Real-Time Video Surveillance over IEEE 802.11 Mesh Networks

Arvind Kandhalu, Anthony Rowe, Ragunathan (Raj) Rajkumar
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, USA-15213
{arvindkr, agr, raj}@ece.cmu.edu

Chingchun Huang, Chao-Chun Yeh
Industrial Technology Research Institute (ITRI)
Hsinchu, Taiwan
{chingchun.huang, avainyeh}@itri.org.tw

Abstract

In recent years, there has been an increase in video surveillance systems in public and private environments due to a heightened sense of security. The next generation of surveillance systems will be able to annotate video and locally coordinate the tracking of objects while multiplexing hundreds of video streams in real-time. In this paper, we present OmniEye*, a wireless distributed real-time surveillance system composed of wireless smart cameras.* OmniEye *is comprised of custom-designed smart camera nodes called* DSPcams *that communicate using an IEEE 802.11 mesh network. These cameras provide wide-area coverage and local processing with the ability to direct a sparse number of high-resolution pan, tilt and zoom (PTZ) cameras that can home onto targets of interest. Each* DSPcam *performs local processing to help classify events and pro-actively draw an operator's attention when necessary.*

In video-streaming applications, maintaining high network utilization is required in order to maximize image quality as well as the number of cameras. Our experiments show that by using the standard 802.11 DCF MAC protocol for communication, the system does not scale beyond 5-6 cameras while each camera is streaming at 1 Mbps. Also, we see high levels of jitter in video transmissions. This performance degrades further for multi-hop scenarios due to the presence of hidden nodes. In order to improve the system's scalability and reliability, we propose a Time-Synchronized Application-level MAC protocol (TSAM) capable of operating on top of existing 802.11 protocols using commodity off-the-shelf hardware. Through analysis and experimental validation, we show how TSAM is able to improve throughput and provide bounded delay. Unlike traditional CSMA-based systems, TSAM gracefully degrades in a fair manner so that existing streams can still deliver data.

1. Introduction

Video surveillance has been evolving significantly over the years and is becoming a vital tool for many organizations for safety and security applications. Initially, it was dominated by analog cameras connected using coax cables. For cost and performance reasons, there was a switch to digital switching systems and now IP-based delivery of data.

Video cameras are now being installed at an unprecedented pace in applications that require the coverage of large areas. In order for these systems to be effective, the cost and difficulty of deployment must be reduced. Though frequently discussed, there has been little success in terms of adding advanced machine interpretation of video. Continuous watching of multiple video streams by human operators and manual browsing of thousands of video frames for crime scene and forensic analysis are neither reliable nor scalable. This has brought about the need for a collaborative effort from the systems and vision research communities to develop a surveillance system that is low-cost, reliable, easy-to-manage, easy-to-deploy and can process video data for automated real-time alerts and effective retrieval of archived footage.

In this paper, we describe *OmniEye*, an intelligent surveillance system. *OmniEye* is comprised of custom low-cost smart camera units called DSPcam that are deployed in conjunction with high-resolution PTZ cameras. The DSPcam has an 802.11 radio that communicates over multiple hops to form a mesh network. The traffic generated on this network is primarily directed to a single point in the network, namely the operator's observation station. Since each camera has local processing capabilities, it can detect the event unfolding in its field of view and appropriately annotate the video for the operator. For example, if *OmniEye* detects a human walking, it can label the video data with a tag to represent human motion. These tags can not only draw attention to the situation on the operator's screen, but also be used for conducting rapid *ex post facto* searches. Searches can now include high-level requests for images of one particular object as it moves across multiple camera views. Upon detection of a suspect target, a local DSPcam using the *TSAM* protocol can request higher bandwidth to transmit higher-quality video data. This context-aware classification of data allows more bandwidth to be allocated for cameras that capture information with higher importance.

With existing wireless IP camera systems, as the number of hops to the base station increases, the throughput of the network rapidly decreases. We find that the throughput sharply decreases once the channel has reached 45% of its capacity. Once the network is saturated, the system experiences

starvation, preventing some cameras from transmitting any data at all. To avoid excessive and costly over-provisioning of the system, the bandwidth of the video streams needs to be dynamically managed. Once real-time control of actuators is introduced into the system, jitter and latency become even more of a problem. Packets used to control the PTZ camera or high-priority alert messages to the base station could be lost due to collisions or arrive too late to be of use to the operator.

We propose a Time-Synchronized Application-level MAC protocol (TSAM) that operates over an existing 802.11 MAC protocol. The goals of TSAM are to provide a real-time, scalable communication infrastructure for video streaming on commodity hardware. TSAM eliminates contention between nodes by disabling 802.11 collision back-off and allocating exclusive communication slots. It is therefore able to provide bounded end-to-end delay delay across multiple hops and collision-free operation. An added benefit of the time synchronization comes in the form of temporal registration of images which simplifies coordination among the camera-nodes.

The orgranization of the rest of the paper is as follows. We discuss related work in Section II. In Section III, we describe the architecture of our OmniEye surveillance system and also describe DSPcam in detail. In Section IV, the contention-based approach to medium access used by IEEE 802.11 is described and its performance is analyzed. Following this, we describe our TSAM protocol, present the steps involved, and the design decisions required at each step to implement the protocol as a layer above IEEE 802.11 in a real-world wireless ad hoc domain to mitigate the characteristics of a contention-based MAC. In Section V, an experimental evaluation of TSAM is presented showing that bounded latency and contention-free communication over the existing IEEE 802.11 DCF improve both reliability and scalability. We then go on to describe certain image processing and computer vision algorithms used in the surveillance system. Finally, in Section VII, we provide our concluding remarks.

2. Related Work

He et. al. [1] outline the real-time constraints in surveillance systems. Surveillance applications need to meet certain real-time constraints in response to transient events, such as fast-moving targets. Their work presents the real-time design and analysis of *VigilNet*, a large-scale sensor network system for surveillance. The authors show that while the real-time performance is a major concern in these applications, it should be compatible with other important system properties such as the accuracy of detection and classification of the targets. Loyall et. al. [2] claim that there is a growing class of distributed real-time embedded (DRE) applications that have characteristics and challenges beyond those of traditional embedded systems. They present a represententative DRE application which is surveillance, detection, and tracking of time-critical targets.

