Enabling Green Video Transmission over Internet of Things

(2nd Quarter Report)

Dr. Ghalib Asadullah Shah (PI)

Next Generation Wireless Networking Lab,
Al-Khwarizmi Institute of Computer Sciences,
University Of Engineering and Technology, Lahore

June 17, 2014

Contents

List of Tables

List of Figures

Chapter 1

Deliverables

Deliverables in this quarter spanned all the three aspects of the project. The first aspect of the deliverable was with regards to low power video acquisition. Compressive sensing based techniques from the literature were to be explored, simulated and profiled. The findings are presented in Chapter 2 of the report. Encoding techniques which require less computational power are presented in the chapter. The encoder is tested with various decoders, which offer a trade-off between performance and complexity.

Secondly the feasibility of the existing power saving techniques in IEEE 802.11 were to be evaluated for IoM. For that we performed a simulation based study for power saving mechanisms. The results were to be compared in order to identify the most suitable power saving mechanism. This part of the deliverable is presented in Chapter 3 of the report. The operation and power saving capabilities of IEEE 802.11 standard's specified power saving protocols i.e. PSM and PSMP is explained and energy efficiency performance of the two algorithms is comprehensively analyzed through simulations.

Finally the hardware specifications were to be provided at the end of the second quarter. In the previous deliverable we presented the theoretical background to the possible hardware implementation of the green camera node, particularly so the implementation of the CSV Encoder, and the technological limitations in the implementation of CS at the acquisition stage(s). Continuing from there, in this deliverable we have presented which hardware modules (controller, camera e.t.c) are best suited to our camera node and why? We have also discussed the hardware layout for the initial and later stages of the development process.

Chapter 2

Compressive Video Acquisition

2.1 Rationale

State of the art video encoders like H.26x and MPEG have enabled the current revolution in multimedia streaming over the networks. Multimedia streams are encoded only once at the source using one of these encoders and transmitted over the networks. It is subsequently decoded at the end terminal each time it is played back. This framework necessitates a computationally complex encoder and a simpler decoder.

In the IoT setup, computational capabilities, memory and power are severely limited on the streaming nodes. Here a very simple encoder needs to be run at the low power source with the encoded stream decoded on powerful machines at the receiving end.

New video acquisition techniques have been proposed recently based on the body of work from the thriving new research area of compressive sensing. The techniques, therefore, are clubbed together under the umbrella of compressive video acquisition. During this quarter we have chosen some of the techniques from the compressive video acquisition literature which we have simulated, verified and profiled. These techniques will be implemented on hardware in the subsequent quarters.

2.2 Theory

In compressive sensing, an under determined system of linear equations is solved by relying on the information that the unknown vector is sparse in a certain basis. A signal vector $\mathbf{x} \,\epsilon\, \mathbb{R}^N$ can be expanded in an orthonormal basis $\mathbf{\Psi} \,\epsilon\, \mathbb{R}^{NxN}$ and this can be represented as $\mathbf{x} = \mathbf{\Psi s}$.

With a linear measurement matrix $\mathbf{\Phi}_{R\times N}$, with $R \ll N1$, the measurements are collected as

$$\mathbf{y} = \mathbf{\Phi\Psi s} + \mathbf{e} \tag{2.1}$$

where $\mathbf{e}$ is an unknown noise component upper bounded by a known power amount $\|\mathbf{e}\|_2 \leq \epsilon$.

If the unknown vector $\mathbf{x}$ is $\mathbf{k}$-sparse in $\boldsymbol{\psi}$ basis, i.e. $\mathbf{s}\,\epsilon\,\mathbb{R}^N$ contains at most k zeros, then it is recoverable accurately if the matrix $\mathbf{A} := \mathbf{\Phi\Psi}$ satisfies the restrictive isometry property (RIP) of order $\mathbf{k}$. Mathematically RIP is expressed as follows:

$$(1-\delta_k)\|\mathbf{s}\|_2^2 \leq \|\mathbf{As}\|_2^2 \leq (1+\delta_k)\|\mathbf{s}\|_2^2 \tag{2.2}$$

The unknown signal can be recovered by solving the following optimization program:

$$\hat{\boldsymbol{s}} = arg\min_{\tilde{s}} \|\tilde{s}\|_1 \tag{2.3}$$

subject to

$$\|\mathbf{y} - \mathbf{\Phi\Psi}\tilde{\boldsymbol{s}}\|_2 \leq \epsilon \tag{2.4}$$

Equivalently the optimization problem can be reconstituted as an unconstrained optimization problem in the following manner

$$\hat{\boldsymbol{s}} = arg\min_{\tilde{\boldsymbol{s}}} \|\mathbf{y} - \mathbf{\Phi\Psi}\tilde{\boldsymbol{s}}\|_2 + \lambda\|\tilde{\boldsymbol{s}}\|_1 \tag{2.5}$$

These results can be adapted for compressed video acquisition.

2.2.1 Encoder

Each frame of a video sequence is divided into virtual non-overlapping macro-blocks of fixed length N. Each maro-block is vectorized to form a vector of length N^2 and observed through an $N \times P$ measurement matrix

$$\mathbf{y}_t^m = \mathbf{\Phi x}_t^m \tag{2.6}$$

where $t = 1, 2, \cdots$ is the frame index and $m = 1, 2, \cdots, M$ is the macro-block index. Dimension of the measurement matrix Φ is $R \times N^2$ and in general $R \ll N^2$. Number of measurements R determines the amount of compression and also the quality of reconstruction. Complexity of this operation is $O(RN^2)$ for each macro-block. These measurements are quantized to an appropriate amount of bits and transmitted. Measurement through a random matrix and quantization are the only two operations performed at the encoder.

2.2.2 Decoder

The quality of reconstructed video largely depends on the number of measurements and the sparsity which according to the compressive sensing theory, should be proportional to the sparsity level of the signal. Thus the amount of compression and reconstruction quality largely depends upon the sparsity level of the video signal. The challenge therefore is to find an appropriate basis in which the signal is most sparse. In this quarter we have tried two different sets of basis: The DCT basis and the KLT basis. In the following we present the rationale behind using the two basis and the video reconstruction results that were obtained as a result.

2.3 CS based Video Acquisition Mechanisms

2.3.1 Discrete Cosine Tranform

DCT expresses an image or a portion of it as a linear combination of two dimensional cosine functions. Most of the natural images can be reasonably approximated in the DCT basis using a few non-zero coefficients corresponding to the first few low frequencies. The DCT basis functions for $\mathbb{R}^N$ are

$$\psi_k^N = \{\sqrt{\frac{2}{N}} cos(\frac{\pi k}{N}(n+\frac{1}{2}))\} \qquad k = 1, 2, \cdots, N-1 \tag{2.7}$$

2D cosine functions can thus be constituted as

$$\psi_{k_1,k_2}^{2D}(s,t) = \psi_{k_1}(s)\psi_{k_2}(t) \tag{2.8}$$

Since images/video frames are reasonably sparse in the DCT basis the optimization program in 2.5 can be solved for the signal representation in the DCT basis.

2.3.2 Karhunen Loeve Transform

The level of compression is determined by the number of measurements which in turn depends upon the sparsity level of the signal. Thus techniques which provided sparser representation of signals are desire-able. KLT is one such technique. It captures the entire energy of the signal and the metrics involved are dependent upon the measured signals, thus continuously updated.

In KLT, the signal is recovered in the space spanned by the eigen vectors of its correlation matrix. Two consecutive frames are highly correlated, thus the

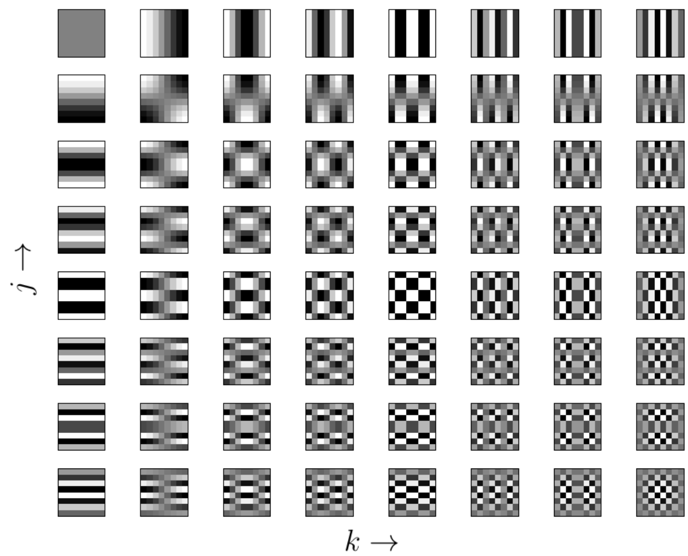


Figure 2.1: 2D cosines Visually

correlation matrix for m^{th} macro-block x_t^m in the t^{th} frame can be estimated using the region and the corresponding block in the previous frame x_{t-1}^m. A rectangular area of dimensions $w \times w$ centered at x_{t-1}^m is constituted and a sliding window of dimensions equal to the size of the macro-block is used to obtain the correlation matrix heuristically in the following manner [1]:

$$\hat{\boldsymbol{R}}_2^m = \sum_{i=1}^{B} \mathbf{d}_i \mathbf{d}_i^T \tag{2.9}$$

Then the eigenvalue decomposition of the correlation matrix $\hat{\boldsymbol{R}}_2^m = \boldsymbol{Q\Lambda Q}^T$ is used to find the basis for recovery $\boldsymbol{\Psi}_{2,KLT}^m = \mathbf{Q}$. Subsequently the optimization problem can be constituted in the following manner

$$\hat{\boldsymbol{s}} = arg \min_{\tilde{\boldsymbol{s}}} \|\mathbf{y} - \boldsymbol{\Phi\Psi}_{2,KLT}^m \tilde{\boldsymbol{s}}\|_2 + \lambda \|\tilde{\boldsymbol{s}}\|_1 \tag{2.10}$$

which leads to the recovery of the signal using $\hat{\boldsymbol{x}}_2^m = \boldsymbol{\Psi}_{2,KLT}^m \hat{\boldsymbol{s}}_2^m$

For higher order estimations, the first frame is reconstructed using the DCT basis, which is then used as the reference frame for the subsequent frames. For order two KLT basis, also depicted in figure 2.2, the first frame is re-estimated in the KLT basis, using the recovered Frame two as the reference. Figure 2.3 depicts the sequence for order four KLT.

