

COORDINATED DISTRIBUTED POWER MANAGEMENT WITH VIDEO SENSOR NETWORKS: ANALYSIS, SIMULATION, AND PROTOTYPING

Nicholas H. Zamora, Radu Marculescu
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA

ABSTRACT

In this paper, we explore various design issues for Coordinated Distributed Power Management (CDPM) policies in wireless Video Sensor Networks (VSNs). These CDPM policies help to efficiently power manage such networks while benefiting from the advantages gained by using distributed techniques. The design issues we explore include power management under dynamic and adaptive timeout thresholds, two-hop broadcast information dissemination, hybrid CDPM, and remote wakeup. Our investigations use an advanced, event-triggered VSN simulator, as well as a set of VSN prototype nodes we built as a proof-of-concept. Our prototype network includes four digital signal processing (DSP)-based wireless video nodes which form a small multi-hop network. Last but not least, we propose a novel analytical approach to predict the CDPM policy performance, and show that this analytical method matches the measured power savings in the prototype.

1. INTRODUCTION, RELATED WORK AND NOVEL CONTRIBUTION

A wireless Video Sensor Network (VSN) uses low-power imaging cameras, processors, and communications transceivers to survey and monitor an area of interest. VSNs operating on energy-limited battery devices are considerably challenging to implement due to the high data rates and large processing requirements typically associated with real-time video processing. This challenge exacerbates the need for effective and efficient Power Management (PM) policies.

Cost reductions in memory and processing power along with improvements in video capturing devices with CMOS image sensors [1] are making wireless, coordinated, battery-powered video networks affordable. A VSN operating on limited energy resources [2] needs to be power efficient to reduce maintenance costs with fewer battery replacements and allow the system to be alive for years and months instead of days and hours. Energy-efficient design should cut across all levels of abstraction, from circuit level [3], up to software and network levels [4], especially as image sensors (cameras) have higher power consumption rates compared to other sensors.

In our previous work [5], we have proposed novel Coordinated and Distributed Power Management (CDPM) policies for VSNs working at the application level. More specifically, in [5] we proposed two CDPM policies for video

sensor PM which are general and effective policies for various configurations of wireless VSNs. The first technique extends the idea of local timeouts to a set of nodes, while the second technique uses a voting algorithm with each node running an election to decide when and if to shut down. However, our previous work in CDPM policies only describes basic policy algorithms without considering the density of the network, the scalability of the network, nor the possibility of using remote wakeups from other nodes in the wireless network.

This paper extends our previous work on CDPM policies, adding significantly to it. More precisely, the contributions in this paper include:

- A novel two-state FSM analytical model is introduced to predict the performance of CDPM policies. We also show how this model accurately estimates the performance of the CDPM policies by comparing the estimates to physical measurements of our prototype VSN system.
- Adjusting the timeout threshold values based on network density. Making the threshold adaptive in this way helps maintain a fixed sleep fraction for the VSN nodes.
- Scalability of the coordination behavior is explored when we consider forwarding the PM policy information over multi-hop communication paths and using that information from two hops away in the CDPM policy decision making.
- Capturing the best characteristics of both previously proposed CDPM policies is possible by using a hybrid policy. We describe the implementation of this hybrid policy which improves performance beyond what is possible from either coordinated timeouts or a voting policy alone.
- Utilizing remote wakeup capabilities as a predictor to improve performance is also explored. We also show how the improvement in performance is significant, although this comes with a cost.

We note that the CDPM policies we investigate do *not* require location or topology awareness. Consequently, no coordination in terms of global broadcasting, complex routing, tight time synchronization or other overhead is necessary to use these CDPM policies. We also explore exactly how performance of these policies is affected when considering using message forwarding (with two-hop CDPM policies), as well as time synchronization for remote wakeups.

1.1. State of the art

Dynamic PM policies have been previously studied in [6]. Other CDPM policies for sensor networks appear in [7], where

the authors introduce a PM policy while focusing on maintaining the network connectivity and guaranteeing a minimum coverage region. Unfortunately, that work assumes that sensors have circular sensing areas and lie on a flat 2-D plane. These assumptions clearly do not hold for VSNs as considered in this paper, but they may be reasonable for other wireless sensor networks in which the sensor is unidirectional (as opposed to directional image sensors). With dense video coverage, the fields of view (FOVs) of neighboring sensors often overlap. The authors in [8] propose to save energy by partitioning the data to be captured in time and assigning a time schedule for each node to be actively surveying the area. While this technique does not save energy if there are no overlaps within the FOVs, it can work on top of, or as a complement, to our CDPM policies. Our future work will consider FOV correlation and the effects this has on CDPM performance.

In [9], the authors use a hybrid automaton model which considers the underlying sensor application when executing the PM policy. A Markov-based model is used in [10] to accurately predict video node lifetimes when design decisions are changed (e.g. schedule-driven versus trigger-driven design). The authors in [11] allow for a range of power modes with varying degrees of functionality and they effectively use these modes to minimize energy consumption based on prediction models for the environment. Finally, although research on multiple camera systems is beginning to appear [12][13], real-time multiple video streaming over a mesh network will require a push to the next level in terms of wireless bandwidths capacity. Current technology is too limited in bandwidth to allow multiple real-time video streaming over a mesh network, however future wireless protocols such as 802.11n [14] could enable these systems in the next decade.