Due to random back-offs in CSMA/CA-based 802.11 DCF, it is difficult to provide bounded latencies. Furthermore, when the channel becomes saturated, the latencies become too high for acceptable quality of video transmission. A number of approaches has been proposed to support prioritized transmissions for real-time IEEE 802.11 traffic. For example, in [3], the existing MAC protocol is enhanced for real-time traffic by selectively discarding packets and sharing station information. In [4], the authors propose a new Quality of Service-aware MAC protocol for mobile and ad hoc networks that chooses the nodes to forward the data in order to avoid congestion. Both of these approaches necessitate changes to the underlying IEEE 802.11 standard, which implies that firmware changes are made as MAC functions are normally hard-coded on a wireless card. Given the widespread deployment of IEEE 802.11, an approach to utilize this widespread penetration without further firmware change is more generally applicable.

Another important consideration is the bandwidth constraint of the network. The capacity of wireless networks has been studied extensively. Much of the previous work has focused on computing theoretical throughput bounds [5, 6]. Bianchi [5] used Markov processes to analyze the saturation throughput of 802.11 Distributed Coordination Function (DCF). [6] examined the capacity of several simple configurations of ad hoc networks and traffic patterns using simulation and analysis from first principles. They observe that the capacities are much lower from their optimal capacities. They attribute this to the fact that 802.11 DCF is unable to find the right rate-allocation for the nodes on the channel. Also, the existence of hidden and exposed nodes reduces the capacity further, especially in multi-hop topologies.

In this paper, we describe an application-level TDMA-based protocol for multi-hop networks. In [7], the authors show how TDMA can facilitate bounded latency in communication given an appropriate schedule in sensor networks. In [8], Song et. al. give an introduction to the architecture of WirelessHART which is a TDMA-based wireless mesh networking technology and describe the implementation of the protocol stack. In [9], the authors implement a TDMA protocol over 802.11 for single-hop wireless networks to reduce power consumption of the nodes. A loosely synchronized Overlay MAC Layer (OML) has been used in [10] in order to improve the performance of 802.11 networks. They use a weighted-fair-queueing-based distributed slot allocation mechanism for scheduling the transmission of the nodes. Although they address interference from within the network, they do not consider the effects of external interference. External interference could cause a node to back-off during its transmission slot and transmit in a slot assigned for another node. In our protocol, we eliminate exponential back-off by making the retry limit zero. This ensures that a node transmits only in its assigned slot and eliminates the possibility of external interferences disrupting the functioning of the TDMA protocol.

Apart from optimized performance of the wireless network, a key requirement is a smart camera system which is capable

of capturing, processing, compressing and transmitting image data. Wolf et. al. [11] outline the importance of smart cameras as embedded systems and the various design aspects of smart cameras. In [12], the authors have developed a smart camera using an Altera FPGA and LUPA 4000 CMOS image sensor. It can generate up to 200 frames per second (fps) of VGA data but this implies that it requires correspondingly large amounts of bandwidth. This high data-rate requirement may not be suitable for bandwidth-constrained wireless networks. A neuromorphic temporal contrast vision and a DSP-based smart camera is described in [13]. In [14], CMUcam3 along with FireFly sensor nodes have been used to form an energy-efficient sensor network to perform in-home activity clustering. Such a network is too resource-constrained to perform complex vision tasks and was not intended for video transmission. For OmniEye, we developed DSPcam, a wireless smart camera node that communicates over a time-synchronized 802.11 mesh network.

3. The OmniEye System

In this section, we will discuss the goals and components of the OmniEye system.

3.1. Goals

The primary goal of the OmniEye system is to enable video streaming with local processing over commodity 802.11 wireless networks. We propose the use of smart cameras with local processing capabilities to automatically tag video as it is captured. These tags should be stored with the archived video footage in an attempt to expedite searching and clustering of important events. For example, in the event of a robbery, it would be ideal if a perpetrator could be selected and then automatically traced back across multiple cameras to his point of origin. In current systems, this requires extensive manual manipulation of multiple video feeds. OmniEye leverages the TSAM MAC protocol to improve the scalability in terms of number of video streams in a given network.

3.2. System Architecture

Figure 1 depicts the architecture of OmniEye. It consists of smart camera nodes called DSPcams communicating over an IEEE 802.11 mesh network to the base station with few high-resolution PTZ cameras connected over ethernet to the base station. The DSPcams transmit video data continously at a low frame-rate by default. If an event of interest such as motion or appearance of a new object occurs in the DSPcam's Field of View (FOV), it increases the frame-rate. The DSPcams also tag the data in real-time with relevant information and transmit it to the base station. Such a mechanism optimizes the usage of bandwidth, and enables easy and effective retreival of data.

CSMA-based networks perform better if data are collected sporadically keeping the instantaneous load on the network

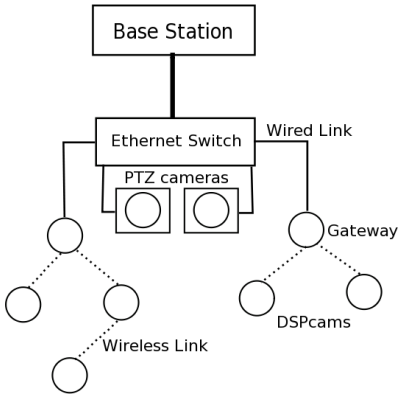

Figure 1. Architecture of OmniEye.

