Priority-Based Routing Framework for Multimedia Delivery in Surveillance Networks

Adil A Sheikh
Science and Technology Unit
Simplicity Labs
Umm Al Qura University
Makkah, Kingdom of Saudi Arabia
aasheikh@uqu.edu.sa

Emad Felemban, Saleh Basalamah
Department of Computer Engineering
Simplicity Labs
Umm Al Qura University
Makkah, Kingdom of Saudi Arabia
{eafelemban, smbasalamah}@uqu.edu.sa

***Abstract*—Wireless sensor network consisting of nodes equipped with cameras or advanced low-cost image sensors is known as a Visual Sensor Networks (VSN). The main function of VSNs is to capture images and send them to sink nodes for processing. One of the most common applications of VSN is surveillance. Such applications require large amounts of data to be exchanged between camera nodes and sink. Image data is considerably larger than common sensor data such as temperature, humidity, pressure, etc. For data delivery in VSNs, the communication is constrained by many stringent QoS requirements like delay, jitter and data reliability. Moreover, due to the inherent constraints of wireless sensor networks such as low energy, limited CPU power and scarce memory, the architect of VSN must choose appropriate topology, image compression algorithms and communication protocols depending on his/her application. This paper focuses on one of these aspects, namely the communication protocol for VSN. In this paper, we present a new routing framework for VSN to deliver critical imagery information with system's time constraint. We have implemented our proposed framework using Contiki and simulated it on Cooja simulator to support our claim.**

Keywords-Routing Framework; VSN; Image Transmission; Priority-Based Routing; Contiki; Cooja

I. INTRODUCTION

The primary requirement of a wireless sensor network is to sense environment factors using low-power, low-cost sensors and route meaningful data to power-rich sink nodes for processing. This requirement becomes challenging in a VSN as the amount of data to be transferred is much more than a traditional wireless sensor network due to the type of data being shared. Applications of surveillance require very large amounts of data to be exchanged between camera nodes and sink. In traditional wireless sensor networks that sense light, humidity, pressure, etc. the traffic generated by a sensing node is limited to the scalar data [1]. In most cases, the memory size required to store and send is 16-bits per reading [1]. On the other hand, a VSN node, equipped with a camera generates vector data. For instance, a raw Red-Green-Blue (RGB) image of 128 x 128 pixels with 24-bits per pixel (8 bits per color) will be of 128 x 128 x 24 = 393216 bits (approximately 48 kilobytes). These are magnitudes larger than traditional sensor data.

To minimize the size of the image data, image compression techniques such as Discrete Cosine Transforms [2] or Discrete Wavelet Transforms [3][4] can be used. Although these algorithms reduce the size of an image, yet it is not comparable to traditional sensors data. Therefore, image data compression is not enough. The processing power of each node is also limited. Additionally, the topology of the network and routing protocols play a crucial role in transporting imagery information from visual sensing nodes to sink nodes. Hence, the tasks of capturing image data, compressing it and sending it to sink are some of the most challenging tasks faced by VSN architects.

As mentioned before, using image compression algorithms the size of data can be reduced to some extent. Also, a category of image compression algorithms generate multiple layers of compressed image data. The first layer contains the most prominent features of the image, for example, the edges of objects or coarse image data. The subsequent layers contain the details that when merged with the first layer, restore the original image. Some image processing algorithms consist of multiple passes requiring different levels of details of the encoded image for each pass. Using such algorithms in VSNs, system response time can be reduced. If the sink nodes receive image data required for first pass sooner than data required for subsequent passes, it can start processing the first pass and take action accordingly while data of subsequent layers arrive at the sink node. This paper helps alleviate the routing challenges of such image processing algorithms by proposing a routing framework based on four features. (1) The visual sensing nodes should be able to specify priority to outgoing packets. In this way, image data for first pass can be sent at higher priority than data for subsequent passes. (2) The intermediate or routing nodes should be aware of packet priority so that higher priority packets are forwarded before lower priority packets. (3) If packets from two nodes collide, high priority packets should be retransmitted before low priority packets. (4) Finally, in event of congestion, lower priority packets should be dropped before any high priority packet is dropped.

The next section summarizes the various communication protocols being used in VSN architectures as of today. Section III discusses typical VSN application scenario along with details of VSN components essential for delivering critical image information within system's time constraints.

Section IV defines our proposed priority-based routing framework. The implementation of our proposed protocol is discussed in Section V. Simulations were carried out to quantify the usefulness of the routing framework. In Section VI, simulation environment and results are discussed. Finally, the paper is concluded along in Section VII.

II. EXISTING ROUTING TECHNIQUES

The research on routing techniques for image transmission has mostly been limited to wired networks [5]-[9]. Research on QoS supported routing protocols for mobile ad-hoc networks has been summarized by Chen et al. [10] and Hanzo-II et al. [11]. Liebeherr et al. [12], Wang et al. [13], Stoica et al. [14], Younis et al. [15] and Soldatos et al. [16] discuss techniques to deliver image data on the Internet. None of these are applicable to VSNs.

Most of the work done in the field of routing techniques for VSNs has been conducted to achieve energy efficiency. The first routing protocol focused on QoS in VSNs by trying to minimize the average weighted QoS metric throughout the lifetime of the network. Sohrabi et al. [17] proposed Sequential Assignment Routing (SAR) that enforces maintenance of routing tables with status of all nodes.

RAP [18] is a priority-based routing protocol that uses velocity monotonic scheduling and geographical forwarding to achieve QoS, however, its requirement of geographical awareness can only be fulfilled by having a pre-defined network topology or additional hardware to determine geographical location.

SPEED [19], proposed by He et al., is a spatio-temporal, priority-based, QoS-aware routing protocol for sensor networks that provides soft real-time, end-to-end delay guarantees. SPEED does not provide differentiated packet prioritization. Moreover, a forwarding node can only forward the packet at a speed less than or equal to the maximum achievable speed even though the network can support it.