1.2. Novel contribution

The CDPM policies for VSNs we explore in this paper are unique as compared to policies proposed for other sensor networks and systems for several reasons:

- The CDPM policies explored in this paper consider the content of the video data sensed locally, as well as the video data sensed by other video nodes within the network. In other words, the video containing moving objects sampled by neighboring nodes is considered when each node executes its own PM policy.
- Most current research assumes that power consumption for wireless transmission dominates the power consumption required for processing at the local sensor node. Our CDPM policies require a very low energy overhead, while still benefiting from information gathered from the neighboring nodes. The required communication overhead to utilize these policies is more than made up for by the increased overall energy savings.
- The CDPM policies use a best-effort approach to utilizing all the information available and do not require special hardware such as GPS receivers or additional topology or location-discovery algorithms. Our policies perform well in both hand-crafted, as well as randomly distributed network systems.
- These CDPM PM policies are naturally scalable with the size of the system. This is because our policies only consider the

neighbors one-hop away from the local node. To investigate this issue, we consider how the CDPM policy performance is affected when relaxing this restriction and extending the coordination to two-hop neighbors.

The remainder of this paper is organized as follows: Section 2 briefly describes our node-level driver application for object tracking. In Section 3, we define a CDPM policy and compare distributed to centralized PM policies. Section 4 describes our CDPM policies, prototype nodes, and a novel analytical model for estimating the performance of such policies. Section 5 explores various design parameters and considerations for CDPM policies. Finally, Section 6 summarizes our findings and points out future research.

2. OBJECT TRACKING DRIVER APPLICATION

As a driver application, our video sensor nodes execute motion detection and estimation algorithms to achieve object tracking within the FOV. Within a single video sensor node, we use a sum of absolute difference technique for each macroblock to detect motion. More precisely, we implement:

$$SAD_{m,n,k} = \sum_{i,j \in MB} |pix_{i,j,k} - pix_{i,j,k-1}|$$

where i and j are pixel coordinates in the horizontal and vertical directions, and $pix_{i,j,k}$ is the pixel value at location i,j in frame $\#k$. If this $SAD_{m,n,k}$ value exceeds a threshold, then we mark macroblock m,n as containing movement in frame k . Once we have computed the SAD of each macroblock, we implement the mean-shift algorithm [15] to identify objects and estimate their motion.

Following the motion detection and estimation steps, the video sensor node schedules a broadcast network message containing information about movement detected and estimated. As described next, the CDPM policies use the results of the motion detection and estimation from neighboring nodes to more effectively make power management decisions for each VSN node locally.

3. DISTRIBUTED POWER MANAGEMENT POLICIES

For the purposes of this paper, a *distributed PM* policy refers to any policy in which there is *no* single (centralized) entity executing the policy and making PM decisions for multiple target video sensor nodes. This definition is consistent with a distributed system as described in [16]. Distributed PM policies enjoy many advantages as compared to centralized policies:

- Distributed systems generally do not suffer single point of failure vulnerabilities which is characteristic of centralized control. The benefit of this is that any single failing node in a distributed power managed system will only adversely affect itself (in the worst case, the local nodes nearby could be affected as well if the node's failure mode is not silent and the severity is critical). A failing centralized controller, on the other hand, could fail by allowing the entire system to remain fully active at all times, draining battery power inefficiently. Worse, the controller could shut down the system permanently by powering down the entire system.

- A distributed policy can quickly react to local events such as a burst of motion detected in the nearby vicinity or a low battery condition. Centralized policies will generally need to wait for such events to be relayed (possibly over a congested network) to the centralized controller, queued in a process list, processed, and the response returned to the node of interest before a course of action is taken. This additional delay could be detrimental to the underlying video processing algorithms.
- A distributed policy is more scalable as the number of nodes or devices in the system grows. With a centralized controller, scalability is very limited so careful consideration must be taken to ensure the size of the system is manageable and additional controllers might be needed as the system grows.
- Arguably most importantly, distributed policies often result in simpler hardware or software design, can be more easily maintainable and debug-able for the system designer.

Unfortunately, distributed policies do have their own disadvantages as well:

- A distributed policy running on a local node does not have the same global view of the system as would a centralized controller. This drawback can make it difficult to execute any policy in which coordination or synchronization with all other network nodes is required. However, if the system is static (i.e. nodes do not enter or leave the system during its lifetime) then a static policy could be established off-line.
- A node executing a distributed policy might not have sufficient memory and/or processing resources available for running an advanced PM policy. This is a real issue with sensor networks other than VSNs but less of an issue with VSNs because the node already presumably has enough resources to process the video in real-time.

While centralized power management systems are well understood, designing, implementing, and analyzing CDPM policies for video sensor networks are unsolved and barely explored research issues. Next, we give a brief overview of our baseline CDPM policies and describe a much-needed analytical model for predicting CDPM policy performance.