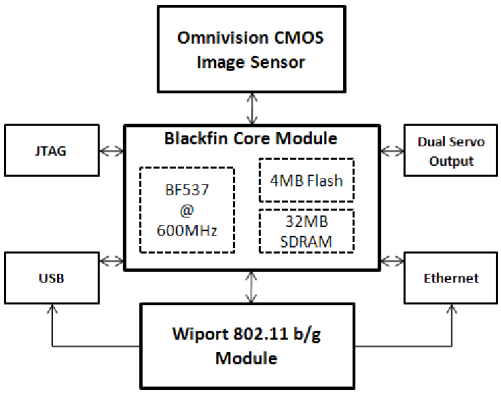

Figure 2. Block diagram of DSPcam Hardware.

relatively low. In a surveillance-type application, periodic and synchronous sampling of the environment is necessary. Such applications also require transmission of large blocks of data (e.g. images). These communication patterns require efficient scheduling. Over-provisioning bandwidth will only go so far since cameras and other high-bandwidth sensors will likely consume the additional resources. The real-time transmission of video data and PTZ camera control packets is facilitated by a Time-Synchronized Application Level MAC Protocol (TSAM). Every node in the system is assigned transmission slots dynamically depending upon the frame-rate and quality of video data. This time-synchronized mechanism provides for bounded end-end delay, improves throughput and also maintains channel fairness.

In the following sections, we will give a detailed description of our smart camera nodes i.e. DSPcam and our Application-level multi-hop MAC protocol, TSAM.

3.2.1. DSPcam: A Wireless Smart Camera Node. DSPcam shown in Figure 3, is our custom-designed high-performance wireless smart camera-networking platform. It consists of the following main hardware components: a CMOS image sensor, the central processor and the wireless module. Figure 2 depicts the hardware architecture.

The CMOS sensor is an OmniVision OV9653, which

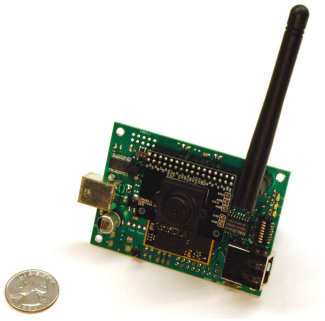

Figure 3. The DSPcam hardware.

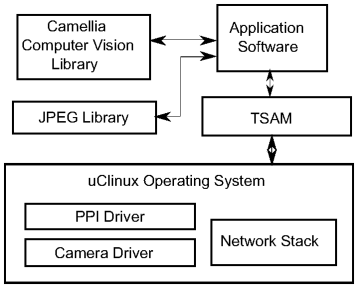

Figure 4. Block Diagram of DSPcam Software Components.

(a)

(b)

Figure 5. (a) DSPcam GUI showing a particular location. This image is taken as the background. (b) As soon as a new object enters the screen, motion is detected and displayed on the GUI.

is capable of producing a maximum resolution of SXGA (1280x1024) at 15 fps. In our system, we have configured the sensor to give VGA images in YUV format at 30 fps. The central processor is an Analog Devices Blackfin operating at 600 MHz. The Blackfin processor sets up the sensor through the Serial Camera Control Bus (SCCB) and uses its Parallel Peripheral Interface (PPI) to capture the images. DSPcam has 32 MB SDRAM. The captured images are transferred into RAM using Direct Memory Access (DMA). The DSPcam applies the necessary image processing algorithm and compresses the image. The processed image is then transmitted over IEEE 802.11 using a Lantronix wireless module called *WiPort*. Wiport supports both 802.11b/g. A USB interface is also provided for connecting the DSPcam to a PC platform for testing.

The software architecture is built on an open-source development environment comprised of the *uClinux* embedded operating system, the *gcc* compiler and a large set of Linux utilities. Figure 4 depicts the software architecture of the system. An open-source image processing and computer vision library called Camellia [15] runs natively on the DSPcam. The familiar Linux environment combined with the computer vision library enables quick development of software necessary for our application. We developed an API for interfacing with TSAM, which abstracts the details of time synchronization from the user.

3.2.2. Image Processing on DSPcam. Before any data are transmitted over the network, the required algorithms locally identify the image *tag* and then compress the image. There are many computer vision algorithms ranging from object detection to abnormal activity detection in the literature. Our focus in this paper is not the computer vision problem but the systems infrastructure that enables and enhances the application with existing algorithms for surveillance purposes.

The Omnivision image sensor has been configured to give images at VGA (640x480) resolution in the YUV color space. These are sub-sampled to form a 320x240 image. This image is compared against previously stored images to detect motion. The processed image is then converted to RGB format and JPEG-compressed. A JPEG compression with a quality factor of 70% reduces the size of the image by almost 90% while still maintaining good image quality. Although the CMOS sensor can produce 30 fps, the actual frame rate transmitted depends on the number of slots assigned to the node as well as the presence of motion.

Object and motion detection are done using an adaptive background subtraction on a frame-by-frame basis. Figure 5(a) and Figure 5(b) show the DSPcam Graphical User Interface where the background and motion detection is shown.

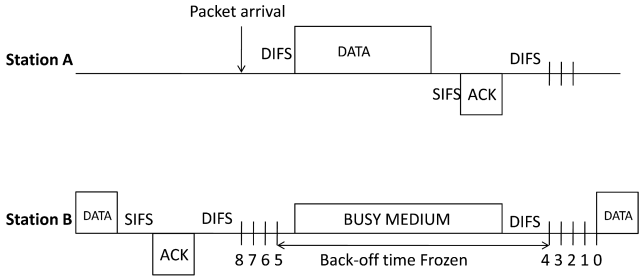

Figure 6. Time-line of the IEEE 802.11 DCF operation.

4. Time-Synchronized Application-level MAC (TSAM) Protocol

We first analyze the existing IEEE 802.11 Distributed Coordination Function (DCF) and identify its limitations. Next, we describe in detail our proposed TSAM protocol and how it overcomes these limitations. In our experiments, DSPcam nodes are communicating over an IEEE 802.11b medium although it also supports 802.11g. We use lower bandwidth to highlight the saturated operation with a fewer number of nodes. 802.11g should follow similar trends at higher traffic levels.