Real-time Power-Aware Routing (RPAR) [20] is another routing protocol that achieves application specific end-to-end delay guarantee at low power by dynamically adjusting transmission power and routing decisions based on the workload and packet deadlines. RPAR also calculates average link quality taking link variability into consideration.

Multi-path and Multi-SPEED (MMSPEED) routing protocol [21] supports probabilistic QoS guarantee by provisioning QoS in two domains, timeliness and reliability. MMSPEED adopts a differentiated priority packet delivery mechanism in which QoS differentiation in timeliness is achieved by providing multiple network-wide packet delivery speed guarantees.

III. VSN APPLICATION SCENARIO

This section explains the VSN application scenario discussed in this paper. In a typical VSN application, there are three types of nodes that make progressive image transmission possible. A brief description of each VSN node type and our network model is given below.

A. Visual Sensing Node

The visual sensing node contains the sensor that captures images. Depending on the application this sensor can be of type that captures multi-colored images, grey-scale images, thermal or infra-red images [22], etc. Nodes equipped with these sensors require more power to run additional hardware and software components such as frame-grabbers and image encoders. These nodes capture raw images, encode them and send them towards sink nodes for processing.

B. Intermediate Node

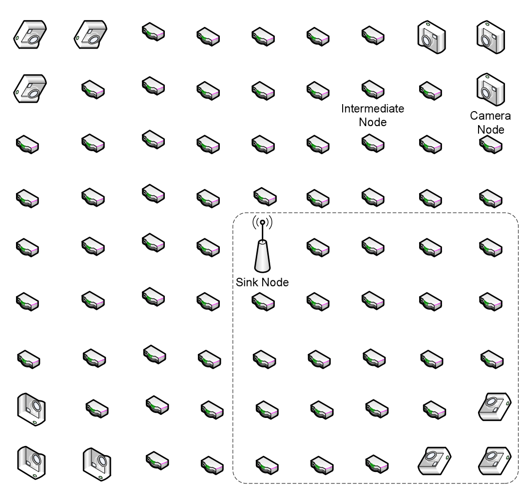
Their primary task of intermediate nodes is to send packets from camera nodes to the destination sink node. Depending on the VSN application, these nodes may also take part in sensing other scalar environmental variables such as temperature, humidity, pressure, level of certain chemicals, etc. Additionally, these nodes may also take part in encoding image data as a class of image encoding algorithms [23] offloads some processing to intermediate nodes in order to conserve power of camera nodes.

C. Sink Node

The sink nodes are responsible for processing the images captured by the camera nodes. For this purpose, sink nodes are power-rich and have high computation ability. In order to take action depending on the VSN application, these nodes may additionally contain actuators or may be connected to a fourth type of nodes called actuator nodes.

D. Network Model

The network model discussed in this paper does not restrict the number or position of any node type. One of the network topologies for a surveillance application is depicted in Fig. 1. For purposes of testing and evaluation, the network model we have used in this paper consists of one-quarter of this topology. Our visual sensing nodes are placed on the periphery of the network. The intermediate nodes are placed in the bulk of the network. In our network model, there is only one sink. It is also placed in the periphery of the network, on the opposite side of visual sensing nodes. This is depicted by the dashed-line in Fig. 1.



 ISBN: 978-1-61208-320-9

Figure 1. An Example of a Surveillance Network using VSN

In our scenario, the visual sensing nodes take images and encode them into two layers. First layer contains coarse image information collected in the first pass of image encoding and second layer contains fine image information collected in the second pass. The VSN application uses the routing framework to send this layer with high priority towards the sink. The sink uses first pass information to reconstruct the encoded image with a certain level of detail. Based on processing first pass, the sink can take action, if necessary. The VSN application uses routing framework to send second pass layer at low priority towards the sink. The sink uses the second pass layer to reconstruct a detailed image for further processing if image information from the first pass required additional image details to take action.

Most of the nodes in our network are the intermediate nodes. They are only responsible for routing packets from visual sensing nodes to sinks. They do not take part in sensing or sharing processing load of the sensing nodes or sink nodes. When the network is deployed, the intermediate nodes create routing tables that are necessary to take routing decision when packets are received. To achieve their primary task of routing image data from visual sensing nodes to sink nodes, the routing tables in intermediate nodes are updated throughout the lifetime of the network as some nodes may die due to depleted power or other environmental conditions, while other nodes may be added to the network when required. The routing framework makes sure that intermediate nodes forward high priority packets (first pass image layer) faster than low priority packets (second pass image layer). This way, the routing framework facilitates sink nodes to reconstruct first pass image much sooner than when the entire image data is received at sink. As required, the sink node can add the second pass information to the first pass to construct a more detailed image.

IV. PRIORITY-BASED ROUTING FRAMEWORK FOR VSN

This section provides detail of how the priority-based routing framework works. The framework is distributed into network layer and medium access control layer of any protocol stack. Additionally, a thin Application Interface Layer (AIL) encapsulates the details of network layer and medium access control layer. Functional details of these layers are provided in the sub-sections below.

A. Application Interface Layer

The AIL (Application Interface Layer) is the application layer component of priority-based routing framework. It is a very thin layer that provides VSN application with a set of primitives that can be used for fragmenting image data into packets, sending them, receiving them and assembling them to re-generate image data. The AIL hides the implementation details of the entire framework. The VSN application passes image data along with its priority to the routing framework using the AIL. Based on its configuration, AIL of the sending node fragments the image data into packets of size that network layer can send. AIL also inserts image number and packet fragment number into the packet. This information is used by the AIL of sink node to join the fragments to construct image data sent.

B. Network Layer

The network layer component of priority-based routing framework works in two phases explained below.

