FUZZY LOGIC BASED CONGESTION CONTROL FOR PROGRESSIVE IMAGE TRANSMISSION OVER WIRELESS SENSOR NETWORKS

by

Çağatay Sönmez

B.S., Computer Engineering, Dokuz Eylül University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2012

FUZZY LOGIC BASED CONGESTION CONTROL FOR PROGRESSIVE IMAGE TRANSMISSION OVER WIRELESS SENSOR NETWORKS

APPROVED BY:

Prof. Cem Ersoy (Thesis Supervisor)	.....
Assoc. Prof. Fatih Alagöz	.....
Prof. Hakan Deliç	.....

DATE OF APPROVAL: 29.05.2012

ACKNOWLEDGEMENTS

Firstly, I would like to give my sincere thanks to my advisor Prof. Cem Ersoy for providing an opportunity to work under his guidance. I am very grateful for his academic guidance, for teaching opportunities, for encouraging me to pursue my master's degree and for a well-organized inspiring research environment. His positive outlook and confidence in my research inspired me and gave me confidence.

I thank all of the members of the thesis jury, Assoc. Prof. Fatih Alagöz and Prof. Hakan Deliç for their time valuable comments for making this thesis better.

I deeply thank Sinan Işık, M. Yunus Dönmez and Ozlem Durmaz İncel for their help, advices and guidance during my research; their discussions, ideas, and feedbacks have been absolutely invaluable. Especially during the implementation of my thesis, Sinan and Yunus spent their time to share their knowledge and expertise in OPNET.

I would like to thank M. Aykut Yiğitel for sharing the source code of Diff-MAC which is used in implementation of my thesis. I thank him for his valuable help. I also want to thank all my friends in NETLAB for always giving me their feedbacks during the discussions of different ideas and thanks to all other members of CMPE family that we spent cheerful moments with.

Finally I would like to thank my team leader Hürriyet Keskin Cum for her understanding during the time when I was out of office and for her reviews.

This work is supported by the European Community's Seventh Framework Programme (FP7-ENV-2009-1) under the grant agreement FP7-ENV-244088 FIRESENSE, by COST Action COST IC0906 WiNeMO Wireless Networking for Moving Objects and by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

ABSTRACT

FUZZY LOGIC BASED CONGESTION CONTROL FOR PROGRESSIVE IMAGE TRANSMISSION OVER WIRELESS SENSOR NETWORKS

The availability of low-cost camera hardware has enabled the Wireless Multimedia Sensor Networks (WMSNs), and has made it possible to develop multimedia applications such as surveillance systems, traffic control systems, environment monitoring and object tracking. Unlike scalar data, multimedia requires higher bandwidth due to the larger amount of information to be transmitted. Congestion is a challenging problem for Wireless Sensor Networks (WSNs) due to waste of scarce communication resources and reduction energy efficiency. Compared to traditional WSNs, the probability of congestion occurrence in WMSNs is higher due to the high volume of data arising from multimedia streaming. In this thesis, problems for multimedia transmission over WMSNs are examined and Sensor fUzzy-based Image Transmission (SUIT); a new cross-layer progressive image transport protocol is proposed as a solution. SUIT provides fuzzy logic based congestion estimation and an efficient congestion mitigation technique which decreases the image quality on-the-fly to an acceptable level. In case of congestion, SUIT drops some packets of the frames in a smart way and thus transmit frames to the sink with lower, but acceptable quality. In this way, SUIT improves the continuity of the video streaming. The performance of SUIT is evaluated by comparing it with two example transport protocols proposed for WSNs. According to the simulation results, SUIT provides better energy consumption, frame delivery, frame loss and frame latency performance than its competitors.