4. CDPM POLICY IMPLEMENTATION AND ANALYSIS

In this section, we describe the CDPM policies used in our experiments with specific details on the prototype node implementation.

4.1. Prototype VSN network

We have constructed 4 prototype VSN nodes (Figure 1), based on a Texas Instruments mid-range 200MHz DSP [17]. Each node contains a CMOS video sensor capturing frames at 30 fps and a ZigBee IEEE 802.15.4 wireless transceiver to form the multi-hop network. The topologies we constructed were complete (i.e. each node can hear all other nodes), as well as a single path (i.e. node n can hear nodes $n-1$ and $n+1$ only). The nodes implement the object tracking driver application as described in Section 2 and the CDPM policies as described next. They use a CSMA-CA MAC for network broadcasts with random back-off times following a packet collision. The

transceivers have a maximum data throughput of 250kbps. Overall, our prototype node has an active power consumption value of about 400mW and low-power consumption of about 10mW (see Figure 2), and so these nodes are low power video sensors as opposed to high-performance, high-power surveillance cameras.



Figure 1 – Three of the prototype VSN nodes. These nodes form a multi-hop wireless mesh network implementing the CDPM policies investigated in this work and the object tracking driver application as described in Section 2.

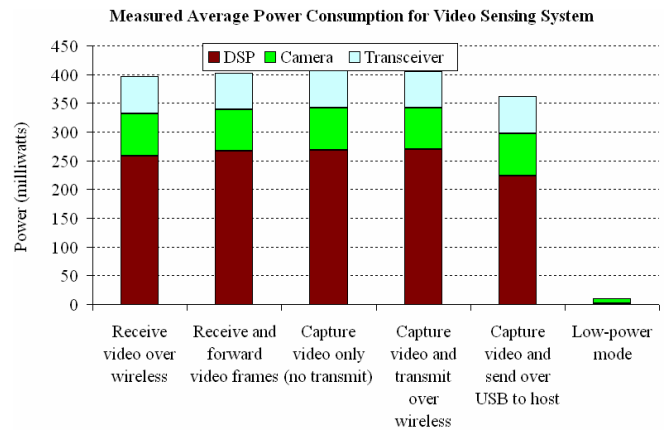


Figure 2 – Power consumption values for our prototype node in various active modes and the low-power mode.

4.2. Coordinated timeout and voting policies

We allow each video sensor node to make its own policy decisions (hence the *distributed* nature of the CDPM policies), while at the same time each node considers real-time video data information from its neighboring nodes (hence *coordinated*). The first baseline PM policy we consider is called “local timeout with predictive wakeup”. This policy is shown graphically in Figure 3, and it is most easily understood as shown here when there is no movement within a camera’s FOV.

The local timeout with predictive wakeup policy works as follows. Each video sensor begins in fully active mode (point A of Figure 3) and resets a timer which keeps track of how long no motion has been detected in front of camera’s FOV. If the value of that timer exceeds a threshold value, then the camera is put into a low-power mode for a predetermined length of time. Once the sleep time expires (point B of Figure 3), the node returns to the active mode again and the process repeats.

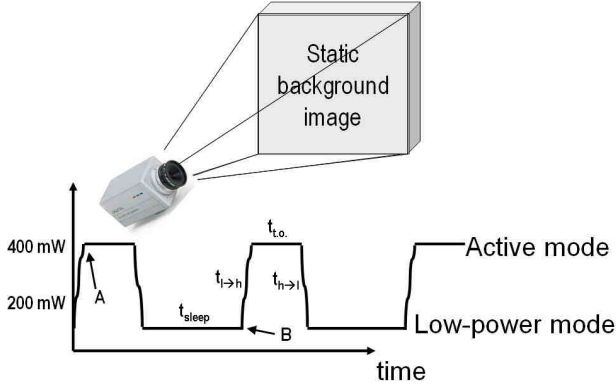


Figure 3 – Power consumption rates for active and low-power modes for a video sensor node with no motion detected within its FOV. Note the periodicity in time and the asymmetry of the active/low-power duty cycle.

The graph shown in Figure 3 represents the maximum sleep time fraction achievable, which for the well-known baseline policy is:

$$t_{sleep\ fraction,max} = \frac{t_{sleep}}{t_{sleep} + t_{t.o.} + t_{l \rightarrow h} + t_{h \rightarrow l}} \quad (1)$$

where t_{sleep} is the fixed time the node is in the low-power mode, $t_{t.o.}$ is the no motion timeout threshold value, and $t_{l \rightarrow h}$, $t_{h \rightarrow l}$ are the times for the low-power to active mode and active to low-power mode transitions, respectively. In this paper, we can safely ignore these transition times since the prototype’s transition times we have measured are far less than 1msec, while t_{sleep} and $t_{t.o.}$ have a minimum value of the time for 1 video frame, which at 30 fps in our prototypes is at least 33msec. In our experimental setup, we set t_{sleep} and $t_{t.o.}$

to be 30 and 15 seconds, respectively, and so we can see that the transition energy consumption values do not significantly contribute in equation (1)¹.