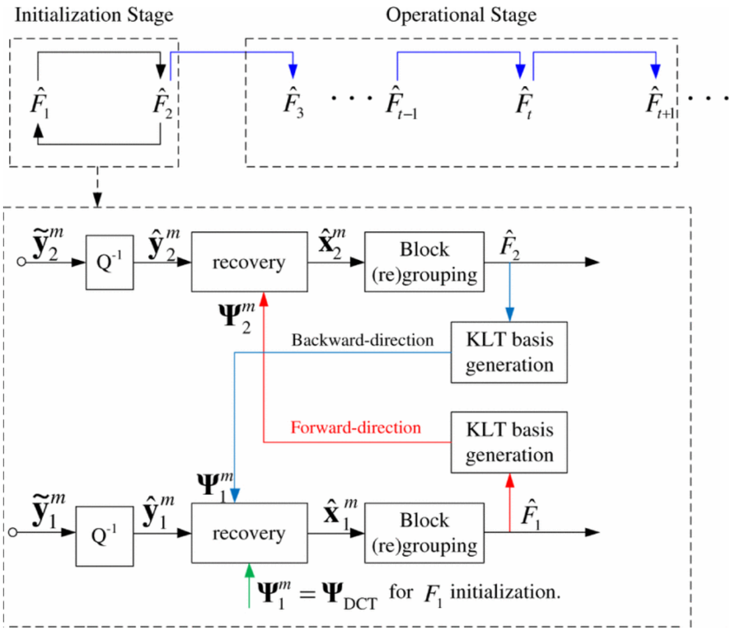


Figure 2.2: Decoder Block Diagram [1]

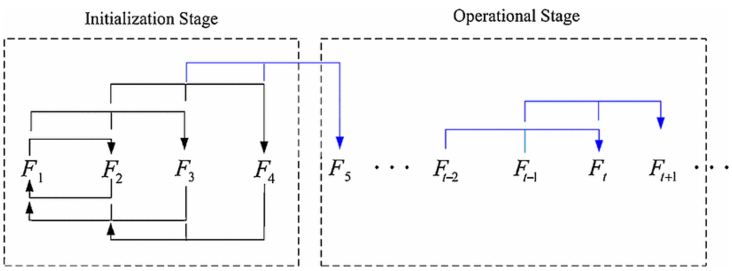


Figure 2.3: Sequence for order 4 [1]

2.4 H.264

The H.264 standard was published in 2003 and is based on MPEG-4. There are a number of advantages provided by H.264 video coding standard [2]. H.264 provides support for bit rate adaptation that increases flexibility for packet level transmission at the network adaptation layer (NAL). Although it is a video compression method for converting real time digital video sequence into bitstreams that requires less transmission capacity. The H.264 uses pre-defined tools available in decoding phase termed as profiles. Each profile of H.264 defines certain tools. In encoder phase also H.264 have variety of tools available. For transmission it provides number of support for illuminating transmission errors and packeting in compressed format.

2.4.1 Coding Fragments of H.264

H.264 profile have following coding parts in general.

- Intra-coded slice (I slice)
- Predictive-coded slice (P slice)
- Context-based Adaptive Variable Length Coding (CAVLC) for entropy coding

The architecture of H.264 increases the capabilities offering solutions for encoding phase that support temporal, spatial and SNR qualities. The H.264 encoder is more complex than any previous video coding standard. There are computation and memory overheads in the development of embedded encoder/decoder. It does not offer any compatibility with the older versions or video coding standards. To generate scalable compressed video bit streams, punctured turbo codes are applied. To get overall performance improvement, usually following steps are performed

- Predict a new frame from a previous frame and only specify the prediction error.
- Prediction error will be coded using an image coding method.
- Prediction errors have smaller energy than the original pixel values and can be coded with fewer bits. In this standard the system is broken into two layers [3]: Network Abstraction Layer (NAL) and Video Coding Layer (VCL)

Video information is transformed into bitstreams through VCL, after the conversion the NAL layer maps the transformed bitstreams into NAL units those are HDLC-like and byte-oriented transportation-layer before delivery. Usually the NAL header is composed of three fields. NAL unit header is the first byte after the NAL unit code prefix. The bits are categorized as following.

1. Forbidden-bit (1 bit)
2. NAL-storage-idc (2 bit)
3. NAL-unit-type (5 byte)

forbidden-bit is used to indicate whether the NAL unit is corrupted or not. *NAL-storage-idc* is of two bits, which have relative importance, when the picture is buffered. The *NAL-unit-type* is of one-ten bits according to need. Usually they are referred as reserved bits. Particularly three operations are performed during the mapping phase those are

1) Byte Alignment: NAL unit adds byte header, that defines the categorization of the NAL unit. Synchronization is performed in the NAL unit through byte sequence.

2) Emulation Prevention: Emulation prevention bytes are used to interleave NAL unit payload data [4]. Those bytes prevent accidental data generation within the payload. These bytes are termed as *start code prefix*, which are inserted in the data pattern. NAL unit structure are used in both bitstream-oriented and packet-oriented transport systems.

3) Framing: H.264 slices frames according to five types, which are listed below

- I(Intra) Slice: This type reference only itself. Usually it is the first received image of the video sequence or a still image. All the first frames from video sequences needed to be built from I-slices
- P(Predictive) slice: It takes reference from decoded or predicted slices for construction of video sample/image. Prediction is mostly not accurate therefore some residual images may be added.
- B(Bi-Directional) Slice: It takes reference from future and former P or I slices, except of that Bi slices are very similar to P slices. For the reason, I an dP slices are decoded after B slices.
- SI and SP (Switching) Slices:Usually they are used for transition between video samples of different natures and type. There probability or usage is although very less.

2.4.2 H.264 Main Modules

Spatial dependency layer in H.264 have the essential requirement of its own prediction module for performing intra and motion-compensated predictions within the layer. Another module that manages the scalability of quality is termed as SNR refinement module. In the inter-layer prediction module, the dependency is managed between spatial layers by reusing residual signal and motion vectors so as to improve compression efficiency. All the modules are finally merged in a multiplex, a single integrated scalable bitstream having different spatial, temporal and SNR levels. The process is combination of forward and inverse (encoding and decoding) path as depicted in figure-2.4. The video frames are structured in macroblocks upon which prediction is performed using either Inter or Intra predictions. After the transformation and quantization process, the resultant product is forwarded to **Entropy Encoder Module**. The output packets are finally formed in the **Network Abstraction Layer (NAL)** module. Decoding or Inverse path involves reconstruct of Macroblock data from previously transformed module which consist of **Deblocking Filter** [5] and **Transform and Quantization(ITQ)**.

Encoding Modes

Generally there are six encoding modes presented in academic and commercial bases, which are listed below

- Single Pass-Constant Quantizer
- Single Pass-Constant Rate Factor
- Single Pass-Bitrate
- Two Pass- Average Bitrate
- Two Pass-File Size
- Lossless Mode

2.4.3 Prediction Model

This section individually explains the characterization of all types of scalability [6]. It is generally suitable if the encoding and decoding complexity tallies itself proportionally with accordance to temporal and spatial resolution. A hierarchy should be established for video compression tools which involves scalability complexity.

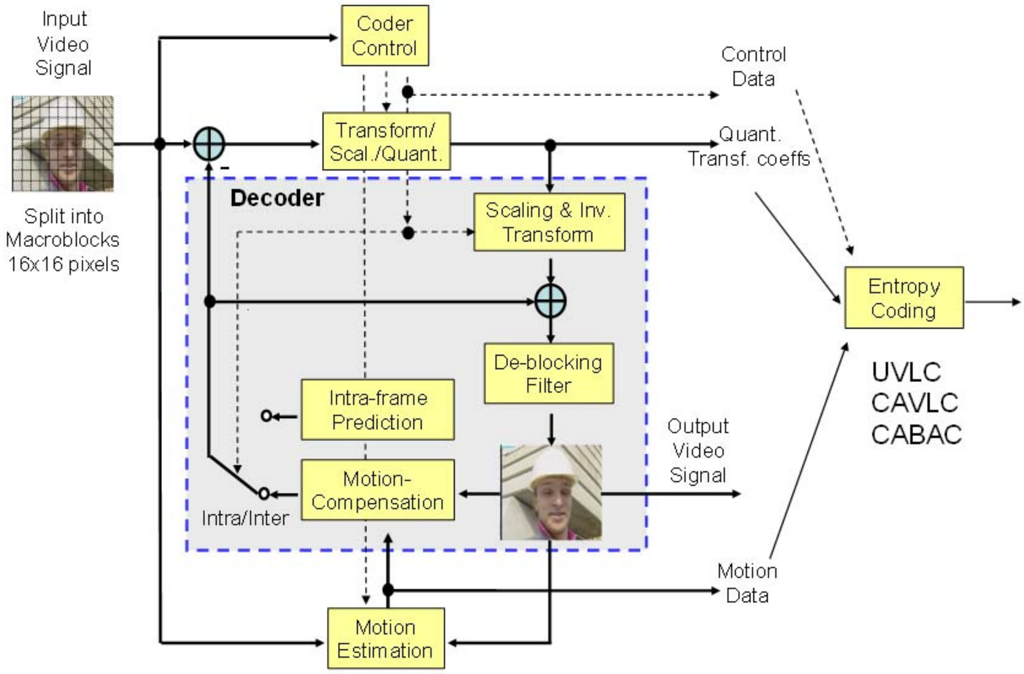


Figure 2.4: H.264 Codec Block Diagram [?]

Temporal Scalability

In this category video stream is transferred as subset of bitstream. Video frames are classified as three distinct types: I (intra), P (predictive) and B (Bi-predictive) as already known. In this procedure, three types of simplified motion compensation, termed as Temporal Prediction.

Temporal Prediction

- **No Motion Compensation**

Work well in stationary regions

$$f(t, m, n) = f(t - 1, m, n) \tag{2.11}$$

where t=time period, m=macroblocks and n=number of frames / block

- **Uni-directional Motion Compensation**

For uncovered ranges, its performance does not work well, as new objects are appearing

$$f(t, m, n) = f(t - 1, m - d_x, n - d_y) \tag{2.12}$$

where t=time period, m=macroblocks and n=number of frames / block, x and y are the coordinates
• **Bi-directional Motion Compensation**

$$f(t,m,n) = w_b f(t-1, m-d_{b,x}, n-d_{b,y}) + w_f f(t+1, m-d_{f,x}, n-d_{f,y}) \quad (2.13)$$

Spatial Scalability

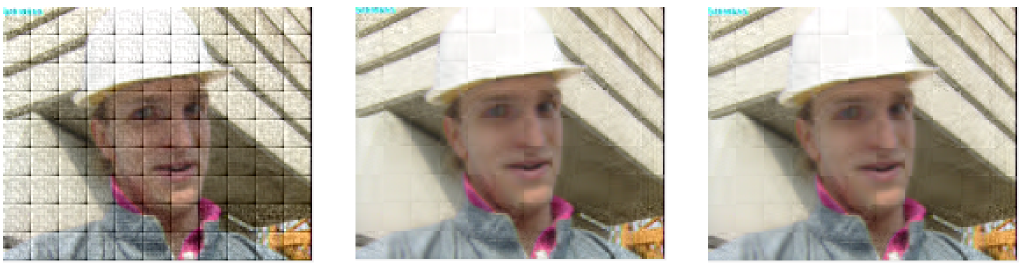
Layered structure is used in spatial scalability; by layering the improvement of lower layer resolution is achieved. In this domain previously there have been two prediction types introduced. 1) Extended Spatial Scalability
2) Inter-Layer Predictions
a. Inter-Layer Motion Predictions
b. Inter-Layer Intra Texture Predictions
c. Inter-Layer Residual Predictions

2.5 Results and Comparison with H.264

Simulation Setup

Simulations were run on three test sequences: foreman, hall monitor and container. The number of measurements were fixed at 512. LASSO algorithm was used for solving the optimization program. We experimented with the sparsity penalty λ in the optimization program. The simulation runtime decreases with increasing number of measurements while it increases as we decrease λ. The following frames are frame number 12 of all the sequences recovered in DCT, KLT basis with orders 1 and order 2 respectively.