1) Network Configuration Phase

When the VSN is deployed and brought up, the VSN nodes send advertisements to their neighbors declaring identities and their number of hops from sink. These advertisements are sent periodically. Initially all nodes are configured as being infinitely away from sink node. When sink node advertises, it declare its number of hops from sink as 0. The nodes receiving this advertisement add the respective sink node to their routing tables and mark their number of hops from sink as one hop. Now when such a node sends out its own advertisement, it declares its number of hops from the sink instead of infinity. The nodes at multiple hops from sink update their routing table with sink address along with the addresses of their neighbor as next hop address from who they received the advertisement. When a node receives advertisement of a sink from more than one neighbor, it keeps only the neighbor with lesser hops to the sink in its routing table. After a number of cycles of advertising, depending on the number of VSN nodes, the network is established. Each node knows the number of hops to the sink as well as the next hop towards the sink. As the advertisements are sent out periodically, removal and addition of nodes to the network is possible dynamically. Moreover, for maintenance of routing tables, each node keeps track of live neighbors using a watchdog timer associated with each neighbor.

2) Network Operation Phase

Once the network has been established, our routing framework is ready to transport image data from camera nodes to sink nodes. When the VSN application has image data to send, it uses primitives provided by the AIL from previous section. The network layer selects the next hop towards the sink that is selected by the camera node from its routing table. If the sink address as specified by the camera node is not in the routing table, the packet is dropped. A neighbor's entry keep-alive watchdog is reset whenever a packet is received from that neighbor. If a packet is not received from a neighbor within a threshold, the neighbor's entry is deleted from the routing table. In this way, routing tables are maintained during data transmission phase.

C. Medium Access and Control Layer

At the MAC layer, the routing framework works at two levels. The first is the intra-node level where the routing framework makes sure that high priority packets are forwarded before low priority packets. The second level is the inter-node level where the routing framework makes sure that when two neighbors contest for transmission medium, the neighbor with high priority packet gets a chance to transmit its packet before the neighbor with low priority packet. The following sub-sections explain these two levels.

1) Queue Insertion

When a packet arrives at MAC layer for transmission, it is sent instantaneously if the MAC layer is not already receiving or sending a packet. If the MAC layer is busy, the packet is placed in a queue where it waits for its turn. Our priority-based routing framework makes use of this queue. When a packet with high priority arrives, it is placed at the head of the queue so that it is sent in the next go. If a packet of low priority arrives, it is placed at the tail of the queue. As the MAC layer always selects packets from head of the queue for transmission, it is made sure that at intra-node level a packet with higher priority is transmitted first.

2) Differentiated Back-off Window

When two nodes find the medium available and transmit at the same time, a collision occurs. In regular CSMA/CD, both nodes back off for a randomly selected time slot from a pseudo-fixed-size window. If they collide again, the window size is increased exponentially to a certain size. The priority-based routing framework maintains different windows for the different priorities. When a collision occurs, the MAC layer checks the priority of packet that collided and determines back-off times from different windows. For high priority packet, the window is smaller than for a low priority packet. This way, if the node with high priority packet gets a chance to transmit its packet within a smaller window than a node with a low priority packet. This makes sure that at the inter-node level, high priority packets transmit sooner than low priority packets.

V. PROPOSED PROTOCOL IMPLEMENTATION

To quantify the usefulness of the routing framework, a VSN application was created and simulations were run. Contiki OS [24], an open source operating system for devices such as wireless sensor network nodes, was used to implement the routing framework. Modifications were made to MAC and network layer of RIME protocol stack [25] part of Contiki OS. RIME protocol stack provides a set of basic communication primitives ranging from best-effort single-hop broadcast and best-effort single-hop unicast, to best-effort network flooding and hop-by-hop reliable multi-hop unicast. The RIME protocol stack provides multiple options for each protocol layer. The configuration of RIME used for routing framework implementation consists of hop-by-hop reliable multi-hop unicast with a user-defined network layer, CSMA/CD as MAC layer and ContikiMAC [26] as Radio Duty Cycling layer. Modifications made to each layer of RIME protocol stack of Contiki OS are explained in the sub-sections below.

A. Modifications in RIME Network Layer

The custom network layer contains a periodic timer that expires half a second. Whenever the timer expires, a node sends out an advertisement. These periodic advertisements from each node help build routing tables as explained in the previous section. A network packet in RIME protocol stack is 128 bytes long. 24 bytes of this packet are used by RIME for header and remaining 104 byes are available as payload. When used as an advertisement, the payload contains addresses of sink nodes and their corresponding hops count from the node announcing the advertisement.

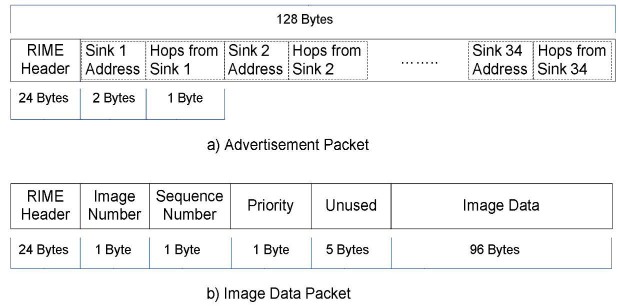


Figure 2. Types of VSN Packets

When a visual sensing node has a packet to send, it uses AIL send primitive to send it. The send primitive of AIL takes image layer, address of sink node and priority of the layer. The AIL fragments the image layer into packets. AIL also insert the image number and fragment number or packet sequence number into the data packet along with 96 bytes of image data. The image number and packet sequence number are used at the sink to reconstruct the image layer. Both advertisement and data packets are depicted in Fig. 2.

When a packet is received at the network layer of an intermediate node from a neighbor, it is checked if the packet is for the node itself or it is an image data packet that needs to be routed to some sink. In case if the packet is to be routed to the sink, the next hop is determined from the routing table that maintains next hop addresses corresponding sinks address. The neighbor is chosen as the next hop whose number of hops from sink is least. The data packet is then sent to that neighbor so that it can forward the packet to the sink or next hop towards the sink.