The coordinated version of this well-known baseline policy simply extends the decision to include the timeout values of the immediate neighbors in the network. Once a timeout occurs within a node, it piggybacks a “timeout” message onto its other regularly scheduled network messages for wireless broadcast. The receiving nodes strip this timeout message from the packet, decrement the time-to-live (TTL) by one, and forward the packet on to subsequent nodes if necessary. In this way, nodes will be aware of the timeout status of their neighbors, and can enter into low-power mode if it and each of its neighbors are simultaneously in a timeout state. The reader is referred to [5] for more details on this policy.

Alternatively to a coordinated timeout PM policy, the nodes could execute what we call a *voting policy* [5]. The voting policy works as follows. Instead of maintaining a timer, each node broadcasts periodically a summary of motion

¹ If a PM policy uses a much shorter sampling period such that these low-to-high and high-to-low times are significant, then the model in the next section needs to be modified to include an additional state for each transition changing power modes.

detection information. Each node collects this summary from its immediate neighbors, and uses the summary as a vote in a local election. If enough of the neighbors vote that there is no movement being detected, then the node can enter the low-power state. In our implementation, we use a constant function of 1 dissenting vote per every 8 neighbors. For example, with 20 neighbors, if more than $\lfloor \frac{20}{8} \rfloor = 2$ neighbors are detecting

movement, the node will not enter into the low-power mode and rather continue to stay in active mode. This dissenting vote constant can be increased for a more aggressive CDPM policy or reduced for a more conservative policy.

4.3. An FSM to capture burst motion

To analyze the CDPM policies as described above, we use a two-state FSM model as shown in Figure 4. This model is based on the well-known Gilbert-Elliott [18][19] model for communication channel burst bit errors.

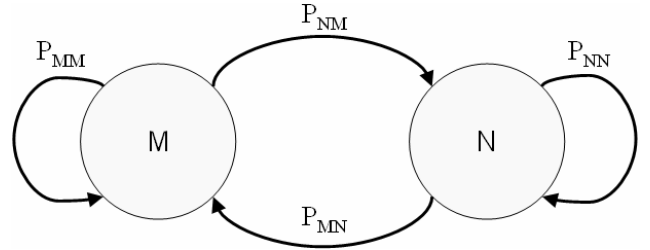


Figure 4 – A two-state FSM for modeling movement within a camera’s FOV. State M represents motion detected in the current frame and N represents no motion.

Each video node in the network executes an instance of this two-state FSM. For a single video node, as each video frame arrives and is analyzed, the node determines whether or not that frame contains significant motion. Frames with motion detected are represented in state “M”, while frames with no motion are represented by state “N”. The key component to modeling “burst” motion is to choose appropriate values for the transition probabilities. Video is typically highly correlated in time, and typically, motion in the current frame implies that motion in the next frame is highly likely (see Figure 5). Choosing a value for the self transition P_{MM} greater than 0.5 will capture this correlation, and values closer to 1 will model longer time correlations. Similarly, values for the self-loop P_{NN} close to 1 model “bursts” of no motion.

Although simple in nature, the model in Figure 4 can accurately predict performance of CDPM policies. We ran the coordinated timeout policy on each of the network nodes and looked at the predicted sleep fractions. The probabilities we set based on analysis of real video clips. We also varied P_{MM} and P_{NN} , independently, from 0.92 up to 0.995 and the results are shown in Figure 7(right). We also implemented the coordinated timeout policy on our prototype video sensor nodes (FOVs are shown in Figure 6) and the actual sleep fractions are shown in Figure 7(left).

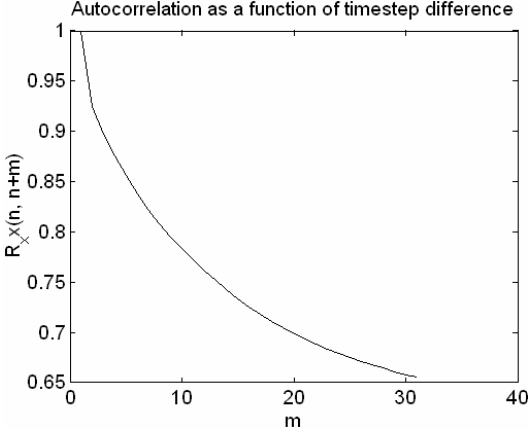


Figure 5 – Simulation results showing likelihood of movement m video frames in the future if the current video frame contains movement. The frame capture rate here is 10fps, so $m=30$ corresponds to 3 seconds in the future. Movement in this experiment was detected just under 3% of the total time, so correlation this high in this figure is statistically significant. Experiments on real video clips show similar time correlations.

Each point in Figure 7(left) is the result of multiple hours of executing the coordinated timeout policy on our prototype nodes during various times of the days and days of the week. At some times (e.g. during the night), less movement is detected and so sleep fractions are increased, and other times (e.g. during mid-day on a weekday), there is a lot of activity and so sleep fractions are reduced. Our parking lot experiments contained movement ranging from 4% to 32% of the total time.