ÖZET

ÇOKLUORTAM KABLOSUZ ALGILAYICI AĞLARDA AŞAMALI GÖRÜNTÜ GÖNDERİMİ İÇİN BULANIK MANTIK İLE SIKIŞIKLIK KONTROLÜ

Telsiz Çoklu Ortam Algılayıcı Ağlar (TÇAA) ucuz kamera donanımlarının ortaya çıkmasıyla önem kazanmıştır. TÇAA lara karşı artan bu ilgi, gözetleme ve nesne takibi gibi çoklu ortam uygulamalarının geliştirilmesine imkan sağlamıştır. Çoklu ortam verisi taşıdığı veri miktarından dolayı skalar veriden daha yüksek bant genişliğine gereksinim duyar. Tıkanıklık, gereksiz iletişime neden olup enerji verimliliğini azalttığından telsiz algılayıcı ağlar (TAA) için zorlayıcı bir problemdir. Çoklu ortam akışından kaynaklı fazla sayıdaki veriler TÇAA larda tıkanıklığın oluşma ihtimalini klasik TAA göre daha yüksek kılar. Bu tezde, TÇAA üzerinden çoklu ortam verisi taşırken oluşan problemler incelenmektedir ve SUIT adında tıkanıklık kontrol mekanizması sağlayan yeni bir çapraz katman kademeli resim taşıma protokolü önerilmektedir. SUIT bulanık mantık tabanlı tıkanıklık tespit etme yöntemi ve resimlerin algılayıcılar üzerinden ilerlerken kalitelerini kabul edilebilir derecede düşüren etkin bir tıkanıklık hafifletme yöntemi sağlamaktadır. Tıkanıklığın oluştuğu durumlarda SUIT resim çerçevelerinin bazı paketlerini akıllıca düşürerek, çerçeveleri daha düşük fakat kabul edilebilir kalitede iletir. Bu yüzden SUIT daha iyi çoklu ortam iletim başarımı sağlayarak video akışının sürekliliğini artırır. SUIT in başarımı TAA lar için geliştirilmiş iki farklı taşıma protokolü ile değerlendirilmiştir. Benzetim sonuçlarına göre, SUIT rakiplerinden daha iyi enerji tüketim, resim çerçevesi taşıma, resim çerçevesi kaybetme ve gecikme başarımı sağlamıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. LITERATURE REVIEW . . . 5
2.1. Congestion Control Protocols . . . 5
2.1.1. CODA . . . 6
2.1.2. SenTCP . . . 7
2.1.3. Fusion . . . 7
2.1.4. CCF . . . 8
2.1.5. Siphon . . . 9
2.1.6. FTE . . . 10
2.1.7. FCE . . . 10
2.2. Reliable Transport Protocols . . . 11
2.2.1. DTSN . . . 12
2.2.2. RSTP . . . 12
2.2.3. PSFQ . . . 13
2.2.4. RMST . . . 14
2.3. Reliable Transport Protocol Providing Congestion Control . . . 14
2.3.1. RCRT . . . 14
2.3.2. ESRT . . . 15
2.3.3. STCP . . . 16
2.4. Summary . . . 16
3. ISSUES FOR MULTIMEDIA STREAMING IN WMSNs . . . 19
3.1. Application Layer Issues (Video Encoding) . . . 20

3.2. Transport Layer Issues 22
3.3. Routing Layer Issues 25
3.4. MAC Layer Issues 27
3.5. Cross Layer Approach 28
3.6. Image Transmission Over WMSNs 29
3.6.1. Sequential DCT based JPEG 30
3.6.2. Progressive DCT based JPEG 31
3.6.3. DCT based JPEG with Restart Markers 33
3.7. Summary 34
4. SENSOR FUZZY-BASED IMAGE TRANSPORT (SUIT) 35
4.1. Connectionless Communication 37
4.2. Packet Ordering 38
4.3. Packet Prioritization 39
4.4. Cross-Layer Data Exchange among Layers 42
5. CONGESTION CONTROL MECHANISM FOR SUIT 44
5.1. Fuzzy Logic Based Congestion Detection 44
5.2. Congestion Mitigation 51
6. PERFORMANCE EVALUATION OF SUIT 55
6.1. System Model and Simulation Parameters 55
6.2. Performance Evaluation of JPEG Encoding Techniques 62
6.3. Performance Evaluation of Congestion Mitigation Techniques 68
6.4. Comparison of Images Transmission with SUIT and Constant Rate . . 76
7. CONCLUSIONS 81
REFERENCES 84