The graph in 2.8 depicts the increase in PSNR as the bit rate is increased. The bit rate is calculated as follows:



(a) DCT basis (b) KLT order 1 (c) KLT order 2

Figure 2.5: Foreman Sequence, Number of measurements = 512

(a) DCT basis (b) KLT order 1 (c) KLT order 2

Figure 2.6: Container Sequence, Number of measurements = 512

(a) DCT basis (b) KLT order 1 (c) KLT order 2

Figure 2.7: Hall Monitor Sequence, Number of measurements = 512

$$bitrate(kbps) = w \times h \times q \times R_f / R_c / 1000 \tag{2.14}$$

where
w is the width of the frame
h is the height of the frame
q is the quantization level
R_f is the frame rate
R_c is the compression ratio which is computed as the ratio of the number of measurements to the size of the vectorized macro-block

PSNR values for different systems for video encoded using different systems are not comparable directly. But they can be compared indirectly. It is well known that for H.264, PSNR 30dB to 40 dB map to acceptable visual quality of streaming data. As depicted, H.264 can achieve these results for very low bitrates. For compressive sensing techniques however, we have found that PSNR values above 20 dB correspond to acceptable quality of video. Encoding times, however, for H.264 are exorbitantly high. These times were computed for encoding only first 12 frames of forman video sequence. For playing the video back at 30fps, the inter frame interval is $\frac{1}{30}$th

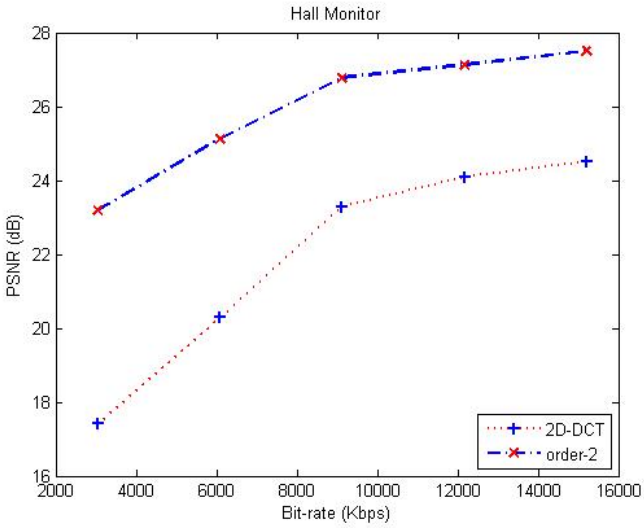


Figure 2.8: Compressive Sensing: PSNR vs bitrate

of a second. On an Intel core i7 machine, with 4 GBs of RAM installed, encoding time per frame for the minimum specs is greater than $\frac{1}{2}$ a second. The detailed results are provided in plots 2.9 and 2.10.

What an end user can expect from a service (in our case video) can vary, depending on the needs and interests of the particular user or the particular application. Quality of Experience is a performance matrix often used to get an overall measure of the quality of service. This technique is survey based, therefore the greater, and the more diversified the sample, and farer will be the results. In this case we showed the twelfth frame of one of three different sequences, recovered in DCT and KLT-II bases, as well as the original frame itself. Randomly selected individuals were chosen to rate the videos on a scale of 0 - 5. The averaged results are shown in the bar chart in Figure 2.11.

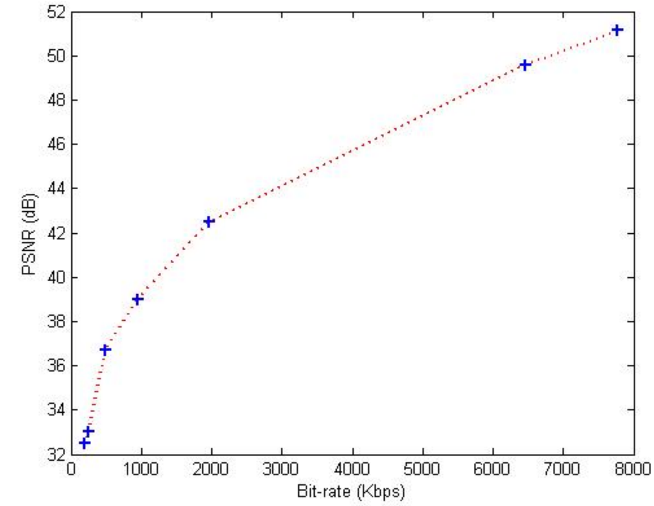


Figure 2.9: H.264: PSNR vs bitrate

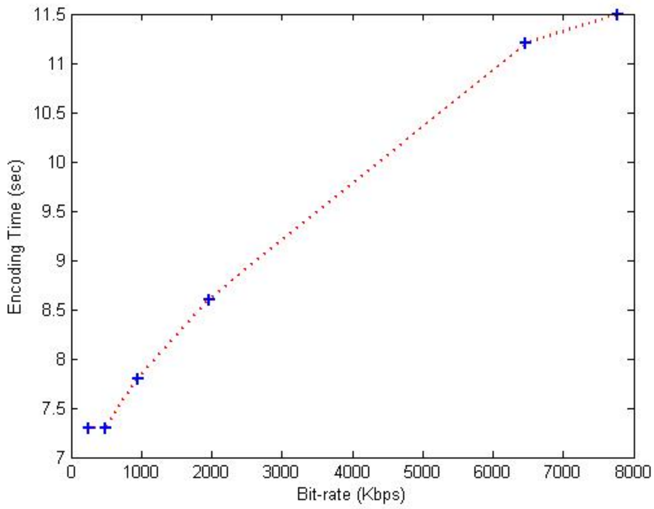


Figure 2.10: H.264: PSNR vs Encoding Time

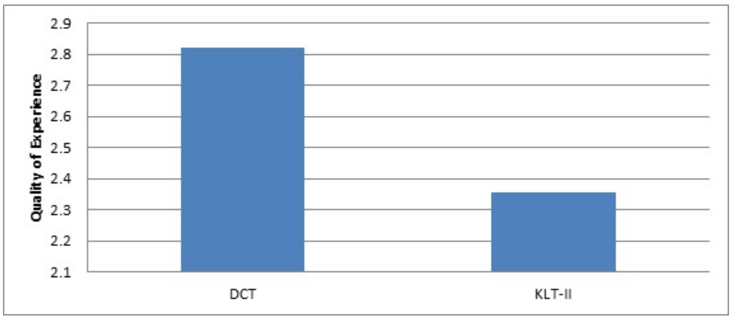


Figure 2.11: H.264: Mean Opinion Score for DCT and KLT-2 basis on a scale of 0-5

Chapter 3

Limitations of Existing IEEE 802.11 Power Saving Techniques

3.1 Introduction

The explosive and ubiquitous adoption of Wireless Sensor Networks (WSNs) due to its low cost sensor technology has enabled by large-scale networks of small devices capable of harvesting information from the physical environment, performing simple processing on the extracted data and transmitting it to remote locations [7]. 'Internet of Things' (IoT) refers to the possibility of connecting sensors, or any other device or 'thing' to the Internet. It has the potential to significantly influence our lives and the way we interact with the devices such as sensors, actuators, mobile phones, home automation devices, smart grid devices, etc [8]. However, recently due to the availability of low cost multimedia devices i.e. CMOS cameras and microphones, the Wireless Multimedia Sensor Networks (WMSNs) have gain lots of attraction. In WSMNs ubiquitously distributed low-power low cost devices communication to retrieve multimedia content from the physical environment in the form of video and audio streams, still images, or scalar sensor data. Conventionally, these tiny devices report data to a centralized network manager or gateway or application server device, which is responsible to store and fuse the data coming from different devices.

The Internet of Multimedia (IoM) is an enhancement to the IoT, whose prime objective is to enable video streaming as part of the realization of IoT. In IoM, resource constraint low-power low-cost heterogeneous multimedia devices can be connected and each device can be globally accessible by a unique

IP address with the same spirit as of the computers and other networking devices connected through the Internet. This approach enables a wide range of applications in the areas of home and building automation, factory monitoring, smart cities, transportation, smart grid and energy management [9]. However, the delivery mechanism of the current Internet architecture, offers 'best-effort' transmission with no guarantee of successful packet delivery or the time of its delivery. Despite the fact that 'Best-Effort' transmission mechanism is suitable for non-real time data communication which is delay tolerable but requires error-free delivery, whereas this approach is not appropriate for time-constraint multimedia content i.e. video and audio.

Specially, real-time multimedia traffic that is continuous in nature requires stringent communication requirements as compared to data communication such as bandwidth, delay, jitter and reliability. These network performance requirements are referred as the Quality of Service (QoS), which represents the level of user experience. However, direct implication of these protocols for IoM is not straight forward, since these protocols are not designed considering energy constrained devices. The current standardization activities of providing Internet-connectivity to 'Things' [10] are not focused to address the challenges of provisioning multimedia objects over 'Internet of Things'. The main obstacles of realizing IoM that enables Internet access to WMSN are limited available power, limited available capacity, and heterogeneity of multimedia devices. Many researchers have investigated a variety of techniques to limit the power consumption of WMSNs. However, these issues have not been addressed considering WMSNs based on IoM architecture.

ZigBee that is based on the IEEE 802.15.4 standard is promoted for IoT, since it is designed for tiny network devices performing simple operations. However, the maximum data rate supported by ZigBee (250 kbps) is not feasible for most of the IoM applications. Especially for real-time multimedia communication, the multimedia devices cannot provide satisfactory user experience with data rate of 250 kbps. For this reason, IEEE 802.11 standard [11] is suggested in the literature for wireless multimedia networks, since it provides a high data rate communication model and holds a great potential for WMSNs. ZigBee and other IEEE 802.15.4 based protocols have been considered for WSNs applications due to their energy-efficient design. However, recently developed power-efficient Wi-Fi models promise multiple years of battery lifetime, have become a strong candidate in this domain [12]. Reuse of existing Wi-Fi infrastructure offers cost savings and faster deployments. Widely deployed Wi-Fi networks reduce the infrastructure cost to a minimum while improving the total cost of ownership. Wi-Fi devices have the advantage of native IP-network compatibility, which is a big plus for IoT.