B. Modifications in RIME MAC Layer

The RIME MAC layer chosen for implementation of routing framework is CSMA/CD [27]. It contains a queue to store packets waiting for their turn for transmission. Modifications have been made to how a packet will be inserted into the queue. When the packet is received by MAC layer from network layer, the priority of the packet is checked. If it is a high priority packet, it is placed at the head of the queue. If it is a low priority packet, it is placed at the tail of the queue. When sending a packet, the MAC layer always picks up a packet from the head of the queue. This way if there is any high priority packet in the queue, it will be transmitted before low priority packets giving precedence to first pass image information at intra-node level.

$$E_c = \left(\frac{2^c - 1}{2} \right) \quad (1)$$

c is the number of times the packet collided

$$E_{c,p} = \left(\frac{2^{c(1+pW)} - 1}{2} \right) \quad (2)$$

$$p = \begin{cases} 0 & \text{if high priority packet collided} \\ 1 & \text{if low priority packet collided} \end{cases}$$

W is the contention window size

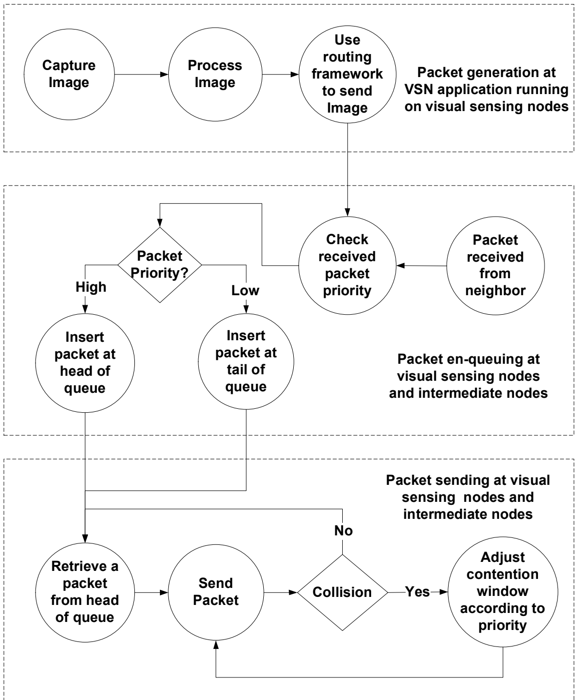


Figure 3. Routing Framework Data Flow

On sensing the medium to be free, if two nodes transmit at the same instance, a collision will occur. When this collision is detected by the MAC layer, it defers the transmission of that packet based on a random time slot out of a pseudo-fixed-sized contention window. The random time slot is selected using binary-exponential back-off algorithm. Without our modifications, the back-off algorithm maintains the same contention window for all types of packets that collide. The expected back-off time, *E(c)*, can be approximated using (1). We introduce a factor *pW* that enhances the back-off time calculation for packets of different priority levels. The factor *pW* in (2) causes contention window to shift for low priority packets, providing inter-node level precedence to high priority packets.

The flow of image data through the modified RIME stack is depicted in Fig. 3. The topmost block represents the VSN application and its usage of AIL. The middle block represents packet en-queueing into MAC layer transmission queue. The bottom block signifies the transmission of packet and calculation of contention window in case of collision.

The type of VSN applications targeted in this paper can be implemented using low-cost sensor network node such as TelosB [28]. Some motes can be equipped with CMUCam4 [29] giving them image capturing ability. The remaining TelosB nodes can be used to route image data from camera nodes to sink nodes. These applications of such VSNs can capture images and use image encoding algorithms such Discrete Cosine Transform [2] or Discrete Wavelet Transform [3][4] to encode images into different level of details for progressive image transmission. In the future, we intend to implement our proposed routing framework with real VSN application to measure its performance.

VI. Simulation Results

For simulations, we created an application that emulates a real VSN application by generating random image layers according to user-defined configurations. The simulation configurations set to quantify the usefulness of routing framework consist of generating 90 x 90 pixels resolution image layers where each pixel is of 3 bytes, 1 byte per color. Therefore the entire image layer is 90 x 90 x 3 bytes (24 Kilobytes, approximately). One data packet can transport 96 bytes hence one image layer is transmitted in less than 256 packets. The ratio of high priority to low priority packets is kept as 50-50%. The size of MAC layer queue is set to 32 packets. The simulations consist of 25 VSN nodes arranged in a regular grid, as depicted in Fig. 4. The channel check rate is set to 64, i.e., in one second the ContikiMAC radio duty-cycling layer checks the channel 64 times to see if a neighbor is transmitting. The dotted-line represents the transmission-reception ranges. The dot-filled circles represent sink node. The empty circles represent intermediate nodes. The circles with stripes denote visual sensing nodes.

The nodes at the corner of the grid have only two neighbors in their transmission-reception range, e.g., Node-20 and Node-24 are in vicinity of Node-25. Nodes on the side have three nodes in their vicinity, e.g., Node-22, Node-18 and Node-24 are in transmission-reception range of Node-23. Finally, remaining nodes of the grid have 4 neighbors in their vicinity, e.g., Node-12, Node-8, Node-14 and Node 18 are in vicinity of Node-13.

The application can emulate different scenarios by modifying simulation configurations. The camera nodes generate packets varying from 1 to 32 packets per second.

Three network configurations, depending on the number of visual sensing nodes, have been tested with a large number of simulations for each configuration. Node-1 was selected as sink in all simulations. For each network configuration, packets were generated at rates starting from 1 packet per second to 24 packets per second. Results of each configuration are given in the below.

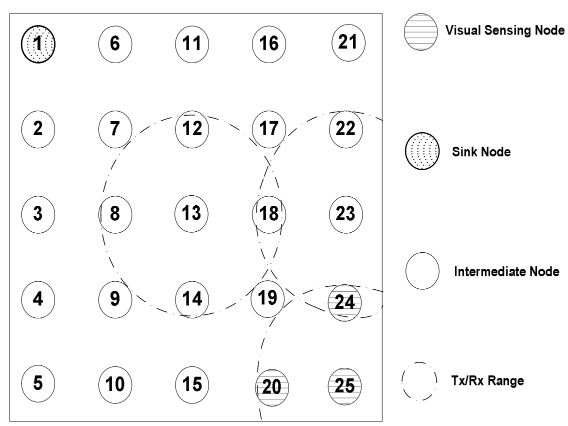


Figure 4. Grid Topology

The first network configuration contains one visual sensing node, Node-25, responsible for generating image layers. It is placed at 8 hops from the sink. Fig. 5 shows the average time taken by high priority packets and low priority packets to reach the sink node from visual sensing node. The lines represent average time taken with our proposed routing framework in place as compared to average time taken without our framework. The lines with circle and square symbols denote average transmission times of high priority and low priority packets, respectively, with routing framework inactive. Simulations were carried out without the routing framework in place to generate reference results. As the routing framework is not managing MAC queues and retransmission times of packets, there is no difference in routing of high and low priority packets. Both types of packets are treated the same way by the network. As a result both high and low priority packets take almost same time to reach the sink node. This is why circle symbols are not clearly visible in Fig. 5.