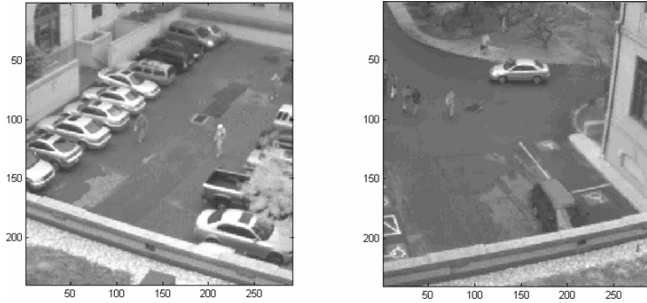


Figure 6 – Two FOVs experimental setup for our prototype video sensor nodes. These two views show different angles of the same parking lot.

Further improvements of the model in Figure 4 can increase the accuracy of the sleep fraction prediction. For example, the model can be enhanced by adding additional states which capture the movement within the previous 2 frames.² For instance, an enhancement using 4 states is shown in Figure 8(left) and the improved prediction is shown in Figure 8(right).

² In general, 2^k states are required to maintain memory for the previous k frames.

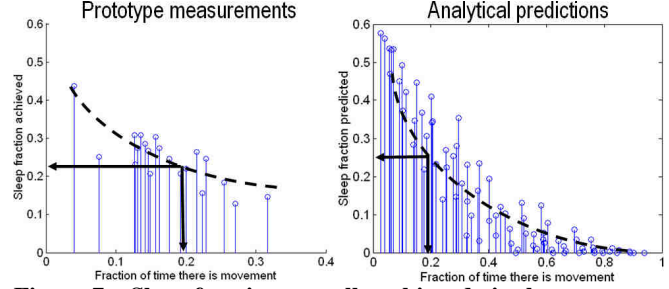


Figure 7 – Sleep fraction actually achieved via the prototype measurements vs. fraction of time movement was detected (left) and analytical predictions for the sleep fraction based on the two-state FSM model (right). Note that with best-fit curves the analytical predictions closely match the sleep fraction actually achieved in the prototype experiments.

As an example, at 20% movement, shown in Figure 7 with the black arrows, the 2 state FSM model predicts a 0.24 sleep fraction when the actual sleep fraction is measured to be 0.22; this represents about 9% error. The 4 state FSM model in Figure 8 more accurately predicts a sleep fraction of 0.23 (or less than 5% error), so the enhancement in fact improves the prediction. Future work will consider expanding the model even further and balancing out the model complexity against the actual improvement in the accuracy of predictions.

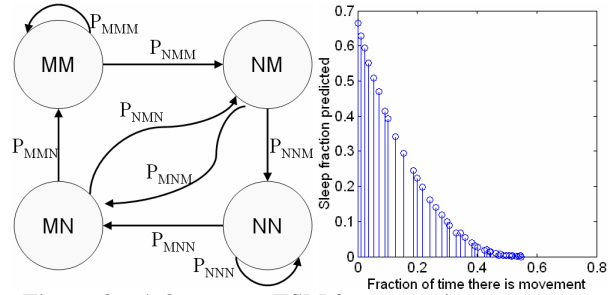


Figure 8 – A four state FSM for capturing movement information from 2 previous frames. The transitions XYZ represent movement (M) or no movement (N) in frames $n, n-1, n-2$ respectively (left). This enhanced model results in improved predicted sleep fractions when compared to prototype experimental results as shown in Figure 7(left).

5. DESIGN CONSIDERATIONS AND EXPERIMENTAL RESULTS

In this section, we consider various improvements and extensions to the coordinated and voting policies described previously. The results in this section are obtained via a simulation framework we have developed specifically for this project. Our simulator uses measured power (i.e. active and low-power consumption values), processing (i.e. MIPS), and communication (i.e. network link latency, jitter, etc.) parameters taken from our prototype system and can extend our VSN to hundreds of nodes. Each experimental point is based on average performance of hundreds of randomly constructed network topologies; more details on the video sensor network simulator can be found in [5].

5.1. An adaptive timeout threshold

Instead of using a fixed timeout value, this threshold can be adjusted dynamically at run-time. With a dynamic network where nodes are constantly joining and leaving the network, the number of neighbors each node will have is constantly changing. If the coordinated timeout policy does not adjust the timeout threshold value, then the policy becomes more conservative every time a node joins the network (because with more neighbors it becomes more difficult for each neighbor to timeout simultaneously), whereas or more aggressive when any node leaves the network. If the goal is to maintain a low-power target (i.e. the design goal is to be in low-power mode a fixed fraction of time independently of the number of neighbors), then the adaptive timeout value should be:

$$t_{ad.t.o.} = \frac{t_{t.o.}}{1 + \sum_{i=0}^{C-1} 1 - \alpha_i, \{0,1,2,\dots,i-1\}} \quad (2)$$

where C is the number of camera neighbors and $\alpha_{i,j}$ is the correlation, ranging from 0 to 1, between the overlapping FOVs for nodes i and j (0 correlation being FOVs which do not overlap at all and a correlation of 1 being FOVs which are identical). If there are no overlapping FOVs (e.g. in a sparsely populated area), then equation (2) reduces to:

$$t_{ad.t.o.} = \frac{t_{t.o.}}{C+1}. \quad (3)$$

Our implementation uses $\alpha_{i,j} = 0$ in equation (2), no correlation for simplicity, however our future work will consider non-zero correlations among neighbors. We implemented this adaptive timeout technique in our simulation framework and the resulting sleep times for various camera densities are shown in Figure 9.