Well-defined and universally accepted IP connectivity overcomes the need of expensive gateway requirements or any network address translation (NAT). In addition, IEEE 802.11 provides various mechanisms to support different types of traffic classes with desired QoS and the higher data rate provides support for better user experience for multimedia communication. Moreover, it can support transmission range corresponding to the view range of video devices that makes it a practical choice. Furthermore economy of scale is another important advantage of Wi-Fi with an expected 22 percent annual growth rate between 2010 and 2015 [13]. For these reasons, Wi-Fi has already been widely accepted for many commercial off-the-shelf video devices, which are largely deployed for video surveillance and monitoring applications, making it a good candidate for IoM if the energy efficiency mechanisms of IEEE 802.11 are devised comparable to IEEE 802.15.4.

Recently, lots of efforts have being made to improve energy efficiency of IEEE 802.11 based WLANs. Most of the studies focused on reducing collision probabilities, duty cycle, delay, etc. However, to the best of our knowledge none of them have the key optimization parameters considering resource constraint wireless multimedia devices running real-time video traffic applications. In this work, we promote feasibility of IEEE 802.11 for IoM and further enhance its power saving mechanism to effectively improve energy efficiency considering the key optimization parameters for resource constraint multimedia devices.

3.2 Power Saving Mechanisms in IEEE 802.11

In this chapter, we discuss the power saving mechanisms specified by the IEEE 802.11 standard. The protocols include Power Save Mode (PSM), Automatic Power Save Delivery (APSD) and Power Save Multi Poll (PSMP). Another power management technique is introduced in IEEE 802.11n and IEEE 802.11ac, for wireless stations equipped with multiple antenna i.e. operating in MIMO mode. In this technique, if the network operation and performance requirement allows then only a subset of the available antennas are kept active and rest of the antennas are switched off saving significant energy. However, since we are considering small low power devices where multiple antennas are not feasible thereby not commonly used. Similarly, the APSD power saving mechanism is specified by the IEEE 802.11e standard. The APSD provides relative QoS between different traffic categories by using different defer durations and backoff time. But the overall network resources and energy utilization is still sub-optimal due to backoff and collision probability. Therefore, in IEEE 802.11n an improved scheduling algorithm for power saving is proposed referred as PSMP. PSMP is similar to APSD, however there is no backoff required as the schedules for both uplink and downlink transmissions are scheduled in Time Division Multiple Access (TDMA) manner. In the rest of the chapter we describe the operation of both PSM and PSMP protocols.

3.2.1 Power Save Mode (PSM)

IEEE 802.11 standard specifies two different states for a station: Continuous Aware Mode (CAM) or Power Saving Mode (PSM). In CAM, a station is continuously capable of transmitting or receiving packets. Whereas, in PSM a station only periodically switch on its radio to send or receive data and the rest of the time it remains in the sleep mode or doze state by switching off its radio. In an infrastructure based WLAN, the access point (AP) periodically, usually after 100 msec, broadcasts a Beacon Frame. These frames include Traffic Indicator Map (TIM) field which notifies stations if there are any packets buffered at the AP for them. Station in PSM periodically wake-up to receive beacons and check if their Association ID (AID) is identified in the TIM. If there are pending packets for a station it stays awake and contend for the channel to send a Power Save Poll (PS-Poll) frame to the AP. The AP then transmits the pending packet and it sets MoreData flag as well if there are more packets buffered at AP. The station retrieves all the pending packets by sending PS-Poll frames till MoreData flag is set to 0, after that the station goes back to doze state.

The graphical representation of PSM operation is shown in Fig. 3.1. The PSM significantly reduces power consumption as compared to CAM. However, the downlink traffic is introduced with some delay due to the listen interval. The delays are variable and QoS is not considered that is why PSM is not suitable for delay sensitive applications like VoIP. In addition, PS-Poll frames overhead consume significant time and energy resources. A station that retrieves its pending data after all other nodes needs to hear all the transmissions till its turn, these overhearing results in power wastage. Moreover, no matter there is any packet pending for a station or not, it must wake-up to receive periodically beacon frames which results in useless power utilization.

3.2.2 Power Save Multi Poll (PSMP)

The legacy-PSM and APSD power saving mechanisms notify the stations about their pending packets buffered at the AP, the stations then contend for the channel to transmit PS-Poll or trigger frames to retrieve the packet.

However, when multiple stations contend the channel and the probability of collision among the request frames increases. These collisions results in bandwidth wastage and additional energy consumption at the power saving stations. In IEEE 802.11n [11] Power Save Multi-Poll (PSMP), these packet transmissions are not required, thus the collisions are avoided.

In PSMP power saving mechanism, AP notifies PSMP enabled station when they have to stay awake using the beacon frames. Unlike its predecessor power saving techniques, PSMP enables the stations to stay awake only on specific service times and when they are required to be in a receiving or a transmitting state. The AP can schedule the transmission opportunities depending upon the application constraints like delay and/or bandwidth constraints. The station in their PSMP Uplink Transmission Time (UTT) or Downlink Transmission Time (DTT) can transmit or receive without any clear channel assessment, respectively. It is worthy to mention here that during the service periods the stations are not the TXOP holders; instead the AP schedules their transmission in its own TXOP. The schedule of the PSMP-UTT and PSMP-DTT of each station is shared using PSMP Action Frames which includes the exact time of start of each transmission and how long it will last. The subsequent transmissions can be separated with RIFS or SIFS. The PSMP mechanism is more efficient as compared to its predecessor, in terms of energy saving as well as bandwidth utilization. The graphical representation of PSM operation is shown in Fig. 3.2.

In case a station is assigned PSMP-UTT duration that is not enough to transmit all the packets in the transmission queue of the station, then it transmits as many packets as it can within the assigned PSMP-UTT duration and request more or larger UTT duration by sharing its queue size or TXOP it requires within the QoS control field of the QoS data frame. The requested PSMP-UTT duration can be assigned depending upon the AP resources, in the present PSMP sequence or in the subsequent PSMP sequence. Moreover, an AP can initiate back-to-back PSMP sequences as well to facilitate additional resource allocation and/or error recovery. These consecutive PSMP sequences is referred as PSMP burst, and this PSMP burst can only be used for QoS data frames and the AP can end a PSMP burst by transmitting a CF-End frame.

If a station doesn't start its transmission in start time of its schedule, then the AP transmits a PSMP recovery frame (similar to PSMP Action frame) after PIFS duration which notifies the station about its updated PSMP-UTT duration that is reduced due to the time wastage by the station. However, the schedule of the other stations remains unchanged. On the other hand, if the transmission has failed then the AP may allocate PSMP-UTT duration in the present PSMP sequence or it may allocate temporal resources in the next PSMP sequence to accommodate the retransmissions. The AP may transmit BlockAck response with the PSMP-DTT transmissions notifying what frames needs to be retransmitted and when they can be transmitted (schedule for retransmission). If a station requires more time for retransmission than the allocated duration, it can transmit packets in non-service period using EDCA. However, these packet transmissions are acknowledged in the PSMP-DTT period of the next PSMP sequence.

3.3 Performance Analysis

In this chapter we compare the performance of the IEEE 802.11 power saving protocols PSM and PSMP. We implemented the PSM and PSMP power saving protocols in NS-2. In case of both PSM and PSMP protocols the operations performed by the AP and the stations are different. In NS-2 we created separate executable files for both AP and stations. To understand the individual operations performed in the AP and stations we designed state diagrams for both PSM and PSMP protocols as shown in figures 3.3-3.6. The PSM model is available for NS-2.1b8a and NS-2.33. Since, NS-2.33 is much later version as compared to NS-2.1b8a and lot of bugs have been fixed in NS-2.33 which were identified in previous versions, therefore we opted NS-2.33 and integrated PSM in it and then implemented basic functionality of PSMP from scratch in NS-2.33 for comparison with PSM and standard 802.11 network simulation scenario with no power saving mechanism employed.

Multimedia communication by resource constraint nodes particularly over lossy wireless links, is a challenging network operation which depends upon critical network parameters like energy consumption by nodes, network lifetime, delay, jitter etc. Energy consumed by the node depicts the amount of power it consumed to transmit some particular amount of multimedia traffic. This energy consumption can be used to estimate the network lifetime, which means the time between network start time till the first node (anyone) utilized all of its battery energy i.e. node dies. Similarly, packet delay which is referred to the time taken for a packet to be transmitted across a network from source to destination, which reflects the satisfactory user experience specially in real-time multimedia communication. Another important performance metric for multimedia traffic is jitter which is referred as the time difference between the inter-packet arrival of packets to the destination. Jitter needs to be below a specific level as per the application requirements so that the user may not experience any lag in the multimedia content. In the performance analysis below we vary the packet size, packet interval, and traffic generation rate, to evaluate the performance in terms of above mentioned