With the priority based routing framework actively managing MAC queues and retransmission times, high priority packets (denoted by line with triangles) take much lesser time than low priority packets (denoted by line with crosses). The legend for all figures has been kept similar to Fig. 5 for easy comparison by the reader. At lower packet generation rates the difference in average transmission times is less visible because the MAC layer queues are almost empty. Moreover, as each node has lesser packets to transmit, collisions rarely occur. As the packet generation rate is increased, the effect of routing framework becomes visible. The average transmission time for high priority packets decreases significantly as compared to the reference simulations. On the same note, average transmission times for low priority packets have increased as compared to the reference simulations.

Fig. 6 represents packet delivery ratios with and without our proposed routing framework in place at the 30 seconds deadline. Packet delivery ratio denotes the ratio of packets generated from the visual sensing nodes to packets received at the sink. At low packet generation rates, the difference in packet delivery ratios is less visible because the MAC layer queues are almost empty and as each node has lesser packets to transmit, resulting in rare cases of collisions. As the packet generation rate increases delivery ratio of high priority packets improves as compared to low priority packets. Moreover, delivery ratio of high priority packets is better than reference graphs when routing framework was inactive.

Fig. 7 represents the packets received over percentage of simulation time with and without our proposed routing framework in place. Without our framework, the number of packets received over simulation time is same for both high and low priority packets. With our framework, the number of high priority packets received is higher than number of low priority packets received. Hence, at any time in the simulation, the sink node receives more high priority packets although the packet generation rate has been kept same for both types of packets in our simulations.

To reconstruct the image at the sink node within a certain time, the image decoding algorithms running on the sink node impose deadlines for each layer. As the image encoding and decoding algorithms are not part of this paper, we have selected a deadline of 10 seconds for high priority packets corresponding to coarse image information of first pass and a deadline of 30 seconds for fine image information of second pass. In a real VSN application, these deadlines will be dependent on the image decoding algorithm. Fig. 8 represents the packet delivery ratio within these deadlines. With our proposed routing framework in place, the delivery ratio of high priority packets that reached the sink node within 10s seconds of transmission is significantly higher than without the routing framework active. With the routing framework active, the delivery ratio of low priority packets decrease as the packet generation rate increases. This decrease is due to the increase in delivery ratio of high priority packets. As the network resources remain same, the increase in packet delivery ratio of high priority packets is compensated with decrease in delivery ratio of low priority packets.

The second network configuration contains two visual sensing nodes, Node-20 and Node-24, both placed at 7 hops from the sink. Whereas the third network configuration contains three visual sensing nodes, Node-20, Node-24 and Node-25. Figs. 9 - 12 represent average transmission times, packet delivery ratios at 30 seconds simulation deadline, packets received over percentage simulation time and deadline based packet delivery ratios for two visual sensing nodes simulations, respectively. Similarly, Figs. 13 - 16 represent average transmission times, packet delivery ratios at 30 seconds simulation deadline, packets received over percentage simulation time and deadline based packet delivery ratios for three visual sensing nodes simulations, respectively. Simulations with two and three visual sensing nodes were carried out to see the effects of having more than one visual sensing node in the network.

As there is an overlapping between the paths from the visual sensing nodes to the sink node for two and three visual sensing nodes simulations, difference in average transmission times can be seen as compared to simulation results of one visual sensing node. This overlap increases the average transmission times for all packet generation rates as compared to simulations with one visual sensing node. Similarly, there is a difference in packet delivery ratios and packets received within deadlines as compared to simulation results of one visual sensing node.

The increase in average transmission times and the decrease in packet delivery ratios are because of two reasons. The first reason is that due to overlapping paths, packets collide. Collisions cause excessive retransmission. When the MAC layer's maximum retransmission threshold is achieved, the packet is discarded causing the packet delivery ratio to decrease. The packets that reach the sink take more time because of multiple retransmissions by the intermediate nodes causing the average transmission time to increase. The second reason is that as collisions increase, the lifetime of packet in the MAC layer queue also increases. This causes the queue to fill up sooner. As a result incoming packets do not find space in MAC layer queue and are dropped causing packet delivery ratio to decrease.