4.1. Performance Analysis of IEEE 802.11 DCF

The IEEE 802.11 Distributed Coordination Function (DCF) is based on Carrier Sensing Multiple Access with Collision Avoidance (CSMA-CA). Figure 6 shows a sample time-line of the DCF operation. There are primarily two metrics that we focus on: throughput and predictability of packet transmission delay. Bianchi [5] was among the first to develop an analytical model for the behavior of 802.11 DCF. He used Markov processes to analyze the saturation throughput of IEEE 802.11 DCF. Jun et al. [16] have addressed the theoretical maximum throughput (TMT) of IEEE 802.11 assuming an ideal channel and no losses due to collisions. The theoretical maximum throughput (TMT) as given in [16] can be obtained by dividing the MAC Service Data Unit (MSDU) by the time it takes to transmit it:

$$TMT = \frac{MSDU\ size}{Delay\ per\ MSDU}$$

The duration of each delay component is determined from the IEEE 802.11 specification [17]. All delay components vary with spread spectrum technology but not with the data rate. In OmniEye, the wireless communication uses Direct Sequence Spread Spectrum at a bandwidth of 11Mbps. RTS-CTS is disabled. Hence, the delay should only be due to Distributed Inter Frame Space (DIFS), back-off, ACK and the actual data transmission.

Figure 7 depicts the variation of TMT with MSDU size in bytes. The maximum transmission size at the IP layer is 1500 bytes. Packets larger than 1500 bytes will be fragmented, although 802.11 MAC layer accepts a default of 2346 bytes. In our system, in order to avoid IP fragmentation, we have used a maximum packet size of 1500 bytes.

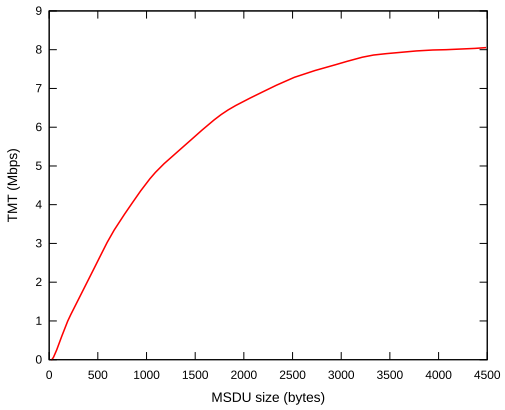

Figure 7. Theoretical Maximum Throughput (TMT) for various MSDU size.

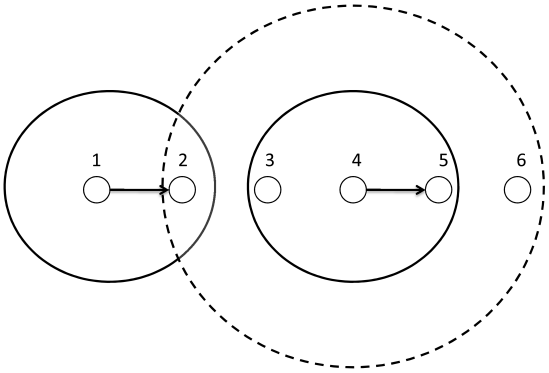

Figure 8. Chain Topology of 6 nodes. The dotted lines represent an interference region, and the solid lines represent the transmission/reception region.

We observe from Figure 7 that the value for TMT for 1500 bytes is close to 6 Mbps. This implies that if every camera node generates video at 1 Mbps, then upto a maximum of only six nodes can be supported in a channel.

Li et. al. [6] have examined the interactions of the 802.11 MAC and *ad hoc* forwarding, and the effect on capacity for several simple configurations and traffic patterns. They observed that the capacities achieved by 802.11 is markedly less than optimal for simple chain and lattice networks with regular traffic patterns. Consider the network shown in Figure 8 where node 1 is the source and node 6 is the sink. An ideal MAC protocol could achieve a chain utilization of 1/4. The actual experimental channel utilization observed is close to 1/7, much less than the ideal utilization. This is largely due to hidden terminal effects. Node 1 can sense the transmissions from nodes 2 and 3. Node 2 on the other hand suffers interference from nodes 1, 3 and 4. The first node transmits more packets into the network generating a backlog. This results in excessive packet loss in the successive nodes. In addition to non-uniform channel allocation of 802.11 DCF, the back-off can cause further reduction in throughput. Suppose that node 1 attempts to transmit a packet to node

2 when node 4 is in the middle of its transmission. Due to interference from node 4's transmission, node 2 would not get the packet resulting in node 1 backing off for the entire duration of node 4's transmission. Finally, when the channel is free, node 1 may remain backed off during the time which it could have utilized for transmission. Enabling RTS/CTS does not greatly improve performance due to RTS/CTS collisions and exposed terminal problems.

The problem of hidden nodes in contention-based mechanism also impacts the transmission delay. Consider the example described before. In such a scenario, when node 1's packet does reach the sink, it may no longer be useful. The randomness in the transmission causes jitter in the video that are only overcome with large buffer periods. There have been many changes in the 802.11 MAC layer that have been proposed in order to meet the real-time applications such as ours. Such changes require new hardware and firmware, which is not practical unless large-scale production is undertaken. In contrast, TSAM can be deployed using existing hardware. The fact that TSAM is implemented in software makes it easier for adapting TSAM to meet the diverse requirements of the ever-increasing spectrum of wireless applications.

4.2. Protocol Overview

TSAM is a TDMA protocol consisting of a central node and multiple slots. The central node initiates the TDMA cycle by transmitting a *Beacon* packet. This marks the beginning of a slotted data communication period. The nodes synchronize themselves upon the reception of this beacon packet. The communication period is defined as a fixed-length cycle and is composed of multiple frames. The beacon packet serves as an indicator of the beginning of the cycle and the first frame. Each frame is divided into multiple slots, where a slot duration is the time required to transmit a constant number of maximum-sized packets. TSAM supports two kinds of slots: *Scheduled* slots within which nodes are assigned specific transmit and receive time slots; a series of unscheduled or *Contention* slots where nodes, which are not assigned a scheduled slot select a transmit slot at random. Nodes operating in *Scheduled* slots are provided timeliness guarantees as they are granted exclusive access to the shared channel and hence operate in a collision-free manner. In our default implementation, each cycle consists of 3 frames and each frame consists of fifty (50) 21ms slots. Thus, the cycle duration is 3.15 seconds.

4.3. Slot Design

A characteristic of the IEEE 802.11 DCF MAC protocol is that the medium must be free for a minimum of the DIFS interval, i.e. 50 μs prior to each transmission. Accommodating this characteristic in a slotted medium access method necessitates a guarantee that there is a minimum of 50 μs before the start of each slot during which the medium is idle. To satisfy this constraint, we use a slot structure with *gaurd* times which guarantees a minimum of 50 μs at the end of the slot within which no wireless transmission is allowed to take place.