LIST OF FIGURES

Figure 3.1. Packet loss effects on progressive JPEG. 32

Figure 3.2. Packet loss effects on sequential JPEG with restart marker. 33

Figure 4.1. SUIT's out of order packet handling. 38

Figure 4.2. Buffer management in SUIT. 41

Figure 4.3. Cross-layer design of SUIT. 43

Figure 5.1. Fuzzy inference system of SUIT. 46

Figure 5.2. Membership functions for the fuzzy variables α, β and γ. 48

Figure 5.3. Membership functions of the consequence classes. 51

Figure 5.4. Quality reduction effects on file size and PSNR. 53

Figure 5.5. Operation of congestion mitigation algorithm. 54

Figure 6.1. Camera detection model, taken from [1]. 56

Figure 6.2. Segments of the JPEG images. 58

Figure 6.3. The average received frame rate at the sink. 63

Figure 6.4. Average frame delivery ratio. 64

Figure 6.5. Average frame latency. 65

Figure 6.6. The average quality of received frames at the sink. 68

Figure 6.7. Average received frame rate at the sink. 70

Figure 6.8. Average frame loss. 71

Figure 6.9. The average source to the sink latency of frames. 72

Figure 6.10. Energy expenditure per sensor node. 73

Figure 6.11. Average quality of received frames. 74

Figure 6.12. Average frame rate at source nodes. 75

Figure 6.13. Average received frame rate at the sink. 76

Figure 6.14. Average frame loss. 77

Figure 6.15. The average source to the sink latency of frames. 78

Figure 6.16. Energy expenditure per sensor node. 79

LIST OF TABLES

Table 2.1. Classification of the protocols based on congestion control. 17

Table 2.2. Classification of the protocols based on reliability. 17

Table 2.3. Classification of the protocols based on congestion control and reliability. 18

Table 5.1. Fuzzy rules in the knowledge base. 50

Table 6.1. Properties of the JPEG files used in the simulation. 57

Table 6.2. Simulation parameters for the evaluation of SUIT. 61

Table 6.3. PSNR values for PJPEG images. 66

Table 6.4. PSNR values for SJPEG-RM images. 67

Table 6.5. Fuzzy logic table of FCE. 70

LIST OF SYMBOLS

d_{AB}	Distance between A and B
d_t	Average delay threshold for class t, 1 $\leq$t$\leq$ 3
f	Original image used in PSNR calculation
f_1^r	The class of α in rule r
f_2^r	The class of β in rule r
f_3^r	The class of γ in rule r
f_c^r	The class of consequence in rule r
fps_{max}	Maximum frame rate
F_{th}	Frame index threshold
g	Transmitted image used in PSNR calculation
h_t	Hop count threshold for class t, 1 $\leq$t$\leq$ 3
K	Frame rate of the camera
L	Congestion level
r	Frame rate
r_1	Frame rate for NC level
r_2	Frame rate for SC level
r_3	Frame rate for FC level
T_{dwell}	Dwell time
$\vec{V}$	Velocity of the target
W_d	Weight for delay
W_f	Weight for frame index
W_h	Weight for hop count
α	Number of contender
α^f	Fuzzy vector obtained for α
β	Buffer occupancy of parent sensor
β^f	Fuzzy vector obtained for β
γ	Traffic load

γ_{in}	Number of incoming packet
γ_{out}	Number of outgoing packet
γ^f	Fuzzy vector obtained for γ
μ_c^t	Membership function of consequence for class t, $1 \leq t \leq 7$
μ_χ^H	Membership function of χ, where χ is α, β or γ, for class H
μ_χ^L	Membership function of χ, where χ is α, β or γ, for class L
μ_χ^M	Membership function of χ, where χ is α, β or γ, for class M
$\mu_c^{f_\alpha^r}$	Membership functions in the rule r for α
$\mu_c^{f_\beta^r}$	Membership functions in the rule r for β
$\mu_c^{f_\gamma^r}$	Membership functions in the rule r for γ
$\mu_c^{f_c^r}$	Membership functions in the rule r for consequence fuzzy classes
θ_r	Output value of rule r evaluated by REM
ω	Output of the centroid defuzzifier