 ISBN: 978-1-61208-320-9

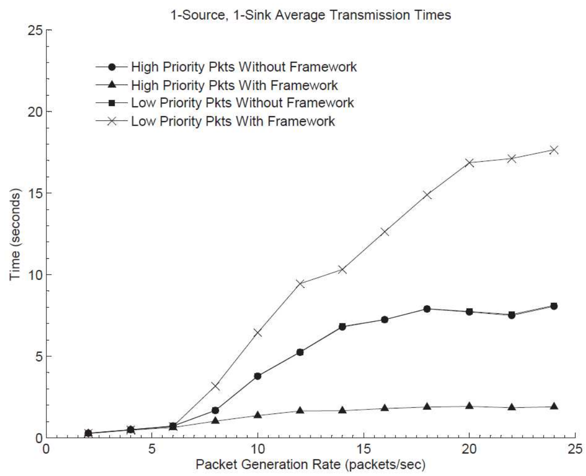


Figure 5. Average Transmission Times for 1 Source

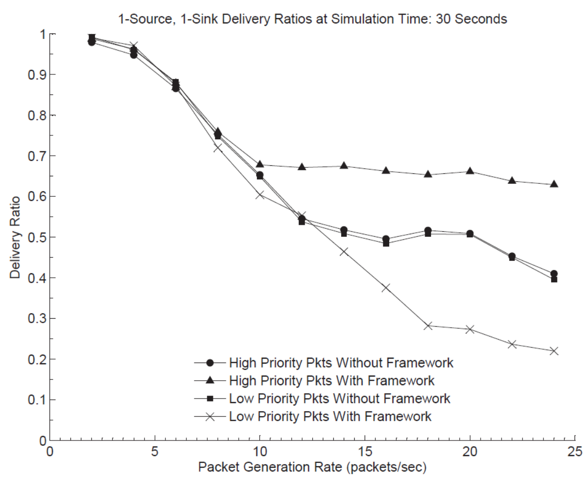


Figure 6. Delivery Ratios for 1 Source at Time: 30s

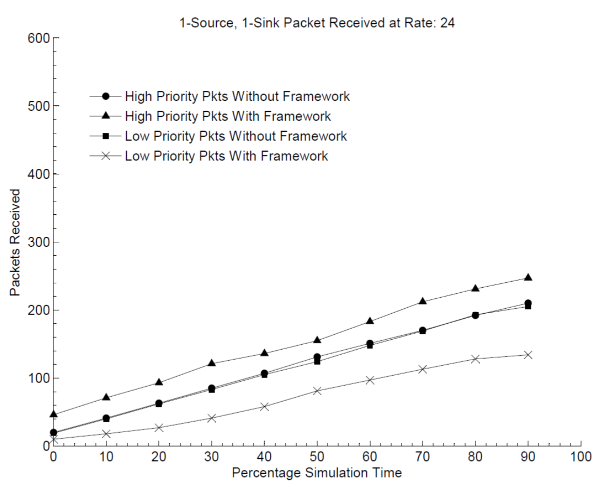


Figure 7. Packets Received for 1 Source over Simulation Time

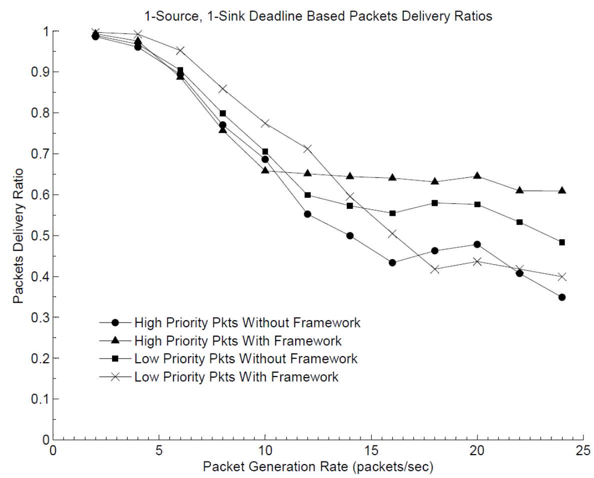


Figure 8. Delivery Ratios for 1 Source within Deadlines

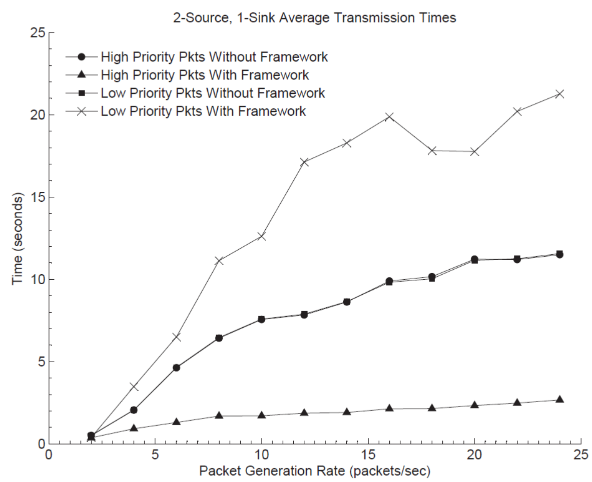


Figure 9. Average Transmission Times for 2 Sources

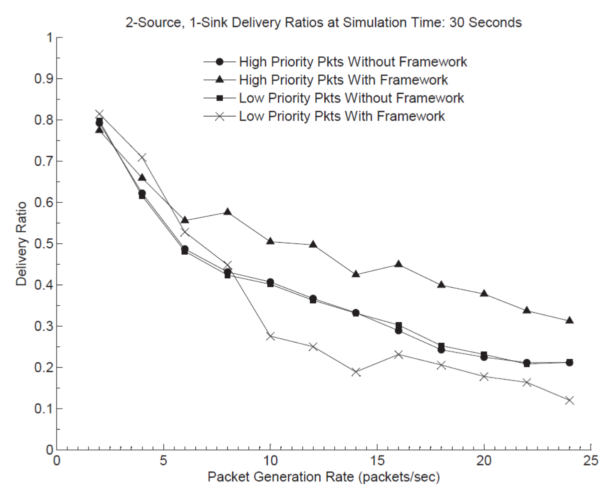


Figure 10. Delivery Ratios for 2 Sources at Time: 30s

 ISBN: 978-1-61208-320-9

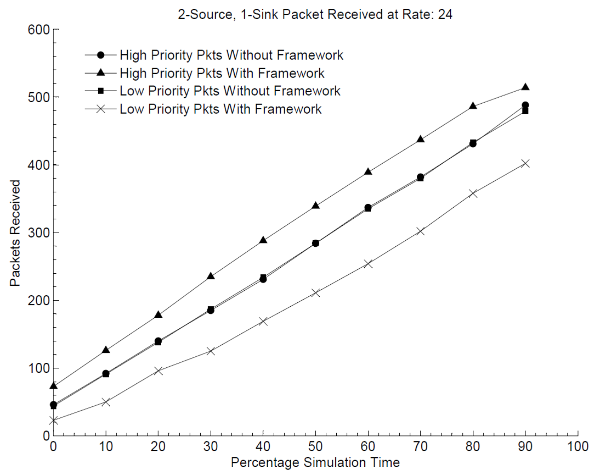


Figure 11. Packets Received for 2 Sources over Simulation Time

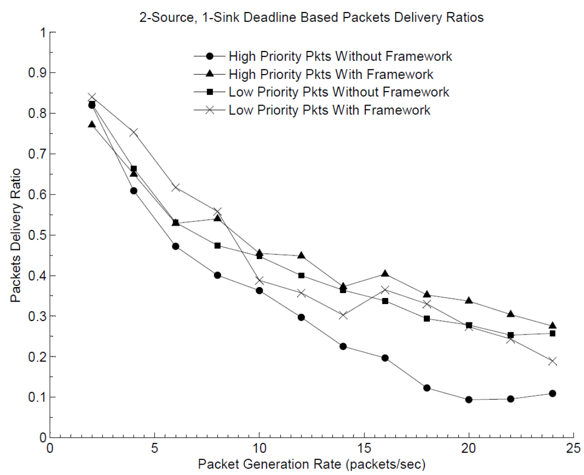


Figure 12. Delivery Ratios for 2 Sources within Deadlines

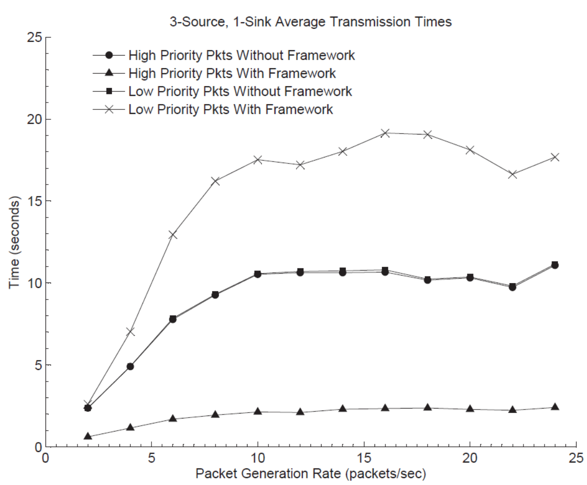


Figure 13. Average Transmission Times for 3 Sources

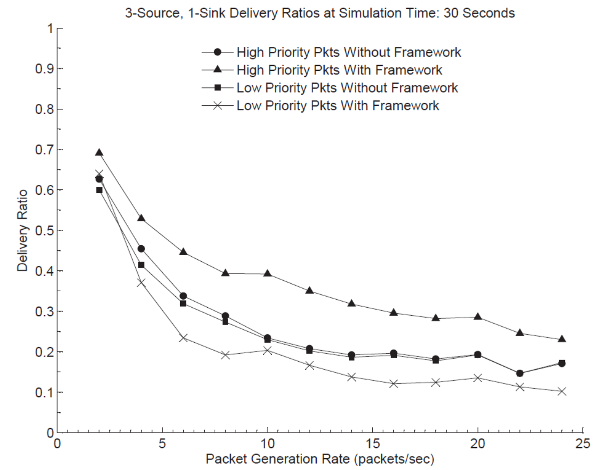


