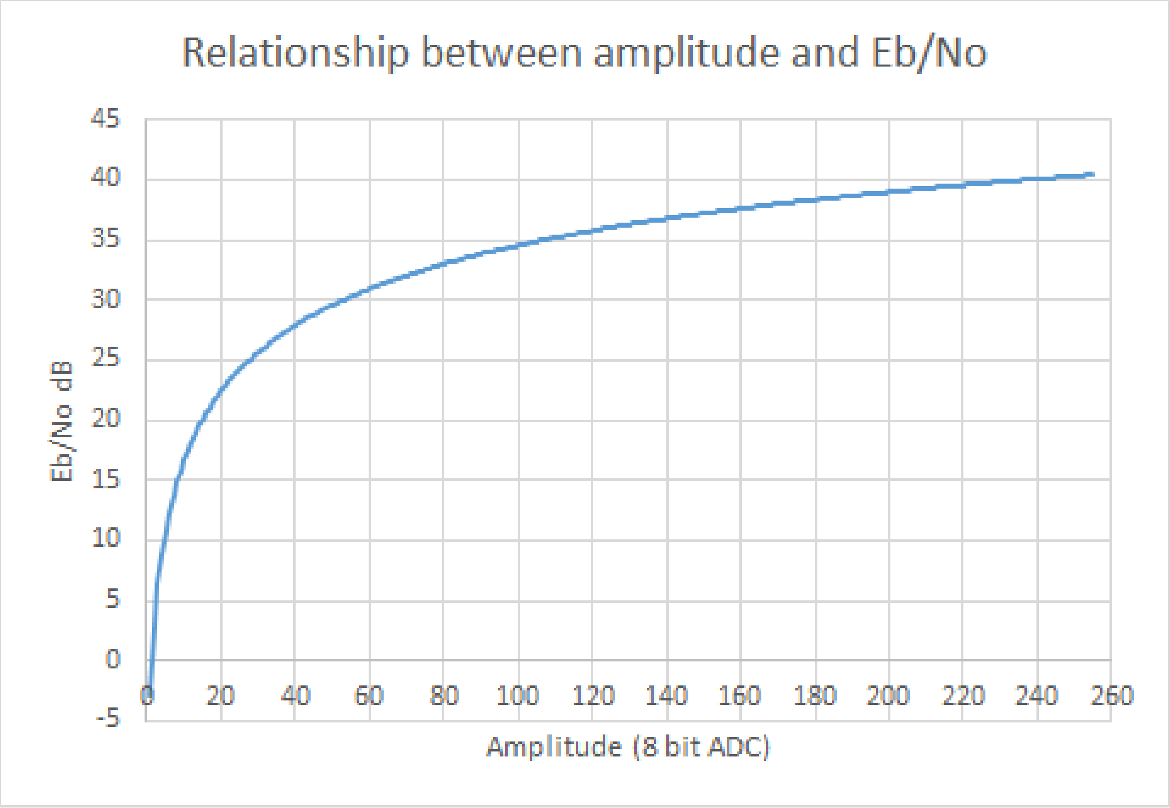
$\alpha = 0.01529$; $\beta = 0.1973$

A closed form solution appears to be ugly but we can numerically solve this as shown below.

Assume $\Delta = 0.1$

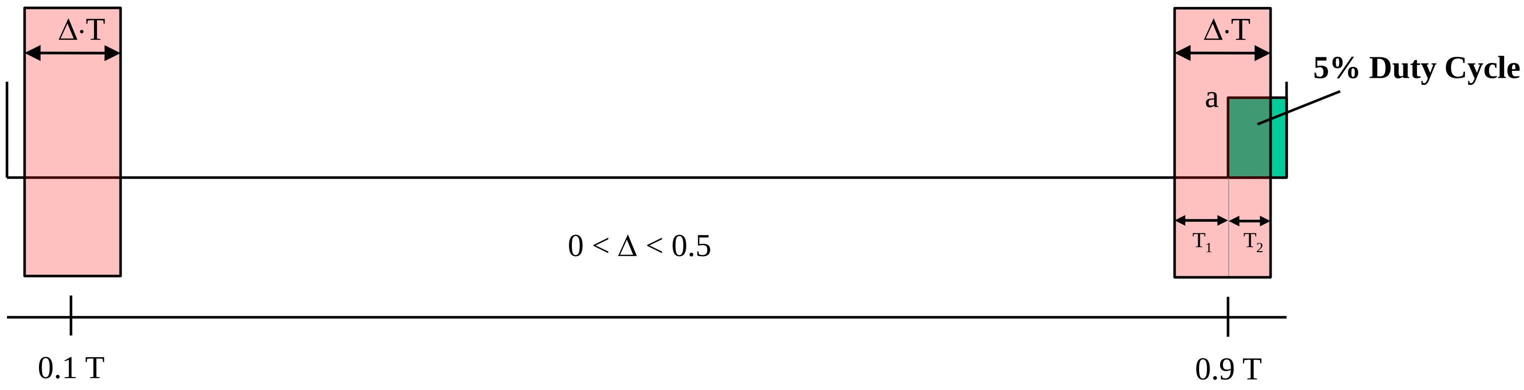
Eb/No	Amplitude
10 dB	5
20 dB	15
30 dB	53
40 dB	236

[1] J. Perez-Ramirez, E. Curry, D.K. Borah, "*Experimental Multiuser Mobile Optical Communication Using Compressive Sensing*", Globecom 2014, OWC Workshop, Dec. 2014



Naturally, when the camera exposure time catches a light transition edge, then the pixel Eb/No *during the exposure time* will decrease in a manner similar to that shown below.

VPPM Symbol sampled at 0.1 bit interval and 0.9 bit interval



Notice that we have a signal reduction due to the exposure time integrating over the light transition edge. We can then modify the equations as:

$$\frac{\varepsilon_b}{N_0} = \frac{a^2 \cdot \frac{T_2}{T}}{\alpha \cdot a \cdot \frac{T_2}{T} + \beta}$$