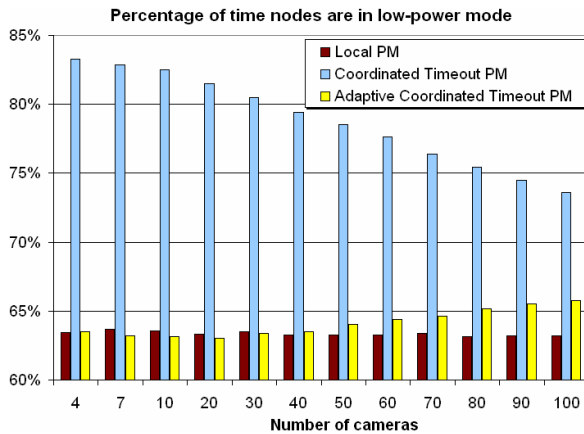


Figure 9 – An adaptive timeout threshold compared to a static value. Note the relative flatness (or independence on camera density) achieved when using an adaptive threshold.

As can be seen, the adaptive coordinated timeout technique does make the achievable low-power mode percentage more independent of the camera density (and network topology),

therefore equation (3) is verified as achieving the predictability in sleep fraction we expect. There is some variability (from 63.5% to 65.5% in Figure 9) which comes from simplifying equation (2) to equation (3). This 2% variability is much better than the >10% variability which occurs when adaptive timeout thresholds are not used. The baseline local timeout PM policy is also included in Figure 9 and shows the variability due to the limited number of experiments (the local PM policy should be completely flat, and any variability is random noise). Because the noise is relatively small, we can have confidence that the results for the other policies are quite accurate.

5.2. Two-hop policy considerations

In our coordinated policies, nodes consider video data from the nearby neighbors. We also increased the coordination to neighbors located two hops away from the current node. This requires that the motion summary information is not stripped away when forwarding a packet for its second broadcast, but leaving that summary in to reach the 2nd hop neighbors. Shown in Figure 10 are results of using the two-hop technique.

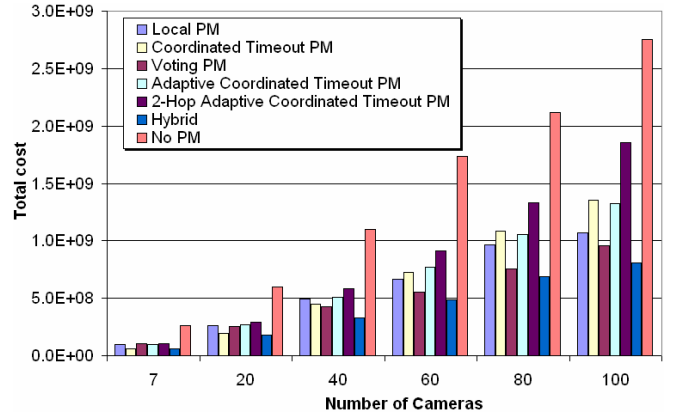


Figure 10 – Relative costs of the various CDPM policies. The hybrid policy achieves the lowest relative cost, and the worst performer is the no PM policy, which is the policy which simply keeps each node in the active mode at all times. See equation (1) in [5] for the cost function used.

The cost of the two-hop policy as shown in Figure 10 is higher overall compared to the other CDPM policies. The cost metric combines amount of movement which is *not* detected by the camera along with inefficiency of the policy (i.e. the amount of time the node could be in a low power mode without missing movement); the cost equation is detailed in [5]. The cost of the two-hop policy is higher because using movement information from network nodes more than one-hop away in the PM decisions results in less accurate decisions being made overall. This is intuitive because for nodes more than one-hop away the video data it sees is less relevant to the local node executing the PM policy.

5.3. Hybrid policy

According to the relative costs of various PM policies as shown in Figure 10, the coordinated timeout PM policy has the lowest

cost for low camera densities, while the voting PM policy has the lowest cost for large camera densities. The point at which the voting policy overtakes the coordinated timeout policy with a lower cost occurs when each node has between two and three network neighbors. Therefore, we define a new hybrid policy where each node considers how many neighbors it has. As such, it uses the voting policy if it has three or more neighbors and the coordinated timeout policy otherwise. We call this the *hybrid CDPM policy*, and the resulting cost is lower than either the voting or the coordinated timeout policy individually. According to our experiments, this is the policy of choice to balance low power and high performance for the underlying video processing algorithm.