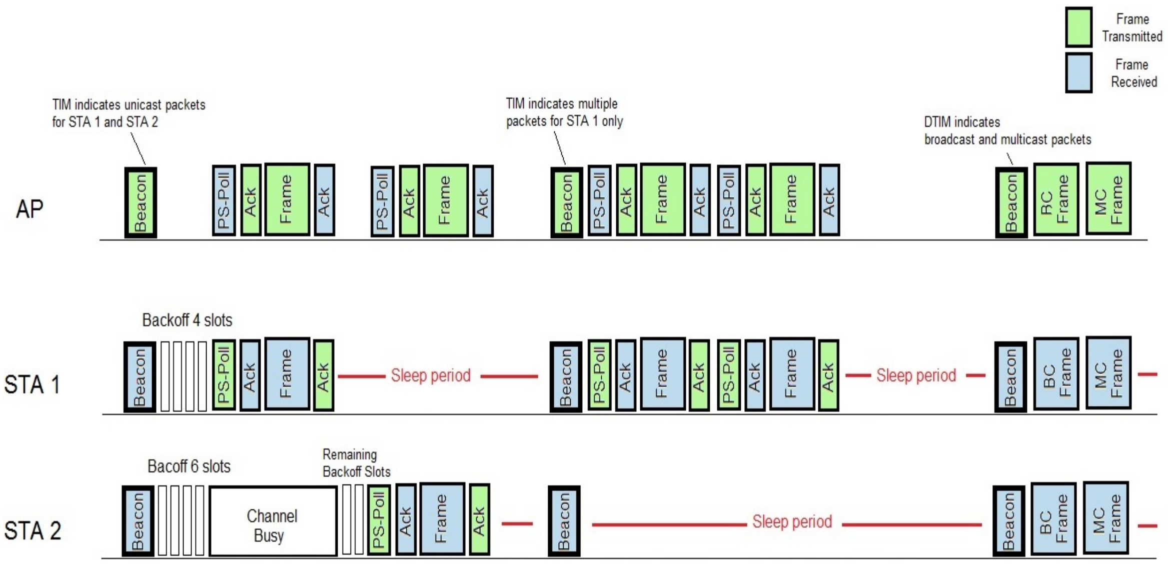


Figure 3.1: Basic PSM operation

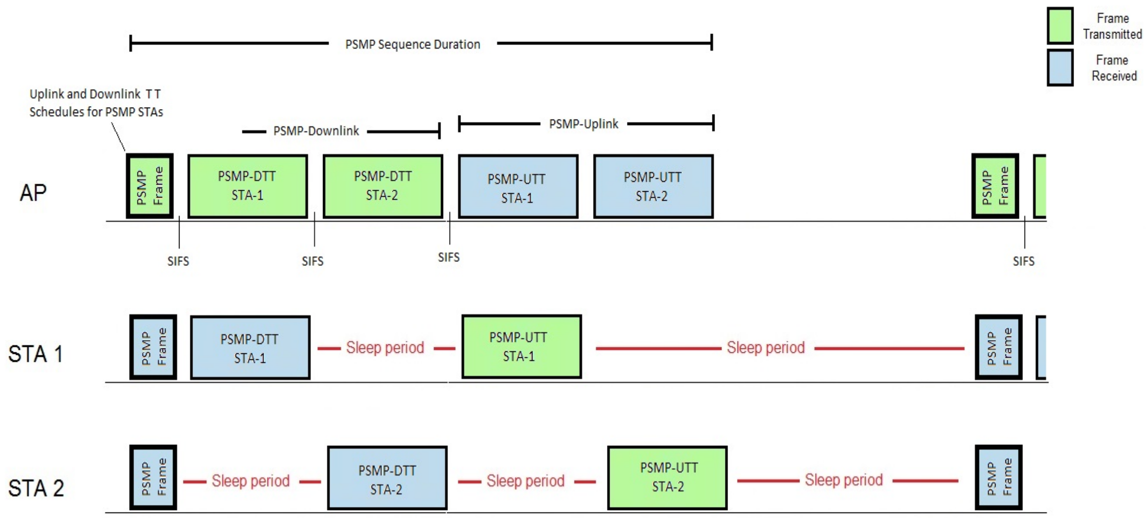


Figure 3.2: Basic PSMP operation

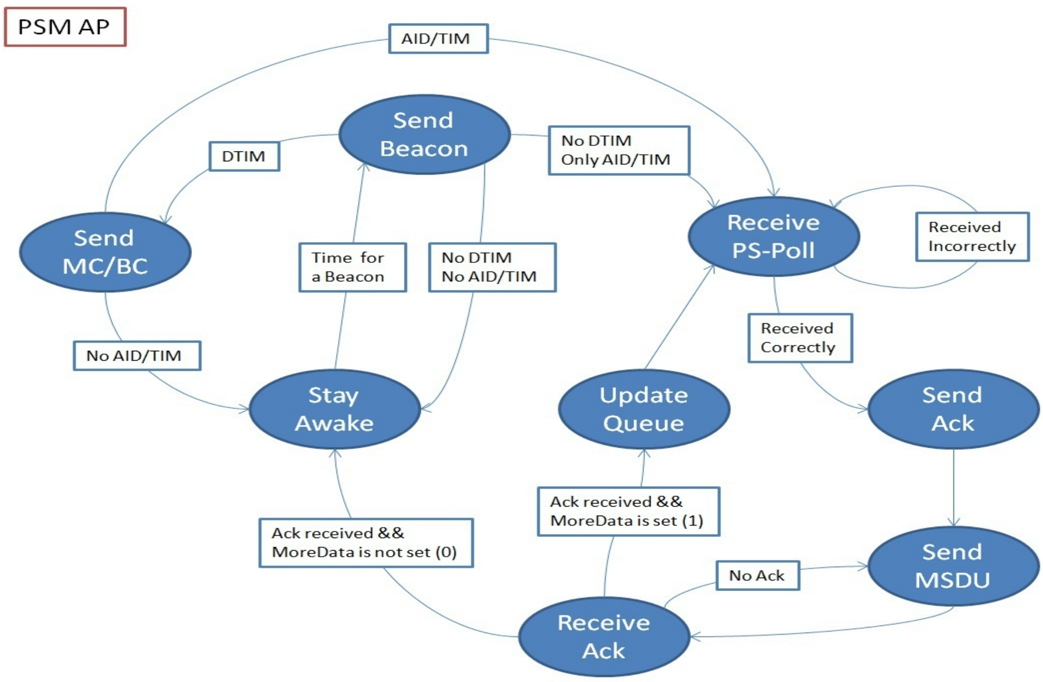


Figure 3.3: PSM AP

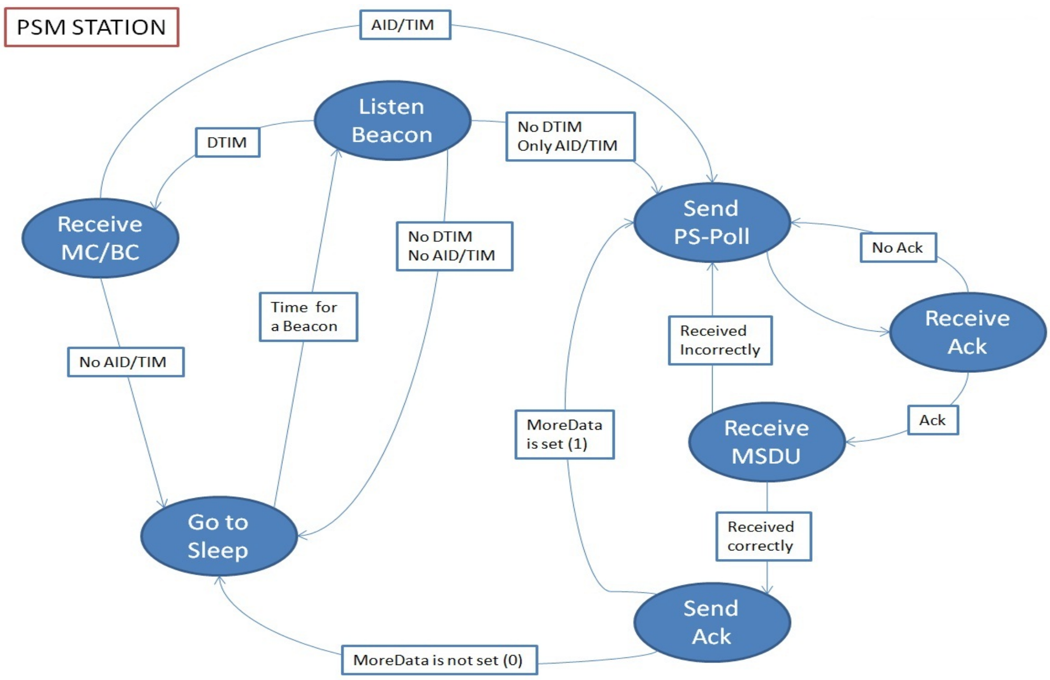


Figure 3.4: PSM STA

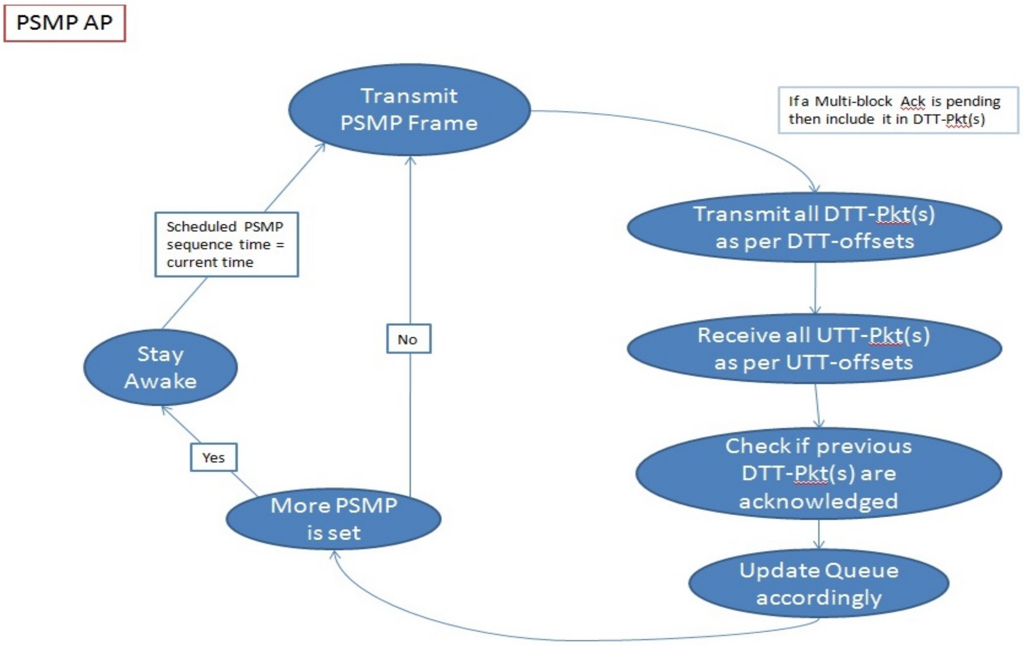


Figure 3.5: PSMP AP

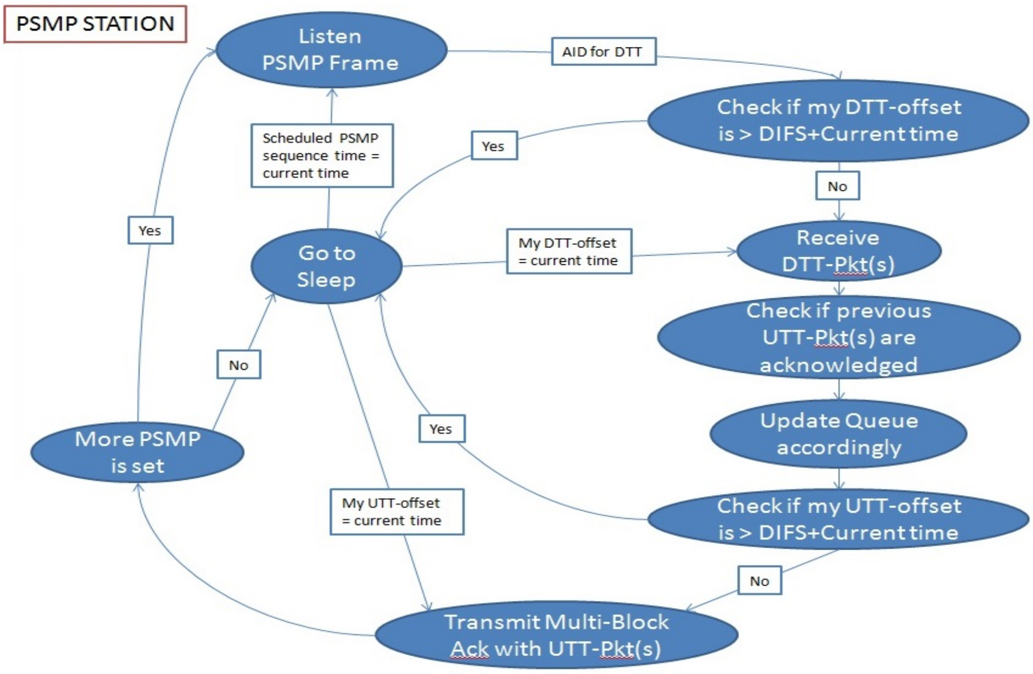


Figure 3.6: PSMP STA

performance metrics compared with the IEEE 802.11 standard without any power saving mode.