LIST OF ACRONYMS/ABBREVIATIONS

ACK	Acknowledgment
AIMD	Additive Increase Multiplicative Decrease
CCF	Congestion control and fairness
CD	Congestion Detection
CN	Congestion Notification
CM	Congestion Mitigation
CODA	Congestion Detection and Avoidance
CTS	Clear-to-Send
DTSN	Distributed Transport for Sensor Networks
EAR	Explicit Acknowledgment Request
ESRT	Event-to-Sink Reliable Transport
E2E	End to End
FCE	Fuzzy Based Congestion Estimation
FEC	Forward Error Correction
FIN	Finish
FIS	fuzzy Inference System
FL	Fuzzy Logic
FLS	fuzzy Logic System
FoV	Field of View
FTE	Fuzzy Based Trust Estimation
GPS	Global Positioning System
H	High
HbH	Hob by Hop
IEEE	The Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JPEG	Joint Photographic Experts Group

L	Low
LN	Loss Notification
LR	Loss Recovery
M	Medium
MAC	Medium Access Control
MJPEG	Motion JPEG
MPEG	Moving Pictures Experts Group
NACK	Negative Acknowledgment
NBOC	Neighbor Buffer Occupancy Table
PSFQ	Pump Slowly Fetch Quickly
QoS	Quality of Service
RCRT	Rate-Controlled Reliable Transport
REM	Rule Evaluation Method
RMST	Reliable Multi-Segment Transport
ROI	Region of Interest
RSTP	Reliable Synchronous Transport Protocol
RTS	Request-to-Send
RTT	Round Trip Time
SMAC	Sensor Medium Access Control
STCP	Sensor Transmission Control Protocol
SUIT	Sensor fUzzy Image Transport
SYNC	Synchronization
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
VOD	Video on Demand
WSN	Wireless Sensor Networks
WMSN	Wireless Multimedia Sensor Networks

1. INTRODUCTION

A Wireless Sensor Network (WSN) consists of a large number of sensor nodes and center nodes called sink nodes. Sensor nodes are distributed over a large target area and can communicate with each other via their wireless transceivers. Sensor nodes are usually resource-constrained wireless devices in terms of energy, bandwidth, memory and processing power. They are generally small-sized devices with limited battery and they can sense phenomena such as motion, sound, heat, and light [2] via the embedded sensing boards. Compared to sensor nodes, sink nodes are much more powerful devices than the sensor nodes and they collect sensed data from the sensor nodes. Generally, WSNs have multihop star-tree topology organized around the sink node(s). Data can be transmitted either in one hop or in several hops to the sink depending on the deployment and transmission capabilities. Due to their ad-hoc communication capability and co-operative nature, very large areas can be covered by WSNs [3].

With the recent developments in sensor technology, typical sensor nodes are able to capture multimedia data via low-cost hardware such as CMOS cameras and microphones. Such low-cost devices which are capable of capturing multimedia information from the environment have made it possible to develop applications for Wireless Multimedia Sensor Networks (WMSNs) [4]. A WMSN is formed with a large number of distributed embedded devices which are equipped with camera modules. These devices are able to retrieve multimedia content from the environment and are able to extract video and audio streams, still images as well as the scalar sensor data from the multimedia content [5]. WMSNs have generated much interest in recent years due to their potential to enable a large class of applications such as surveillance systems, traffic control systems, environment monitoring, control of manufacturing processes in industry [6].

With respect to the characteristics of WMSNs, multimedia traffic produces bursty, high-load traffic in the network [3]. Therefore, WMSN applications require more band-

width than WSN applications. Most common application scenarios for wireless sensors are *periodic reporting*, and *event detection* applications [7]. Periodic reporting is the simplest scenario because it does not require too much bandwidth relative to the other scenarios. Moreover it does not have a time constraint while delivering the perceived information from the environment. Event detection is more challenging than periodic reporting because events can occur at any time randomly. In the latter scenario, the data type is important. Bursty data traffic arising from video transmissions has constraint of real time delivery and requires high bandwidth compared to other types of data traffic. As a consequence, WMSNs require different transmission techniques than the ones proposed for traditional WSNs.