If a slot is guaranteed to encapsulate at most one transmission and the guard space is always free from transmission, there should be no contention for the wireless medium during this transmission. A scheduled transmission in a slot may be delayed by external wireless interference by other IEEE 802.11 hosts not participating in the TDMA protocol, or other wireless devices transmitting in the transmission range of a particular host. Satisfying the requirement that packets be fully encapsulated within a single slot is difficult given the unpredictable number and duration of back-off intervals in the underlying 802.11 DCF protocol due to external interferences. The back-off and retry from one node could affect the slot transmission of another node, setting a cascade of back-offs through the network resulting in the entire disruption of TSAM. In order to overcome this, we have used a back-off limit of zero. This implies that if a node does not find the channel to be free for DIFS amount of time, instead of backing off and retrying transmission, it simply drops the packet. Although this might sound detrimental to the system performance, given the nature of video data, it is better to drop a frame rather than generate a backlog of traffic.

4.4. Communication Scheduling

The default communication pattern is a statically assigned schedule depending upon the number of nodes in the network. For example, if there are 10 nodes that are 1 hop from the network gateway, then each node could be assigned 5 slots per frame. This ensures a fair allocation of bandwidth. An allocation of 5 slots per frame ensures a frame rate of 5 frames per second (FPS). If more nodes are added to the network, then the FPS would have to be reduced or the image data needs to be compressed to accommodate higher rates within the allocated slots. Below certain minimum frame rate and image compression quality, the data may no longer be acceptable. Admission control needs to be done in order to maintain the minimum quality of data. In our network, we want to maximize the number of camera nodes deployed to have wider coverage and multiple views of the same area while meeting the minimum data quality requirements. In order to maximize video throughput, we wish to maximize the number of collision-free concurrent transmissions. Previous methods for allocating TDMA slots given a network topology formed a spanning tree over the network graph, which is colored such that nodes within two hops of each other have unique time-slots assigned to them. The spanning tree ensures total network connectivity, while two-hop coloring guarantees collision-free communication in the presence of hidden nodes.

Our surveillance application also requires support for alert packet transmission. The scheduling has to be done such that the packet delay transmission from a node that has detected an event to the gateway is minimized. There exists a trade-off between maximizing concurrency and reducing the delay. The generation of maximum concurrency schedule is similar to the distance-two graph coloring problem that is known to

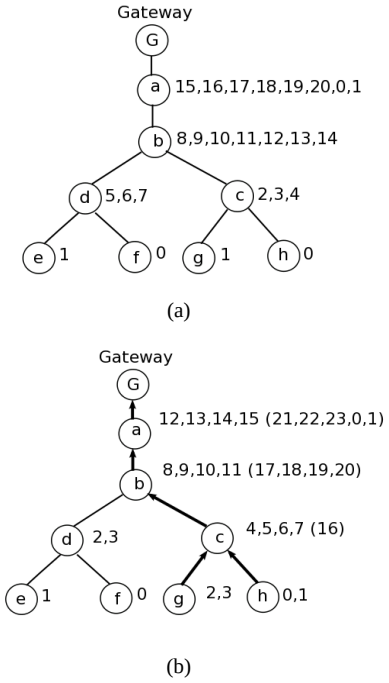

Figure 9. (a) The transmission schedule of nodes under normal operation. All the traffic is directed to the gateway. (b) The transmission schedule of nodes when nodes *g* and *h* detect events of interest.

be NP-complete [18]. In practice, many heuristics can work well and result in a small constant deviation from the optimal.

Another requirement is dynamic rescheduling based upon the rate request from the nodes. During a cycle, if a node transmits a *Request* packet to the central monitoring node, then an updated schedule needs to be transmitted. The new schedule should be such that the nodes on the upstream path to the central node have enough slots for transmitting their own data as well as the data from the other nodes in the path below it. This is accomplished by reducing the number of slots assigned to the nodes other than the ones that sent the recent *request* packet. In case such a scheduling is not possible for the particular quality of data, then certain nodes that do not have any important data to send, i.e. there has not been any change in the TAG information for the last 3 cycles, will not be provided any slots at all.

Figure 9(a) shows the scheduling of a network of camera nodes for video transmission from every node. The scheduling is done starting from the leaf nodes. Every node is allocated one slot for transmission of its own data and an additional number of slots that equal the number of slots allocated for the nodes below it. For example, in Figure 9(a), the leaf nodes *e, f, g, h* are given the lowest slot numbers. The nodes above it, *d* and *c*, are given 3 slots, two for carrying traffic of nodes below them and one for transmitting their own data. In this manner, slots are assigned up to the gateway. Now, suppose that nodes *g* and *h* detect events of interest and need higher bandwidth upstream path to the gateway. Figure 9(b) shows the scheduling for minimum latency transmission to gateway from nodes *g* and *h*. The scheduling is done such that the nodes having important information, i.e. the nodes that detected events of interest are given the lowest slot numbers and the upstream nodes are first assigned slots to carry this information to the gateway. Also, the nodes *g* and *h* are given more slots, in this case two, for transmission of higher-quality video data. Only after this scheduling is done that the slots for carrying the data from other nodes are assigned. In this way priority for video containing important information is given. Also, it is assumed that only the nodes that most recently detected events would be sending control packets to the PTZ camera or sending the alert packets. Such a scheme guarantees bounded end-to-end transmission delay for important data in the network.