5.4. Remote wakeups

Instead of using a fixed sleep time, the neighbors can help determine when to transition back to active mode again. Remote wakeups can be implemented in a couple ways. One possibility is to only power down the other components in the system while keeping the wireless transceiver in active receive mode at all times. However, doing this in our prototype would mean increasing the low-power mode power consumption from about 10mW to more than 70mW (see Figure 2). Clearly, this is not the best option. Another way to implement remote wakeups is to periodically wake the node up and listen to the network, then based on what is received either return to the low power mode or return to active mode. We implement this form of remote wakeup, namely, each node wakes up every 100msec. and listens to the network for 10msec. For the voting policy, the nodes consider messages they receive as votes in a new election and re-run the election at the end of the 10msec period. For the coordinated timeout policy, the nodes consider each message and run the policy again at the end of the 10msec. The other major change for this implementation is that each node must now periodically broadcast its vote or its timeout information every 10msec. This helps ensure that each node, upon waking up every 100msec for 10msec, should receive a network message from each neighbor once (however, collision avoidance could prevent each node from being heard in each 10msec time interval).

Shown in Figure 11 is the average fraction of time each camera in the system is in the low-power and active mode states when adding remote wakeup capabilities to the simulated network of video sensors. With remote wakeups, cameras are in low-power mode less often, 34.3% of the time as compared to 54.6% in low-power mode when not utilizing remote wakeups. Also with remote wakeups, the cameras are more often inefficiently in active mode, but at the same time are detecting more of the movement within their FOVs. In other words, the hybrid policy with remote wakeup causes cameras to burn more power (a decrease in performance) and at the same time detect more of the movement (an increase in performance). Due to this simultaneous increase and decrease in performance, it is not immediately clear if remote wakeups help or hurt overall system performance. However, upon further examination we will now show that the remote wakeup technique hurts the overall performance for our particular experimental setup.

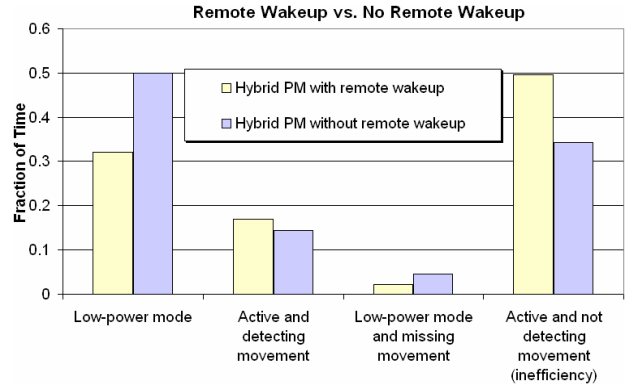


Figure 11 – Steady-state probabilities for a node in various modes of operation. Remote wakeups keeps a node in the low-power mode 34% of the time, as opposed to 54% with no remote wakeups, but the amount of movement missed is improved by more than 50%.

Figure 12 shows the performance space of a CDPM policy in terms of accuracy (i.e. inefficiency of the policy versus the amount of movement being detected). The oracle, of course, can detect all the movement with no inefficiency and this is represented by the lower right-hand-side of Figure 12. The top right of the graph is the “No PM” policy which guarantees all movement is detected, but at the same time the inefficiency is as bad as is possible. At the opposite extreme is the “Always in Low-Power Mode” policy at the bottom left side of the graph.

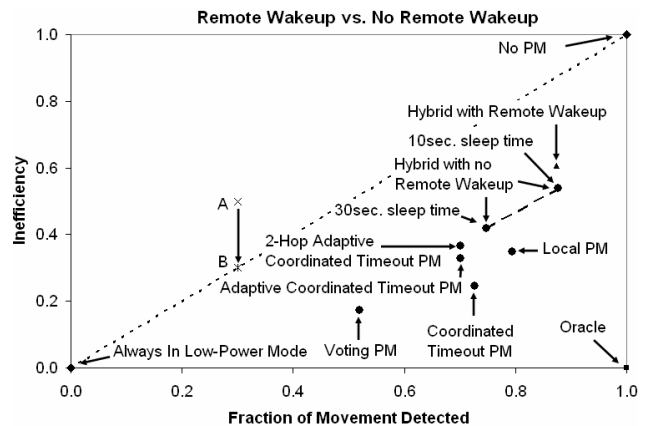


Figure 12 – Effectiveness of various CDPM policies. Adding remote wakeups increases the fraction of movement detected, however reducing the timeout threshold value will result in a policy which dominates in terms of capturing more of the movement and experiencing less time in active mode when no movement is detected (inefficiency).

All the CDPM policies studied in this work are plotted on this accuracy graph so that the policies can be compared directly in terms of performance. Any point in this performance space is dominated by any other point which lies strictly below and to the right of it, and similarly dominates any point which lies strictly above and to the left of it. Note that the local PM policy dominates the 2-hop adaptive timeout policy, and that the coordinated timeout slightly dominates both the adaptive

timeout policies (although one gains predictability in power consumption with the adaptive policies).

The fact that all the policies lie below the dotted line which connects the bottom left and top right of the graph in Figure 12 is a sanity check. Any policy which results in performance above this dotted line is a bad policy because it is dominated by a policy on the dotted line directly below it, such as is shown with example points A and B of Figure 12. Point B lies at coordinates (0.3, 0.3) and is a PM policy which randomly puts the cameras into low-power mode 30% of the time.