Congestion is one of the major challenges for wireless networks especially with high-volume bursty data. It occurs when the nodes attempt to carry more data than it can support such that packet queues of the nodes are full and packets need to be dropped. Probability of congestion in WMSNs is higher than traditional WSNs due to the large amount of video traffic. In addition to the waste of communication and energy that it causes, congestion negatively affects reliability due to the packet losses and degrades the overall performance of the network and quality-of-service (QoS) of the application. Congestion control is a difficult task for wireless networks because identifying the occurrence of congestion is not as simple as in wired networks. Unlike wired networks, end-to-end congestion control approaches cannot be applied to WSNs due to the multihop network topology and high node density. Generally, congestion control protocols for WSNs consider specific parameters such as local buffer occupancy, packet arrival rate, and packet service time. However, deciding on the level of congestion based on a specific parameter may give incorrect results [8]. Instead of using a specific congestion indicator, using multiple indicators, such as a fuzzy-logic-based mechanism proposed in this thesis, can provide better congestion detection performance.

Implementation of multimedia coding techniques is another major challenge for WMSNs. Predictive video coding techniques such as H.263 [9] or MPEG (Moving Picture Experts Group) [10] family provide quite good compression. However, these coding techniques are asymmetric such that the encoder is more complex than the

decoder. They use a complex motion estimation algorithm in the encoding phase. Since sensor nodes have energy, memory, and computational-speed constraints, such encoding techniques are not appropriate for sensor devices. In order to overcome the encoder complexity, distributed source coding techniques are proposed by Slepian and Wolf [11], and Wyner and Ziv [12]. Distributed source coding provides less complex encoder than the decoder, but it is not standardized and not mature enough to be used in WMSN applications. Therefore a significant amount of the WMSN applications [13–16] are using JPEG (Joint Photographic Experts Group) [17] images due to its lower complexity and adequate compression capabilities.

Motivated by these discussed facts, in this thesis, we focus on the following question: "How much efficiency can be achieved in detecting and mitigating congestion in WMSNs by using a fuzzy-logic based congestion detection method and image quality adaptation?" To answer this question, we propose Sensor fUzzy-based Image Transport (SUIT) which is a new cross-layer progressive image transport protocol that provides fuzzy logic based congestion estimation and subsequent mitigation techniques. Main goal of the SUIT protocol is increasing the frame delivery performance by decreasing the quality while frames are flowing over the intermediate sensors. In case of congestion, instead of dropping the whole frame, SUIT reduces the frame quality (as well as the frame size) by dropping subframes of progressively encoded JPEG without corrupting the image file. Thus, SUIT transmits images to the sink with lower, but acceptable quality. In this way, SUIT improves the continuity of the video streaming. To the best of our knowledge, SUIT is the first transport protocol proposed for WMSNs which is able to mitigate congestion by adjusting image quality while image data are being transmitted.

Extensive simulations are performed to evaluate performance of SUIT's congestion control. For congested network cases, SUIT provides better performance than its competitors in all aspects while providing acceptable image quality. The following are the fundamental features of SUIT:

- SUIT is a cross-layer protocol which enables and utilizes information exchange

among application, transport, MAC and routing layers.

- SUIT is a connectionless protocol. It does not require the establishment of a connection before streaming video. Thus real time video data can be sent immediately from source to the sink.
- SUIT can detect congestion efficiently by using a fuzzy logic based congestion estimation method.
- SUIT proposes a novel congestion mitigation technique which can perform quality adaptation on-the-fly, as well as rate adaptation.
- SUIT supports both sequential and progressive JPEG [17] formats. However, it provides best image transmission performance while using progressively encoded JPEG images.

The organization of the thesis is as follows: In Chapter 2, literature review on transport protocols for wireless sensor networks is examined under three parts with respect to protocols which provide reliability, congestion and both. In Chapter 3, issues of multimedia streaming over WMSNs are explained in detail considering the different (application, transport, MAC) layers. In Chapter 4, design and architecture of SUIT are introduced. Chapter 5 explains SUIT's congestion control mechanism which is comprised of a fuzzy logic based congestion detection and a novel congestion mitigation technique. In Chapter 6, performance evaluation of the proposed SUIT protocol is performed with a real-life scenario. Chapter 7 concludes this thesis and provides possible directions for future research.

2. LITERATURE REVIEW

In the literature, there are many transport protocols proposed for WSNs. In this chapter, a brief overview of the literature on transport protocols for WSNs is provided. As we explain in Section 3.2, the main objectives of the transport layer in WMSNs are reliable transmission and congestion control. Thus the literature review is structured into three sections: the protocols which provide only congestion control, protocols that provide only reliable transport and protocols with both congestion control and reliable transport.