4.5. Clock Synchronization

Time synchronization is a critical part of our TSAM protocol. Clock synchronization accuracy of within 50 μs is desirable as this is the minimum gap within which the medium must be guaranteed to be free using IEEE 802.11 CSMA/CA. Thus, this serves as the minimum bound on the guard space at the end of each slot. The major problem with precise network time synchronization is non-determinism. The major sources of these errors are *Send Time*, *Access Time*, *Propagation Time* and *Receive Time*. It has been observed by Elson et. al. [19] as well as Verissimo and Rodrigues [20] that although *Send Time* and *Access Time* may be unknown and can vary highly from message to message, the nature of broadcast dictates that for a particular message, these quantities are the same for all receivers. Reference-Broadcast Synchronization (RBS) [19] uses this principle to eliminate errors introduced by *Send Time* and *Access Time*. Although this scheme does provide time synchronization, in an over-laid TDMA protocol such as TSAM where it is operating over an underlying scheme, we also require the nodes to *complete* their transmission in their alloted slots. In TSAM, the nodes synchronize themselves from the packets broadcast by its neighboring nodes. When any node transmits, the packet also contain its transmission slot and frame number. Since every node is always listening, it can receive periodic broadcast packets which contains the neighboring node's transmission slot and frame number. Using this information it can correct its clock to the global clock. We found through experimental evaluation the period of beacon transmission in order to maintain minimal clock drift to be 3 frames (3.15 seconds). We further extend RBS to reduce the variability of *Send* and *Access Time*. We also give the network communication tasks in the DSPcams the highest priority. This further reduces the jitter in *Send* and *Receive Time* values. In order to reduce the jitter in *Access Time*, as mentioned earlier, we made the retry limit equal to zero.

5. Experimental Evaluation

In this section, we present a performance evaluation of TSAM. The experimental setup for the single-hop evaluation

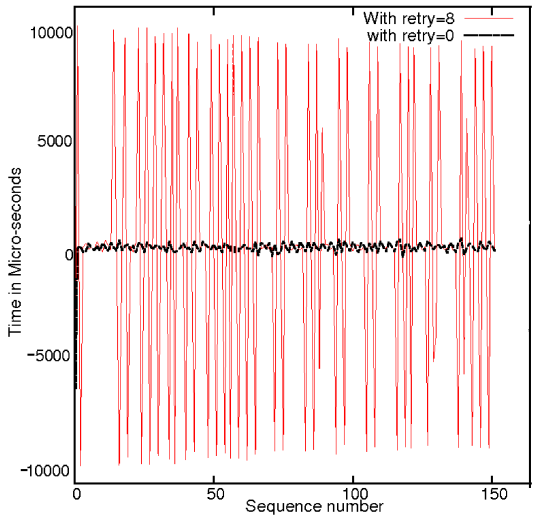

Figure 10. Receive jitter over a sequence of packets.

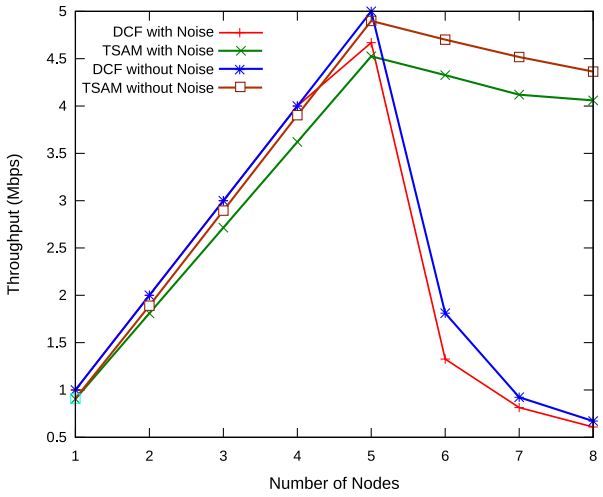

Figure 11. Throughput of TSAM and DCF for various number of nodes.

consists of eight DSPcams transmitting real-time video data to a central monitoring node. Each DSPcam transmits at 4 fps which is about 1 Mbps. Since the actual JPEG compression of different images would result in different data sizes, for the experimental purposes, we use a static JPEG image that is the same for all nodes to transmit. In this way, we ensure that all the nodes transmit the same amount of data and the evaluation is consistent. For multi-hop network evaluation, we use a chain topology.

Figure 10 depicts the receive jitter for a set of packets received on a node. For this experiment, we had one node broadcast packets continuously and record the receive times on one of the receiver nodes. The jitter for DCF operation with retry limit of 8 is of the order of 10 msec. According to our TSAM protocol, the retry limit needs to be zero in order to reduce the jitter which, in turn, ensures the transmission of a packet within in the assigned slot. We observe that

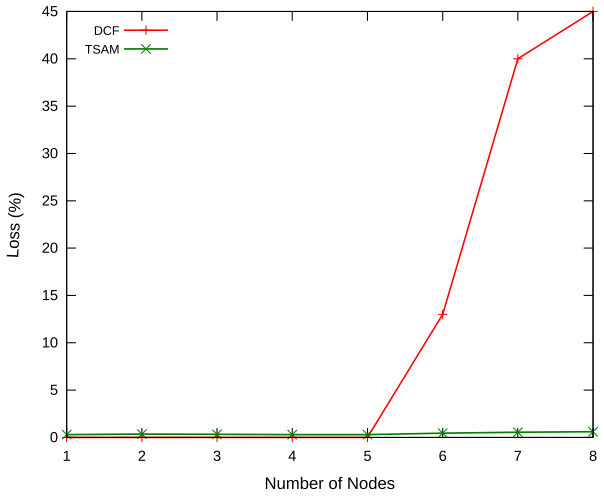

Figure 12. Packet loss as the number of nodes is increased in the network.

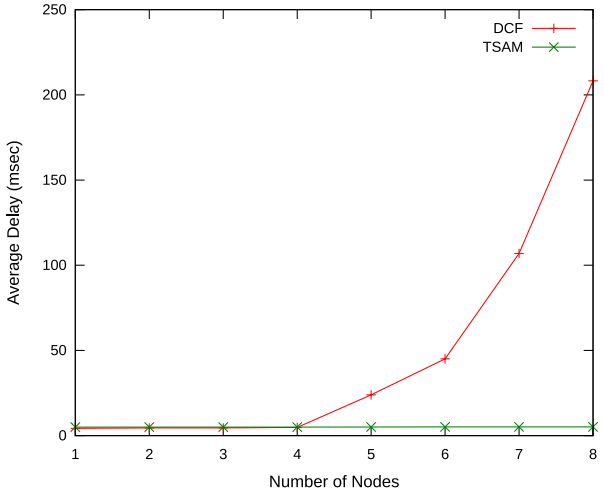

Figure 13. Average delay as the number of nodes is increased in the network.