Figure 14. Delivery Ratios for 3 Sources at Time: 30s

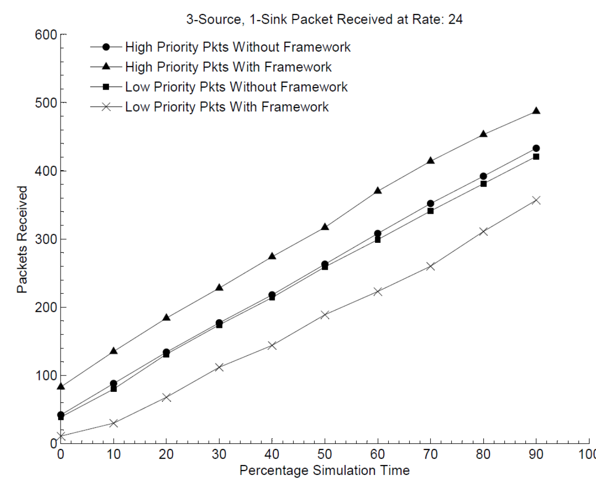


Figure 15. Packets Received for 3 Sources over Simulation Time

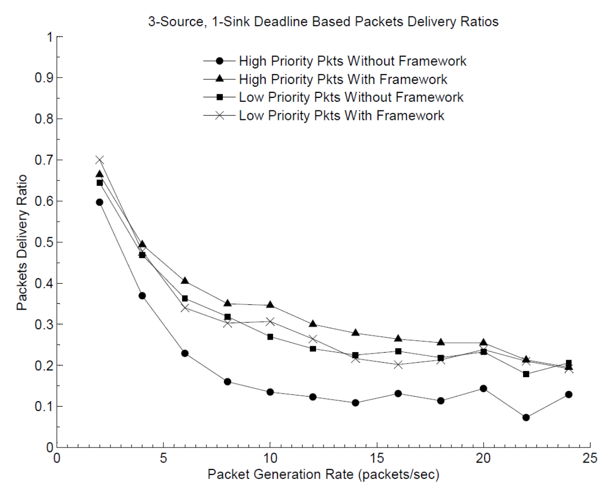


Figure 16. Delivery Ratios for 3 Sources within Deadlines

Hence, we prove that our framework improves system's response time in certain VSN applications.

VII. Conclusion and Future Work

Based on simulation results, we can conclude that our proposed priority-based routing framework assists progressive image transmission in VSNs. Critical imagery information from visual sensing nodes can be received at sink nodes sooner than less critical imagery information. However, there are areas of priority-based routing framework that can be improved. In the future, the authors of this paper intend to integrate this priority-based routing framework with an image encoding/decoding mechanism to measure the performance on a complete VSN platform.