2.1. Congestion Control Protocols

Congestion is a primary factor that deteriorates the QoS. Generally it occurs when the sensor nodes carry too much data which causes packet loss and increases the queuing delay. In WSNs congestion control is one of the difficult objectives of the transport layer because the sensor nodes have limited computation power and battery. Moreover, there is not an exact congestion indicator available for WMSNs. In Chapter 5, these difficulties are explained in detail. Misra *et al.* [18] discuss the performance of protocols performing congestion control while multimedia streaming over WMSN. Basically, congestion control protocols should detect the congestion successfully, notify the associated sensor nodes and then apply the subsequent mitigation techniques. Congestion detection is the most difficult step of the congestion control. In WSNs, commonly used congestion indicators are local buffer occupancy [19–24], channel load [19, 21, 22], packet arrival rate [20, 25], and packet service time [26–28]. After detecting the congestion, other sensor nodes should be informed about it. Two common methods are used for this issue. Congestion can be informed explicitly or implicitly. Protocols which perform notification process explicitly send a message to relevant sensor nodes [19, 21]. Other protocols perform this step implicitly by inserting a congestion notification flag into the header of outgoing packets as a piggyback [20, 22]. Explicit notification brings an extra communication overhead to the network. Implicit notification requires fewer packet transmissions, but takes longer to effect. For con-

gestion mitigation, various methods are proposed. These methods are adapting data transmission rate [19, 21, 27, 29], redirecting data traffic to different sensors [22, 30], and using back-pressure techniques to inform neighbor sensor nodes not to forward data [31].

A number of transport protocols have been proposed and have been implemented to provide congestion control mechanism for WSNs. These studies differ in congestion detection, congestion notification, and the rate-adjustment mechanisms. In the following subsections, details of popular congestion control protocols are explained.

2.1.1. CODA

Congestion Detection and Avoidance (CODA) [19] proposes an energy efficient congestion control scheme for WSNs. It provides three mechanisms for detecting and mitigating congestion: (i) receiver-based congestion detection; (ii) open-loop hop-by-hop backpressure; and (iii) closed-loop multi-source regulation.

CODA detects congestion on the receiver side by monitoring the local buffer occupancy and measuring the local channel load, whic is calculated by tracing the channel busy time in carrier sense multiple access (CSMA) networks. In case of congestion, the receiver broadcasts a backpressure message to its neighbors. A Neighbor sensor node which receives the backpressure message can decrease the sending rate or drops packets based on the local congestion policy. CODA also uses a closed-loop multi-source regulation technique to adapt the event rate at all sources dynamically. According to this technique, if the event rate of the source node is less than the maximum theoretical throughput of the channel, the source node can regulate itself. Otherwise the source node requests constant feedback from the sink to maintain the event rate of all sensor nodes. The sink, receiving this request, sends a notification to all sensor nodes to regulate their event rate.

2.1.2. SenTCP

SenTCP [20] is an open-loop hop-by-hop congestion control protocol. It measures the buffer occupancy and packet inter-arrival time in every intermediate sensor node to calculate the congestion level. Unlike CODA, which uses channel load with buffer occupancy to estimate the congestion, SenTCP uses the mean packet inter-arrival time.

In case of congestion, intermediate sensor nodes produce a feedback signal to the backward direction periodically to perform the rate adaptation. The feedback signal includes the local congestion degree and the buffer occupancy ratio, and it is propagated in a hop-by-hop manner. The intermediate sensor node or the source sensor node which receives the feedback signal from the upstream neighboring nodes can increase or decrease its sending rate according to the defined congestion level in the feedback signal. Then, intermediate sensor nodes decide if they should relay the feedback signal backward continuously.

2.1.3. Fusion

Fusion [21] proposes congestion control and fairness by using three techniques: hop-by-hop flow control, rate limiting, and a prioritized MAC. It uses a congestion bit in the header of each outgoing packet to provide congestion feedback to all neighbor nodes.