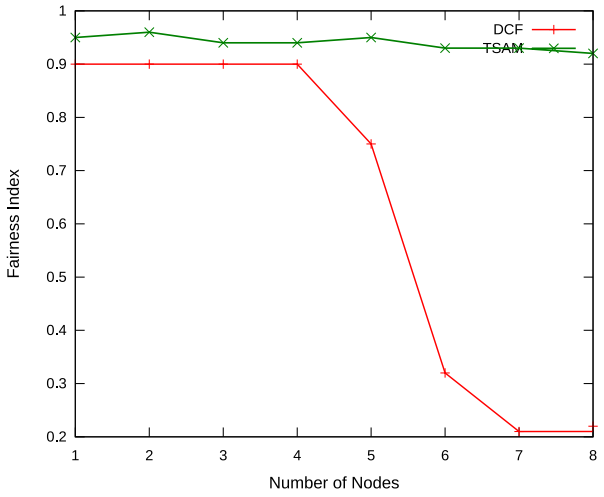

Figure 14. Single-Hop Fairness Index as the number of nodes is increased.

upon making the retry limit zero and giving the network communication task the highest priority in the system, the jitter is reduced significantly.

We conducted the single-hop experiment in noisy conditions as well as noise-free channel conditions. Under noisy conditions, channels 1, 2 and 11 were occupied by external networks. Under noise-free conditions, our network was the only active channel. Figure 11 depicts the throughput of TSAM and 802.11 DCF for various number of nodes under noisy and noise-free channel conditions. We observe that the throughput of DCF (as expected) drops sharply after 5 nodes. TSAM continues to perform at the optimum throughput up to eight nodes. After 5 nodes, the channel becomes saturated. Under saturation, all the nodes have data to transmit and in IEEE 802.11 DCF would collide with one another at a high rate resulting in a rapid throughput drop. This also explains the high loss and unpredictable delay shown in Figure 12 and Figure 13 respectively. Under TSAM, every node is given exclusive channel access in its alloted slot, thereby eliminating collision. TSAM therefore continues to deliver data even under saturation conditions with bounded delay. This is reflected in the fairness of channel allocation. We use the *Fairness Index* given by the following equation [21]:

$$\frac{(\sum_{i=0}^{n} x_i)^2}{n\sum_{i=0}^{n} x_i^2},$$

where n is the total number of nodes and x is the total number of packets received from a particular node for a constant duration of time.

Figure 14 plots the Fairness Index. Although the Fairness Index is close to 1.0 up to 5 nodes, it goes below 0.35 upon the addition of the 6^{th} node when IEEE 802.11 DCF is used. In TSAM, the Fairness Index remains close to 1.0. Under saturation conditions, TSAM rate-allocates the nodes so that the maximum throughput of the entire system remains under the maximum supportable channel bandwidth. That is, the slots are divided among the nodes equally.

Figure 16 shows the average delay with respect to the number of hops in a chain. We observe that, initially, TSAM has higher latency compared to DCF due to its higher overhead. As the number of hops increases beyond 3, the DCF latency increases dramatically while TSAM continues to function with an approximately linear increase. Hence, it behaves in a predictable manner providing bounded delay. In Figure 15, we observe that the fairness index for DCF reduces as the number of hops is increased. In TSAM, it remains close to 0.9. In the case of throughput, we observe that as the number of hops is increased, DCF throughput initially increases and then starts decreasing, whereas TSAM throughput increases and continues to remain close to its peak.

6. Concluding Remarks

Video cameras are becoming a ubiquitous feature of modern life and are useful for surveillance, crime prevention and forensic evidence. The systems infrastructure for streaming

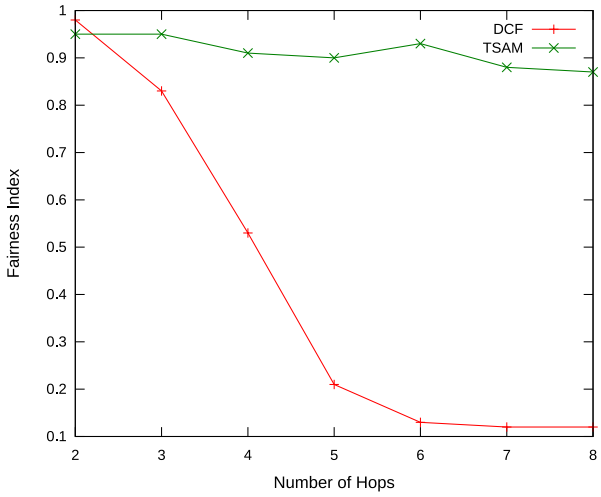

Figure 15. Multi-Hop Fairness Index as the number of hops is increased.

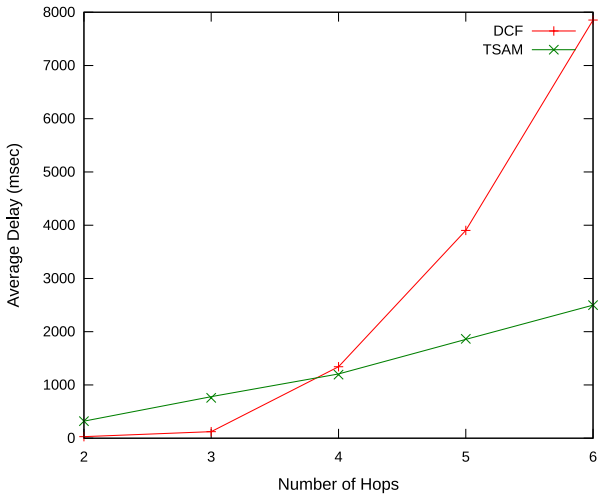

Figure 16. Average Delay as the number of hops is increased.