Acknowledgment

This research is funded by the Institute of Consulting Research and Studies, Umm Al-Qura University, Makkah, Saudi Arabia, Grant No. S2011-9.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," Computer Networks, vol. 51, 2007, pp. 921–960

[2] Y. Huang, H. M. Dreizen, and N. P. Galatsanos, "Prioritized dct for compression and progressive transmission of images," Image Processing Transactions, vol. 1, no. 4, Oct. 1992, pp. 477–487, doi:10.1109/83.199917

[3] K. H. Talukder and K. Harada, "Enhancement of discrete wavelet transform (dwt) for image transmission over internet," in Proceedings of the 8th International Conference on Information Technology: New Generations, (ITNG '11), IEEE Computer Society, 2011, pp. 1054–1055, doi:10.1109/ITNG.2011.184

[4] F. Behnamfar, F. Alajaji, and T. Linder, "Progressive image communication over binary channels with additive bursty noise," in Proceedings of the Data Compression Conference, (DCC '02), IEEE Computer Society, 2002

[5] K. Zuberi and K. Shin, "Design and implementation of efficient message scheduling for controller area network," IEEE Transactions on Computers, vol. 49, no. 2, Feb 2000, pp. 182 –188.

[6] S.-K. Kweon and K. G. Shin, "Providing deterministic delay guarantees in atm networks," IEEE/ACM Trans. Netw., vol. 6, no. 6, pp. 838–850, Dec. 1998, doi:10.1109/90.748093

[7] C. Li, R. Bettati, and W. Zhao, "Static priority scheduling for atm networks," in Proceedings of the 18th IEEE Real-Time Systems Symposium, ser. RTSS '97. Washington, DC, USA: IEEE Computer Society, 1997

[8] W. Zhao, J. Stankovic, and K. Ramamritham, "A window protocol for transmission of time-constrained messages," IEEE Transactions on Computers, vol. 39, no. 9, 1990, pp. 1186–1203.

[9] D. Kandlur, K. Shin, and D. Ferrari, "Real-time communication in multi-hop networks," in 11th International Conference on Distributed Computing Systems, May 1991, pp. 300 –307.

[10] L. Chen and W. B. Heinzelman, "A survey of routing protocols that support qos in mobile ad hoc networks," Network Magazine of Global Internetworking, vol. 21, no. 6, Nov. 2007, pp. 30–38, doi:10.1109/MNET.2007.4395108

[11] L. Hanzo-II and R. Tafazolli, "A survey of qos routing solutions for mobile ad hoc networks," IEEE Commun. Sur. Tuts., vol. 9, no. 2, Apr. 2007, pp. 50–70, doi:10.1109/COMST.2007.382407

[12] J. Liebeherr, D. Wrege, and D. Ferrari, "Exact admission control for networks with a bounded delay service," IEEE/ACM Transactions on Networking, vol. 4, no. 6, Dec 1996, pp. 885 –901

[13] S. Wang, D. Xuan, R. Bettati, and W. Zhao, "Providing absolute differentiated services with statistical guarantees in static-priority scheduling networks," in proceedings of Seventh IEEE Real-Time Technology and Applications Symposium, 2001, pp. 127 –129.

[14] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in proceedings of ACM Applications, technologies, architectures, and protocols for computer communication, (SIGCOMM '99), New York, NY, USA: ACM, 1999, pp. 81–94,.doi:10.1145/316188.316208

[15] O. Younis and S. Fahmy, "Constraint-based routing in the internet: Basic principles and recent research," IEEE Comm. Surveys Tuts, vol. 5, no. 1, Jul. 2003, pp. 2–13, doi:10.1109/COMST.2003.5342226

[16] J. Soldatos, E. Vayias, and G. Kormentzas, "On the building blocks of quality of service in heterogeneous ip networks," Commun. Surveys Tutorials., vol. 7, no. 1, Jan. 2005, pp. 69–88, doi:10.1109/COMST.2005.1423335

[17] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network," IEEE Personal Communications, vol. 7, no. 5, Oct 2000, pp. 16 –27

[18] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. StankoSvic, and T. He, "Rap: A real-time communication architecture for large-scale wireless sensor networks," Charlottesville, VA, USA, Technical Report, 2002.

[19] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "Speed: a stateless protocol for real-time communication in sensor networks," in proceedings of 23rd International Conference on Distributed Computing Systems, May 2003, pp. 46 – 55.

[20] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in proceedings of 14th IEEE International Workshop on Quality of Service, (IWQoS 2006), June 2006, pp. 83 –92.

[21] E. Felemban, C. Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Gurantee of Reliability and Timeliness in Wireless Sensor Networks," IEEE transaction on Mobile Computing, vol. 5, no. 6, June 2006.

[22] T. Fang, C. Fu, B. Falkowski, and B. Wang, "Multiple dynamic range image coding for wireless sensor networks," in proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2008. (SMC 2008), Oct. 2008, pp. 2944 –2949.

[23] E. Manhas, G. Brante, R. Souza, and M. Pellenz, "Energy-efficient cooperative image transmission over wireless sensor networks," in proceedings of Wireless Communications and Networking Conference (WCNC), 2012, pp. 2014 –2019.

[24] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, (LCN '04). pp. 455–462, 2004, doi:10.1109/LCN.2004.38

[25] A. Dunkels, F. O¨ sterlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in proceedings of the 5th ACM International conference on Embedded networked sensor systems, (SenSys '07)., 2007, pp. 335–349. doi:10.1145/1322263.1322295

[26] A. Dunkels, "The contikimac radio duty cycling protocol," Swedish Institute of Computer Science, Technical Report, 2011.

[27] Farooq, M.O., Kunz, T., "Contiki-based IEEE 802.15.4 node's throughput and wireless channel utilization analysis," Wireless Days (WD), 2012 IFIP , vol., no., pp.1,3, 21-23 Nov. 2012 doi: 10.1109/WD.2012.6402828

[28] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in proceedings of the 4th IEEE international symposium on information processing in sensor networks, (IPSN '05), Piscataway, NJ, USA, 2005

[29] L. Surhone, M. Tennoe, and S. Henssonow, "CMU-CAM", Betascript Publishing, 2010. [Online]. Available: http://books.google.com.sa/books?id=0EBfYgEACAAJ, Date Visited: 19-Dec-2013