We simulated an infrastructure based network scenario, in which an access point (AP) as well as two wireless stations i.e. STA-1 and STA-2, are generating constant bit rate (CBR) traffic. The network topology simulated is shown in Fig. 3.7. AP is transmitting to both stations and the stations are transmitting to AP, i.e. both uplink and downlink traffic is active. We simulated the network for 100 secs and evaluated the performance metrics. Unless specified we used packet size of 512 bytes and packet interval of 0.01 secs. To evaluate the effect of packet size we varied the packet size and computed the energy consumed by IEEE 802.11 standard without any power saving mechanism (we will refer it as No-PSM), PSM, and PSMP. In Fig. 3.8 it is shown that as the packet size increases the energy efficiency of No-PSM and PSM increases. It is due to the increase in effective throughput, as for larger packet size the fraction of time utilized by the contention decreases and vice versa. However, the effect of packet size on PSMP is negligible, since there is already no backoff and effective throughput is much higher.

Similarly, we varied the packet interval time and observed the performance of the three protocols. In Fig. 3.9 it can be seen that when the packet interval time is increased the energy consumed by the wireless nodes decreases. It is because when packets are generated after longer period of time the wireless nodes get longer time to sleep and save energy. It should be noted that the AP doesn't operate in any power mode i.e. it never sleeps, however it supports power saving protocols by queueing or buffering the packets destined for stations that are in sleep state.

To evaluate the effect of high traffic generation, we varied the CBR rate and kept the packet size same i.e. 512 Bytes. As a result of generating heavy

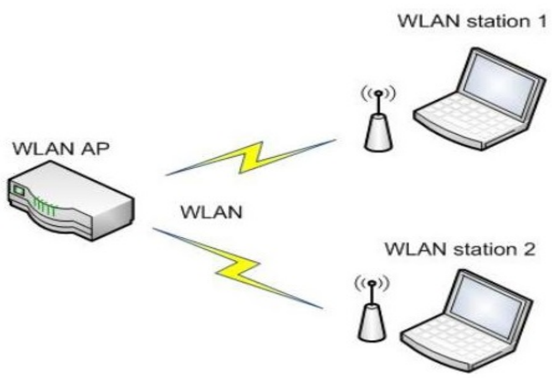


Figure 3.7: Network Topology

traffic keeping the packet size same, the nodes need to stay awake longer and thus consuming large amount of energy, as shown in the Fig. 3.10. However, in case of PSMP there is negligible effect on energy consumption when the CBR rate is changed, it is because in our simulations we restricted the maximum DTT and UTT durations and even if the traffic is very heavy. But prolonging the time packets spend in the queue i.e by not transmitting them to save energy results in buffer overflow and/or delayed transmission effecting user experience. Therefore, it is a trade-off between saving energy and providing QoS specifically when the traffic generation rate is high. However, in PSMP to provide enhanced QoS the PSMP burst can be used as explained in Chapter 3.2.

In addition we analyzed the end-to-end delay for these protocols. The PSM and PSMP are designed for single hop communication and the IEEE 802.11 standard does not specify its operation in multi-hop scenario. Therefore, the delay among individual packets is not significantly different from each other. However, due the contention free operation of PSMP the delay is significantly lower as compared to PSM and standard protocol without power saving mode, as shown in Fig. 3.11. Similarly, we observed the variation in jitter for all these protocols. As shown in Fig. 3.12 the variation in these protocols is nearly same, however the on average the instantaneous jitter value for PSMP is higher then other protocols. It is because in PSM and standard protocol, the stations don't sleep until they retrieve all the pending packet buffered for them in AP's queue. However, in PSMP the Uplink and downlink transmission time is scheduled and limited. Thus, nodes try to transmit and receive as many packets as possible but within the allocated time in the PSMP sequence. As mentioned earlier using PSMP burst more opportunity can be shared by AP with stations but at the cost of additional energy consumption.