To analyze the performance of remote wakeup capabilities, we took the hybrid PM policy without remote wakeups and decreased the sleep time parameter t_{sleep} from 30 seconds to 10 seconds. Comparing this point to the hybrid policy with remote wakeups, you can see from Figure 12 that there is more movement detected while the inefficiency is reduced (i.e. this point dominates in performance compared to using remote wakeups). In summary, using remote wakeups can increase movement detection, but suffers in performance compared to using smaller sleep times for the hybrid policy PM technique.

6. CONCLUSION

This paper proposes several additional CDPM techniques including: adaptive timeout thresholds, scaling the CDPM policies by using two-hop network information, a hybrid policy which chooses the CDPM policy based on network density, and remote wakeup capabilities. We have shown that adaptive timeouts can result in a more predictable sleep fraction independently of the wireless network topology. Further, we concluded that considering data from nodes more than one-hop away results in decreased performance for CDPM policies. Hybrid policies increase the performance of the CDPM technique because each node can independently choose a different policy to implement simultaneously. Finally, remote wakeups can help increase the amount of movement detected, but a more effective method to increasing application performance is to reduce the fixed sleep time parameter and use a more detailed time granularity for making power management decisions.

In addition to exploring CDPM design issues, this paper presented a novel analytical technique for analyzing Coordinated Distributed Power Management (CDPM) policies for wireless Video Sensor Networks (VSNs). This analysis is based on a two-state FSM model. Our experiments show that the predicted performance for estimated sleep fractions closely matches that which is actually achieved using our prototype system.

Future work will explore an analytical model which can capture policy accuracy, real-time deadlines, and FOV overlaps and the effects this has on PM policies.

ACKNOWLEDGMENTS

The above work was sponsored by a fellowship for the first author from the Semiconductor Research Corporation (SRC).

7. REFERENCES

- [1] E. Fossum, "Active Pixel Sensors: Are CCDs Dinosaurs?" Proc. SPIE Vol. 1900, July 1993.
- [2] D. Culler, et. al., "Overview of Sensor Networks," IEEE Computer, Vol. 37, Aug. 2004.
- [3] W. Bachmann and S. Huss, "Efficient Algorithms for Multilevel Power Estimation of VLSI Circuits," IEEE Trans. on VLSI, Vol. 12, March 2004.
- [4] J. Chang and L. Tassiulas, "Maximum Lifetime Routing in Wireless Sensor Networks," IEEE/ACM Trans. on Networking, Vol. 12, Aug. 2004.
- [5] N.H. Zamora, J.-C. Kao, and R. Marculescu, "Distributed Power-Management Techniques for Wireless Network Video Systems," in Proc. Design Automation and Test in Europe (DATE), April 2007.
- [6] L. Benini, et. al., "A Survey of Design Techniques for System Level Dynamic Power Management," IEEE Trans. on VLSI, Vol. 8, June 2000.
- [7] G. Xing, et. al., "Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks," ACM Trans. on Sensor Networks, Vol. 1, Aug. 2005.
- [8] H. Ma and Y. Liu, "Correlation Based Video Processing in Video Sensor Networks," IEEE Intl. Conf on Wireless Networks, Communications and Mobile Computing, Vol. 2, June 2005.
- [9] R. Passos, et. al., "Dynamic Power Management in Wireless Sensor Networks: An Application-Driven Approach," Wireless On-demand Network Systems and Services, Jan. 2005.
- [10] D. Jung, T. Teixeira, A. Barton-Sweeney and A. Savvides, "Model Based Design Exploration of Wireless Sensor Node Lifetimes," in Proc. European Conference on Wireless Sensor Networks (EWSN), Jan. 2007.
- [11] A. Sinha and A. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks," IEEE Design and Test of Computers, Vol. 18, April 2001.
- [12] A. Senior, et. al., "Distributed Active Multicamera Networks," Ambient Intelligence: A Novel Paradigm, pp. 89-105, Springer Science+Business Media, Inc., 2005, New York, NY.
- [13] T. Ellis, et. al., "A Distributed Multicamera Surveillance System," Ambient Intelligence: A Novel Paradigm, pp. 107-138, Springer Science+Business Media, Inc., 2005, New York, NY.
- [14] 802.11n Task Group n Draft 2.0 approved, March 2007, http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm
- [15] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, May 2002.
- [16] W. Emmerich, Engineering Distributed Objects, John Wiley and Sons, New York, NY, 2000.
- [17] Texas Instruments TMS320VC5510 DSP, documentation at <http://focus.ti.com/docs/prod/folders/print/tms320vc5510a.html> as of June, 2007.
- [18] E.N. Gilbert, "Capacity of a Burst-Noise Channel," Bell Syst. Tech. J., 39:1253-1265, Sep. 1960.
- [19] E.O. Elliot, "Estimates of Error Rates for Codes on Burst-Noise Channels," Bell Syst. Tech. J., 42:1977-1997, Sep. 1963.