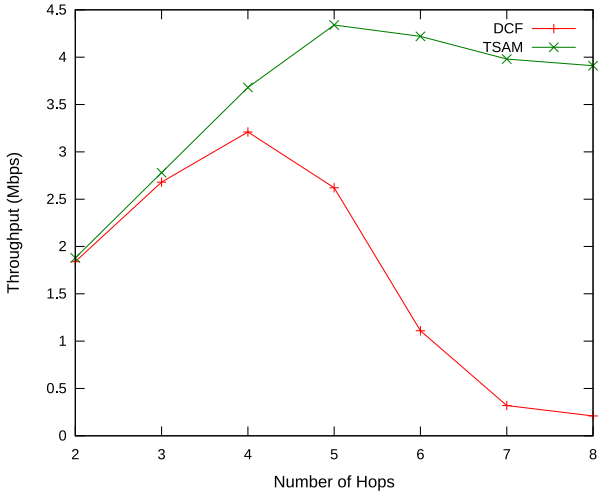

Figure 17. Throughput as the number of hops is increased.

multiple real-time video flows, transmitting automated real-time control/alert packets and enabling machine intelligence is still in its infancy. In this paper, we described *OmniEye*, a wireless mesh network of smart cameras using *DSPcams* and a sparse number of PTZ cameras. Our system brings down the cost and difficulty of deployment. The DSPcams annotate the video with *tags* that describe the event occurring in its Field of view. Tags are used to draw the operator's attention and also for search and retrieval of useful data during forensic analysis. The DSPcams can control the PTZ cameras in real-time to zoom in on the target when detected. Such automation reduces human intervention, improving reliability and reducing latency of event detection.

Two important requirements in a system such as OmniEye where there are many wireless nodes transmitting video data are the abilities to transmit the video with minimum jitter and to transmit the alert packets with bounded delay. We observe that existing IEEE 802.11 DCF protocol fails to meet these requirements. Also, the throughput starts deteriorating rapidly when the channel utilization increases to 50% and this drop in performance is much worse for multi-hop networking. A key contribution of this paper is the design and implementation of a Time-Synchronized Application Level (TSAM) MAC protocol. TSAM can operate on COTS hardware and support multi-hop networks with built-in clock synchronization. TSAM reduces collisions by explicitly scheduling communications. It is able to provide reliable bounded delay in transmission. The reduced jitter in video, improves reliability and reduces latency of control/alert packet transmission. We presented experimental results to validate the performance of TSAM.

In the future, we plan to extend this work to support detection of particular types of events and indexing of these events to enable event-based retrieval of video data. We also plan to integrate Motion-JPEG compression into our system. This compression scheme varies the level of compression within an image based on Regions Of Interests (ROI). This compression scheme will further reduce the band-width requirements and benefit greatly from TSAM's dynamic bandwidth allocation capabilities.

7. Acknowledgement

This work is a partial result of Project 73522Q1200 conducted by Industrial Technology Research Institute under the sponsorship of the Ministry of Economic Affairs, R.O.C, Taiwan.

References

[1] Tian He, Pascal A.Vicaire, Ting Yan, Liqian Lup, Lin Gu, Gang Zhou, Radu Stoleru, Qing Cao, John A.Stankovic, and Tarek Abdelzaher, "Achieving Real-Time Target Tracking Using Wireless Sensor Networks",*12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2006

[2] Loyall J, Schantz R, Corman D, Paunicka J, Fernandez S, "A distributed real-time embedded application for surveillance, detection and tracking of time critical targets", *RTAS*, 2005

[3] Rusty O. Baldwin, Nathaniel J. Davis. IV, Scott F. Midkiff, "A real-time medium access control protocol for ad hoc wireless local area networks",*ACM SIGMOBILE Mobile Computing and Communications*, 1999

[4] S. Sivakeesar, G. Pavlou, "Quality of service aware MAC based on IEEE 802.11 multihop ad-hoc networks" *Wireless Communications and Networking Conference*, 2004

[5] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE Journal in Selected Areas: Communication*, 2000.

[6] Jinyang Li, Hu Imm Lee Charles Blake, Douglas S. J. De Couto, and Robert Morris, "Capacity of ad hoc wireless networks" *7th ACM International Conference on Mobile Computing and Networking*, 2001.

[7] A. Rowe, R. Mangharam, R. Rajkumar, "RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks", *Third IEEE International Conference on Sensors, Mesh and Ad Hoc Communications and Networks (IEEE SECON)*, 2006.

[8] Jianping Song, Song Han, Mok A.K., Deji Chen, Lucas M, Nixon M, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control",*RTAS*, 2008

[9] Jim Snow, Wu-chi Feng, Wu-chang Feng, "Implementing a Low Power TDMA Protocol Over 802.11" *Wireless Communications and Networking Conference*, 2005.

[10] Ananth Rao and Ion Stoica, "An Overlay MAC layer for 802.11 Networks", *MOBISYS*, 2005.

[11] Wayne Wolf, Burak Ozer, and Tiehan Lv, "Smart cameras for embedded systems", *IEEE Computer*, 2002.

[12] F. Dias, F. Berry, J. Serot, F. Marmoiton, "Hardware, Design and Implementation Issues on a Fpga-Based Smart Camera", *International Conference on Distributed Smart Cameras (ICDSC)*, 2007.

[13] Litzenberger, M., Belbachir, A.N., Schon, P., Posch, C., "Embedded Smart Cameras for High Speed Vision", *ICDSC*, 2007.

[14] Anthony Rowe, Dhiraj Goel, Raj Rajkumar, "FireFly Mosaic: A Vision-Enabled Wireless Sensor Networking System",*RTSS*, 2007.

[15] *http://camellia.sourceforge.net*

[16] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications", *IEEE International Symposium on Network Computing and Applications*, 2003.

[17] Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Standard 802.11.

[18] Hari Balakrishnan et al., "The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks", *IEEE Journal on Slected Areas in Communication*, 2004.

[19] Jeremy Elson, Lewis Girod, and Deborah Estrin, "Fine Grained Network Time Synchronization using Reference Broadcasts", *Fifth Symposium on Operating Systems Design and Implementation*, 2002.

[20] Paulo Verissimo and Luis Rodrigues, "A posteriori agreement for fault-tolerant clock synchronization on broadcast networks", *Annual International Symposium on Fault-Tolerant Computing*, 1992.

[21] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems", *DEC Research Report TR-301*, 1984.