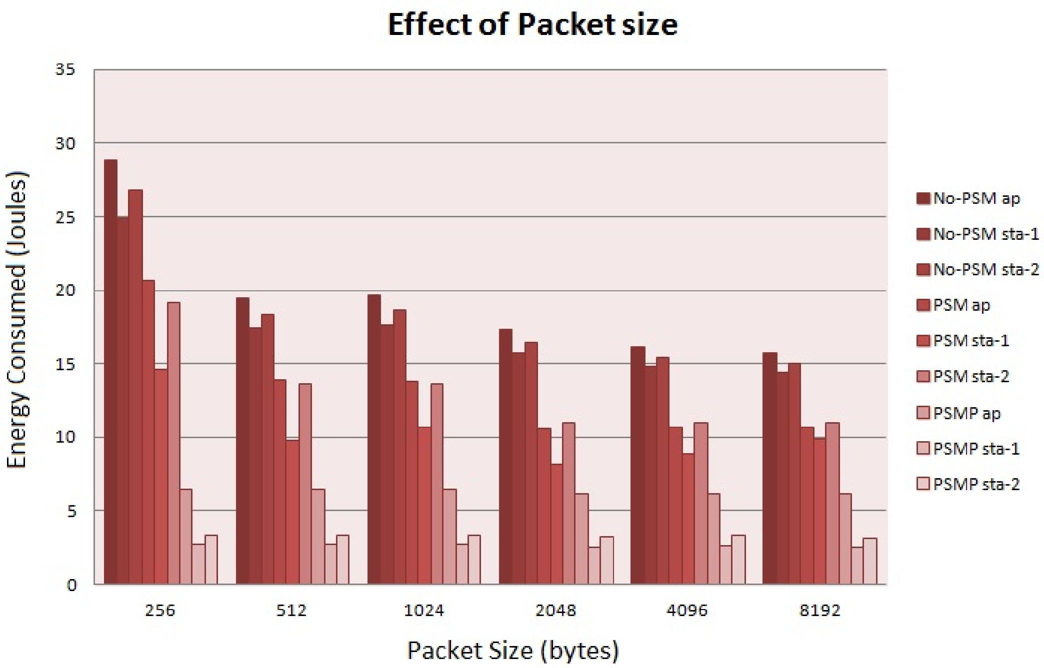


Figure 3.8: Effect of Packet Size

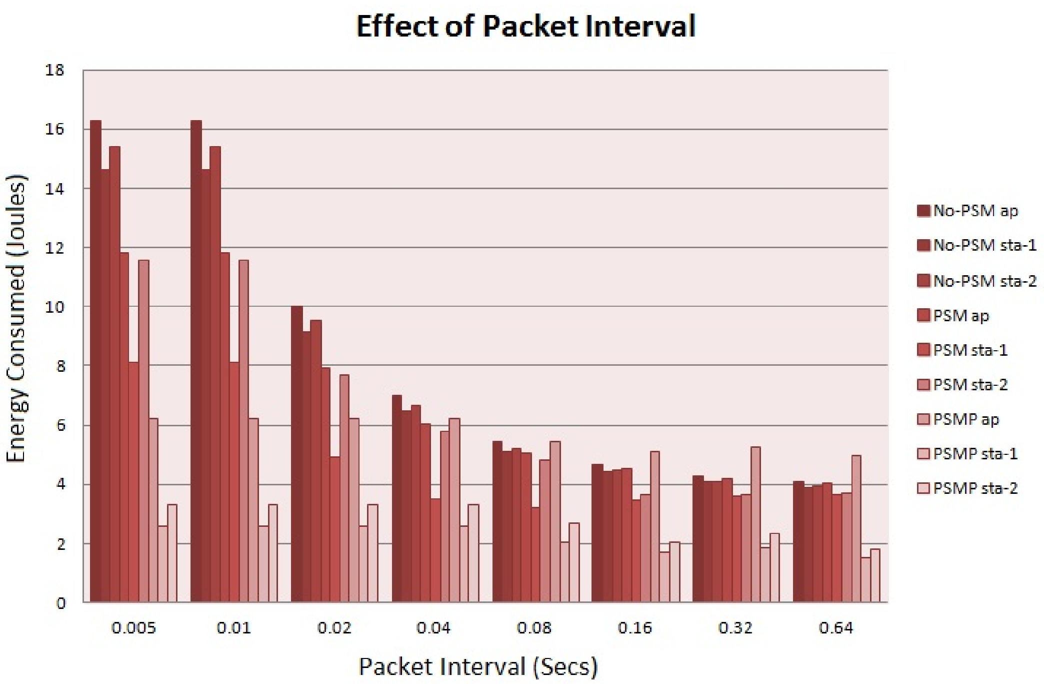


Figure 3.9: Effect of Packet Interval

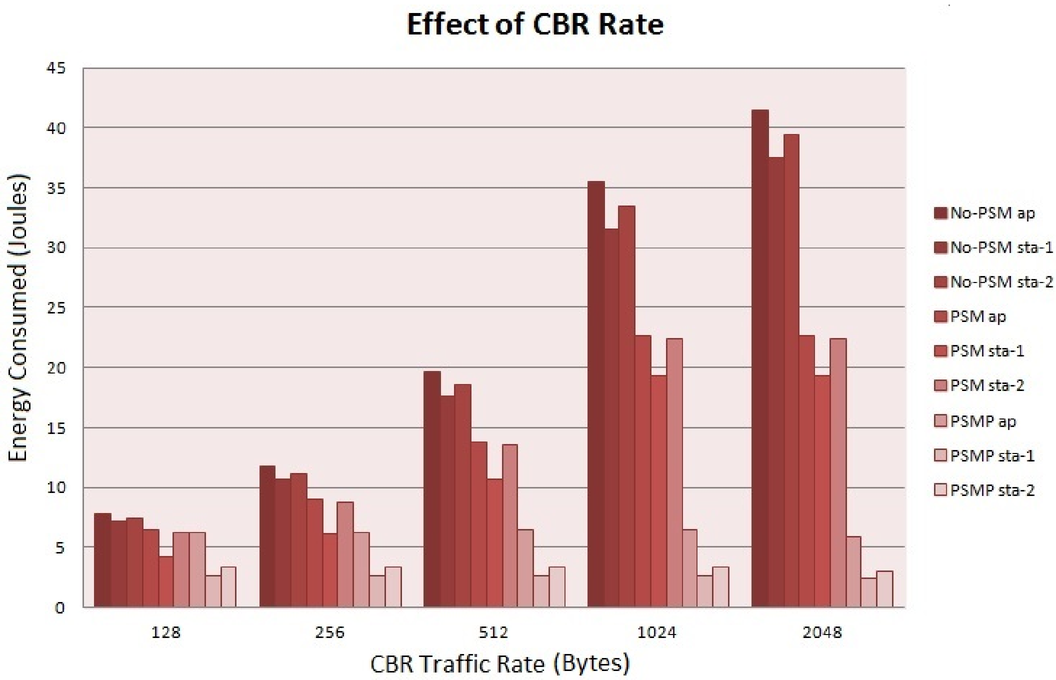


Figure 3.10: Effect of Traffic Generation Rate

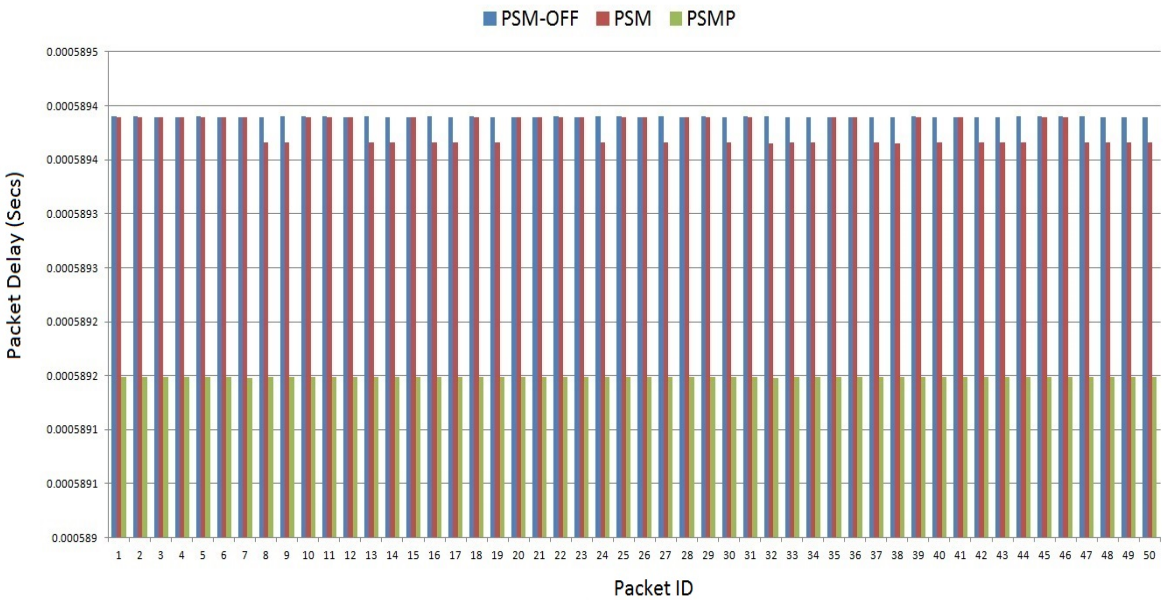


Figure 3.11: Per Packet Delay

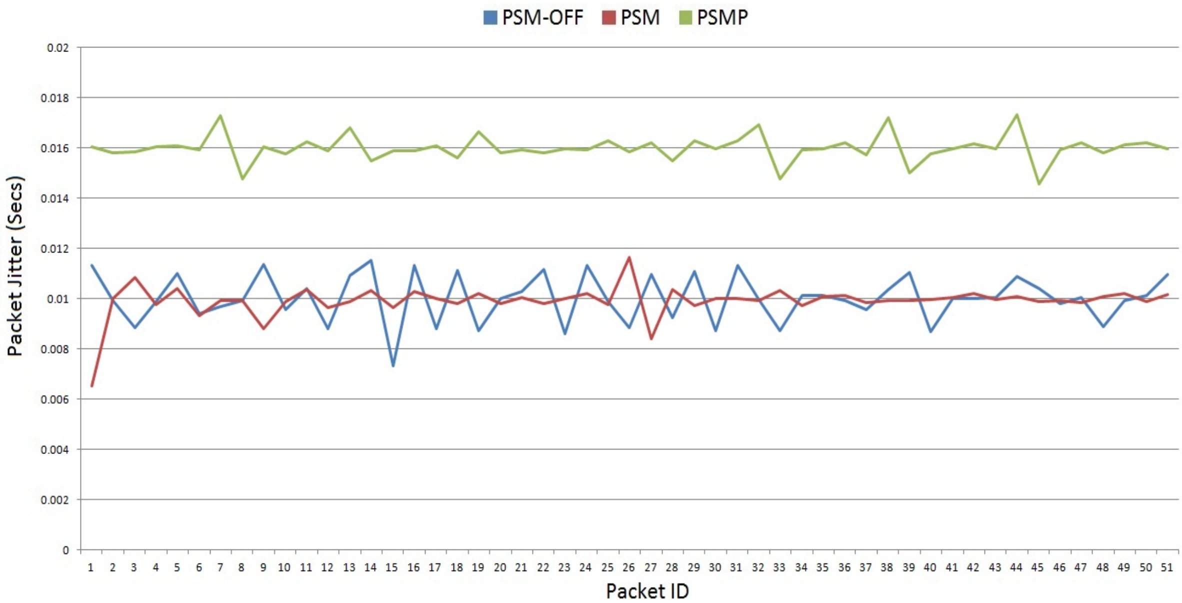


Figure 3.12: Per Packet Jitter

Chapter 4

Green camera node hardware specifications

The camera node has four basic components, a microcontroller, low resolution CMOS camera, a Wi-Fi module and the supply circuit(s). In this section we will discuss the choice of these components and how they will be put together.

4.1 The Microcontroller

.

At the heart of the camera node is the 15 Dollars ARM 32-bit $Cortex^{TM}$-M4-based STM32 F407 MCU. The STM32F407 has a maximum CPU clock of 168MHz, with current consumption at about 180 μA per MHz. The MCU is rated at 210 DMIPS, with up to 1Mbyte of Flash. It has USB OTG HS-FS, 3ADCs, Ethernet MAC and camera interfaces.

4.2 Why Cortex M4?

The camera node is to be a low power, low cost device, capable of handling video frames of CIF resolution 352×288. Therefore higher end processor architectures such as the Cortex A and Cortex R series simply don't fall in $10 range, and with GHz clocks and OS overheads, they don't fall in the low power range when the subject is a node. On the other hand, low to midrange microcontrollers such as ATMEL's ATMEGA8, Microchip's PIC16F and PIC18F series, with maximum clocks of up to 40MHz and 8-bit, 16-bit architectures with no DSP capability, simply can't handle the 30fps frame rates and fairly large matrix multiplications in real time. Therefore we need a high end microcontroller with DSP capabilities i.e. a Digital Signal

Controller (DSC, capable of performing single instruction multiplications). In multimedia processing, inter-frame sparsity needs to be exploited in order to achieve greater compression, therefore neighboring frames (2-3) need to be quickly saved and accessed to and from the memory. When memory operations are large (as in this case) they put a huge burden on the processor core unless the architecture is memory mapped with a dedicated Memory Protection Unit (MPU). Furthermore, our green camera node for IoT involves interfaces such as CMOS camera interface, interface with wiFi module, and a USB port for tasks such as software up-gradation. The Cortex M4 architecture from ARM is undoubtedly the best fit to all these requirements, as it is DSC, memory mapped, multi-stage pipelined and manufacturers such as Texas Instruments, ST Electronics and NXP are manufacturing their own product ranges based on the Cortex M4 ARM architecture. Another important thing about the cortex M4 based microcontrollers is that support material such as development boards, extension boards and a range of open source supporting middleware (such as RTOSs) are easily available at very nominal prices.

4.3 Why STM32F4?

Table 4.1 shows the main features of four Cortex M4 based DSC microcontrollers, two from NXP and one each from ST and TI. They have been placed in the table from heavy to light, from left to right, respectively, in terms of their specifications. NXP's LPC4333 (first one from left) is undoubtedly the heaviest of the four (in fact the most powerful Cortex M4 microcontroller series) with dual core architecture. However its added capability in comparison with the second in line, STM32F407, is only for added price. On the other hand, ST's Discovery board is 10 times cheaper than the NXP's OM13027 board, and 20 times cheaper than the NXP's OM1331 board. And thanks to its very low price the Discovery board is very popular among developers, so a lot of open source material is available for it in comparison with the NXP products.

Now the NXP's LPC4088, at third and TI's TM4C123G, at fourth are clearly lower in performance with only marginally lower prices. Additionally the development board for the LPC4088 (OM13029) is priced at a mighty $277. On top of all this the STM32F407 has a dedicated camera interface, not available on the other boards. So to conclude all this, our choice is the STM32F407, and for the development we will be using the STM32F4 Discovery board together with its expansion boards.

Controller	LPC4333	STM32F407	LPC4088	TM4C123G
Manufacturer	NXP	ST	NXP	TI
Architecture	Cortex M4+M0	Cortex M4	Cortex M4	Cortex M4
Clock (max)	204MHz	168MHz	120MHz	80MHz
Flash (program memory)	1Mbyte	1Mbyte	512Kbyte	256Kbyte
USB	2	2	1	1
Ethernet	1	1	1	1
Camera Interface	No	Yes	No	No
Controller Price (USD)	21	15	12	13
Development Board(s)	OM13027 and OM13031	STM32F4 Discovery	OM13029	TM4C123G LaunchPad
Board Price (USD)	155 and 321	15	277	20

Table 4.1: Specs Comparison

4.3.1 Expansion Boards

Base Board

The Base Board provides easy interfacing with the LCD, camera and WiFi boards from ST as well as other compatible boards. It also provides an Ethernet port and memory card jacket for additional memory.

DISCOVER Camera Board

A digital camera board featuring a 1.3 Megapixel CMOS sensor connects to the base board (STM32F4DIS-BB) to form a complete system.

Key Features

- *Signal system: CMOS 1.3 Megapixel*
- *Resolution: Up to* 1280×1024
- *Supports still photos*

- *Frame rate: 15 fps for SXGA, 30 fps for VGA and CIF*

DISCOVER 3.5 inches LCD Board

A 3.5 inches LCD board with touch screen capability connects to the base board (STM32F4DIS-BB) to form a complete system.

Key Features

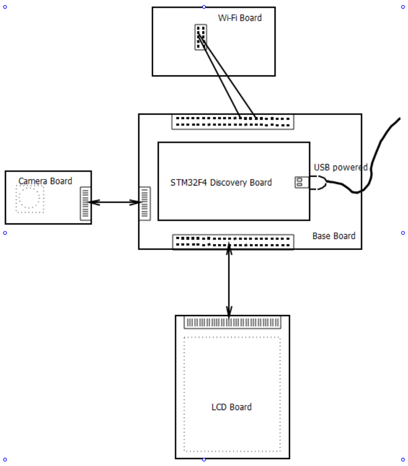
- *Driving IC: SSD2119*
- *Display format: 320 * 240*
- *Color: 262K colors*
- *Backlight: PWM control*
- *Interface: 16-bit 8080 parallel system interface*
- *Touch screen: 4-wire resistive touch screen*

DISCOVER Wi-Fi

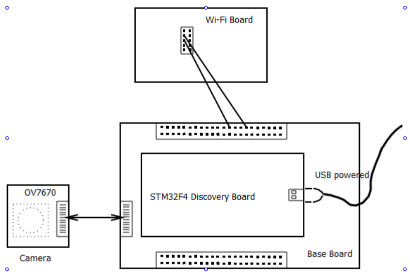
The Discover Wi-Fi (STM32F4DIS-WIFI) features the SN8200 Wi-Fi module from Murata and enables Wi-Fi for your STM32F4DISCOVERY board. Easy entry with built-in TCP/IP stack to quickly connect to your STM32F4DISCOVERY Kit.

Key Features

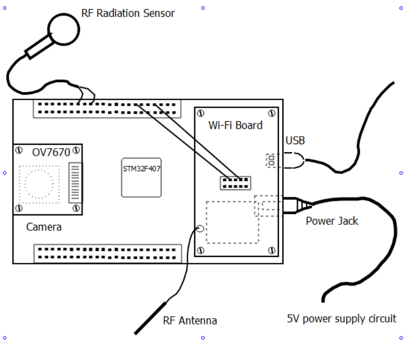
- *AP/STA dual mode*
- *UART, SPI, 2.4GHz IEEE 802.11b/g/n interfaces*
- *Wi-Fi security supplicant supporting WPA-PSK and WPA2-PSK*
- *Built-in TCP/IP stack, DHCP, DNS and HTTP server*
- *On-board STM32 ARM Cortex-M3*
- *Broadcom BCM43362 Wi-Fi chip*
- *+18 dBm transmit power*
- *-96 dBm maximum receiver sensitivity @ 11 Mbps*



(a) Setup for Testing of Hardware



(b) Hardware Setup for Development and Testing of Code



(c) Hardware prototype of green camera node.

The LCD board is not part of the node (see Fig 4.1a and 4.1b). It is just a development aid. The camera board will be replaced by a low cost, low resolution CMOS camera after the initial hardware setup stage. We will discuss the specs of two such cameras in the next section. The WiFi board will become part of the final design of the node, without any changes to its hardware. The Discovery board and the Base Board will remain part of the node throughout the development stage, however a PCB will be designed for the final prototype, replacing these two boards. The Discovery Board is USB powered, so during the basic prototyping we will not need external power supplies. However at a later stage we will make our own PCB design, and will incorporate the power supply system at that stage as illustrated in Fig. 4.1c.

Power supply

For the final prototype the estimated, maximum power requirement of the whole node is as follows;

$$\begin{aligned} P_{total} &= P_{STM32F407} + P_{WiFi} + P_{Cam} \\ &= 200mW + 250mW + 150mW \\ &= 600mW \end{aligned} \tag{4.1}$$

Since the discovery board is USB powered at 5V, we will also provide our node with a stable 5V supply. So we can calculate the current drawn by the node as follows;

$$\begin{aligned} I &= P/V \\ &= 600mW5V \\ &= 120mA \end{aligned} \tag{4.2}$$

So a $5V$, $800mAh$ battery could supply the node for more than 6 hours (with the node in continuous operation), before needing a recharge. This is good enough for experimental purposes. We have already given the high level outline of the design of power source. This design is not innovative so we will use an existing power source which matches both our design and our system requirements. The solar lamps by d.light are a very good match to our requirements and can be easily adapted by removing the led and using the rest as the supply circuit for the green camera node. For instance the WakaWaka Solar Lamp as shown in Fig. **??** has a rechargeable $5V$, $800mAh$

battery, and all the necessary circuitry. The cheaper S2 Solar Task Light might also be good enough.

Carbon Footprints

One thing that we will add to the power source is a simple battery monitor connected to the STM32F407 microcontroller. This will give us drain curves for different batteries in different situations (communication load, amount of ambient light e.tc.). We will use these curves to determine the carbon footprints left by our node in different situations. Another important research direction is to find some empirical relation between the amount of emitted RF radiation, and the carbon footprints left behind by the node. Therefore we will use an RF radiation sensor to monitor the RF radiation in the same conditions, mentioned in previous paragraph. Using the drain curves and the radiation data, we will try to find some empirical relation (if any).

4.3.2 The CMOS camera

As the paradigm of IoT develops, it is evident that 'things' will be predominantly simple, and in large numbers. For this reason and for ease of development we have selected the standard CIF resolution of 288×352. We have to perform compressed sensing inside our controller (in digital domain) so the frame data from the camera (CMOS sensor) needs to be in a raw format (compressed sensing can only be performed on raw data). There are many raw image formats such as, YUV/YCbCr4:2:2 RGB565/555/444 GRB4:2:2 Raw RGB. So we need a camera which gives some of these formats. Another requirement is that the camera interface needs to be compatible with the camera interface on the Base Board. We have selected two cameras, OV7670 and MT9D111 to fit these requirements. The cameras are very similar accept for a few variations in the output video formats. It is not clear at this stage, which format we will need, so we have gone for these two cameras to widen our choice. The salient features of the cameras are listed below.

OV7670 (AL422b)

Photosensitive Array: 640 x 480 IO
Large capacity with 380KB FIFO Buffer IC AL422B, easy to apply with MCU/ARM system via I/O port
24MHZ active crystal
Voltage: 3.3V
Operating Power: 60mW/15fpsVGAYUV
Sleeping Mode: $<$ 20 uA

Figure 4.2: Solar cell based power supply for green camera node (see www.waka-waka.com).

Operating Temperature: -30 to 70 deg C.
Output Format: YUV/YCbCr4:2:2 RGB565/555/444 GRB4:2:2 Raw RGB Data (8 digit)
Lens Size: 1/6 inches, Vision Angle: 25 degree
Max. Frame Rate: 30fps VGA
Sensitivity: 1.3V / (Lux-sec)
Signal to Noise Ratio: 46 dB
Dynamic Range: 52 dB
Browse Mode: By row
Electronic Exposure: 1 to 510 row
Pixel Coverage: 3.6um x 3.6um
Duck Current: 12 mV/s at 6'C

MT9D111
Ultra-low-power, low-cost, progressive scan
Resolution of 2 million pixels (1600 H x 1200 V)
1/3.2 inch optical format
Full resolution frame rate of 15 fps
Real-time JPEG encoder
Integrated micro-controller, increase flexibility
On-chip image flow processor, suitable for single-chip camera module

Numerous automatic functions can be quickly corrected and enhanced image include quick exposure adjustment
10-chip integrated analog-to-digital converter
ITU-R BT.565 (YcbCr), 565RGB, 555RGB, 444RGB raw output data format
JPEG 4:2:2 and 4:2:0 output
The response rate: 1.0 V / lux-sec (550nm)
Master Clock: 6 MHz-80 MHz (Integrated PLL)
Signal-to-noise ratio: greater than 41dB (maximum)
Power supply voltage: 3.1V–5V
IO voltage: 2.8V (+ /-0.5V)
Power consumption: less than 150mW (@ 30 fps)

Chapter 5

Future Roadmap

Power Saving Techniques IoM is an enhancement to the IoT, whose prime objective is to enable resource constraint low-power low-cost heterogeneous multimedia devices to promote multimedia content as part of the realization of IoT. Specially in IoM, the real-time multimedia traffic that is continuous in nature requires stringent communication requirements as compared to data communication such as bandwidth, delay, jitter and reliability. For this reason, IEEE 802.11 standard is suggested in the literature for WMSNs, since it provides a high data rate communication model and holds a great potential for IoM. In this report we performed an analysis of the power saving techniques specified by the IEEE 802.11 standard. We evaluated various key performance metrics and compared the performance of PSM and PSMP with standard network scenario without any power saving mechanism. From the performance analysis we can conclude that PSMP out performs PSM and provides significant energy efficiency gain. However, one important thing is that PSMP as specified in IEEE 802.11n doesn't provide any guidance in case stations access the AP via multi-hop links. As in IoM multiple devices are expected to communicate and there will be some multi-hop links as well, so an extension of PSMP to provide power saving mechanism in multi-hop scenario is important to realize its implication in IoM architecture.

Video Acquisition The immense simplicity of the encoders chosen during this quarter will allow us to port the chosen techniques on the hardware platform in the future quarters while keeping the power consumption to a bare minimum.

Hardware Platform By now we are quite clear about the high level design of the green camera node, and now we are ready to put together the hardware and then go through the repeated hardware-software-hardware-software design cycles.

Bibliography

[1] Ying Liu, Ming Li, and Dimitrios A Pados. Motion-aware decoding of compressed-sensed video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(3):438–444, 2013.

[2] Iain E Richardson. *The H. 264 advanced video compression standard.* John Wiley & Sons, 2011.

[3] Linux news, "http://linuxfr.org/news/h-265-est-finalise", 2013.

[4] Gary J Sullivan and Stephen J Estrop. Methods and systems for start code emulation prevention and data stuffing, 2009. US Patent 7,505,485.

[5] Ntu university, "http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/jpeg/jpeg/enc 2013.

[6] *Motion-Prediction,"http://wiki.multimedia.cx/index.php?title=Motion Prediction".* 2007.

[7] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[8] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[9] Adam Dunkels and Jean-Philippe Vasseur. Ip for smart objects, september 2008. *IPSO Alliance White Paper*, 1.

[10] N Kushalnagar, G Montenegro, C Schumacher, et al. Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals. *RFC4919, August*, 10, 2007.

[11] *IEEE 802.11n-2009 Standard for Information technology, Local and metropolitan area networks, Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*. IEEE.

[12] K. Hall West and G. West. *Wireless Sensor Network Technology Trends Report*. West Technology Research Solutions, LLC, Tech. Rep., 2008.

[13] *Wireless Connectivity Market Data*. ABI Research, 2011.