Hop-by-hop flow control consists of two components which are congestion detection and congestion mitigation. Fusion detects congestion by monitoring a sensor node's queue size. If the queue size exceeds a threshold, the congestion bit of outgoing packets is set; otherwise it is cleared. Receiver node checks the congestion bit of packets coming from the next-hop sensor node depending on the routing path. If the congestion bit is set, it stops forwarding data to support the next-hop sensor node to empty its queues.

Fusion method mitigates the congestion by rate limiting and prioritizing the

packets at the MAC layer. In the rate limiting technique, each sensor monitors the packets which are transmitted to the next-hop sensor node to calculate the sending rate. According to the number of distinct sources routing through the next-hop sensor node, sensors limit their rate and provide fairness. The other mitigation technique of Fusion is a prioritized MAC. Normally in CSMA, all sensors have equal chance of reserving the medium. In a prioritized MAC, congested sensors have prioritized access to the wireless medium. The congested sensors have a smaller backoff window than the non-congested sensors to increase the probability of winning a contention round.

2.1.4. CCF

Congestion control and fairness (CCF) [27] protocol controls the congestion and also provides fairness in the network. The basic concept behind controlling the congestion relies on the routing tree structure. The parent (receiver) node calculates its available bandwidth and assigns it equally among the downstream nodes. This method provides a fair bandwidth distribution and also prevents congestion as the senders cannot exceed the assigned bandwidth.

CCF uses the packet service time to calculate the available bandwidth which is also the available service rate. Packet service time is the period between the time the transport layer sends a packet and the time it receives a notification that the packet has been transmitted. After calculating the available bandwidth, it is divided equally to the child nodes. CCF keeps the total number of child nodes of each sensor node in order to divide the bandwidth fairly. The tree structure is used for calculating the number of children. Each sender adds its sub-tree size as a piggyback into the data packet header. So, the receiver nodes can store the sub-tree size of their child nodes by reading the incoming packet header. The sender node calculates its own sub-tree size by summing all children nodes' sub-tree sizes. Consequently, each node knows how many children it has.

2.1.5. Siphon

Siphon [22] is a congestion control protocol which has congestion detection and mitigation features. Siphon proposes a virtual sink concept. Virtual sinks are specialized nodes and they are different from the physical sink. They have both a primary low power radio and a secondary long-range radio (e.g., IEEE 802.11). Siphon provides three mechanisms: virtual sink discovery and selection, congestion detection, and traffic redirection.

For congestion detection, it uses two techniques which are (i) node-initiated and (ii) physical sink initiated congestion detection. Node initiated congestion detection technique is adapted from CODA [19]. In case of congestion, a sensor node enables the redirection bit in its header and forwards the packet to its nearest virtual sink. The virtual sink which receives a redirection-bit-enabled packet activates its redirect algorithm. On the other hand, in physical-sink-initiated congestion detection technique, the actual sink measures the application fidelity by monitoring event data quality. If the measured application fidelity is lower than a predefined threshold, the physical sink broadcasts a control message to the virtual sinks which receive control messages via their secondary long-range radio. As a result, in the face of congestion, Siphon uses a hop-by-hop data delivery model between the source nodes and the virtual sink and an end-to-end approach between the virtual sink and the physical sink.

For congestion mitigation, instead of changing the event rate, Siphon redirects the packet to the virtual sinks. Traffic redirection method is enabled by the use of one redirection bit in the packet header. Basically this method works as follows: Sensor nodes check the redirection bit of incoming packets. If it is set, packet is forwarded to its virtual sink neighbor. Otherwise, the packet is sent to the next hop which is selected by the underlying routing protocol. When a virtual sink receives a redirected packet, it routes the packets using its own long range communication network toward the physical sink, bypassing the underlying sensor network routing protocols.

2.1.6. FTE

Fuzzy Based Trust Estimation (FTE) [26] is a congestion control scheme based on fuzzy logic systems for WSNs. Basic idea behind the FTE is finding the corrupt or malicious sensors, that are re-forwarding the received packet several times or dropping the incoming packets, and then isolating them to remove the traffic overhead arising from their packets. In [26], the authors define malicious sensor nodes behavior and their negative effects on sensor network. In order to find malicious nodes and mitigate congestion caused by them, FTE uses a fuzzy logic based trust estimation model. Each sensor monitors its neighbor and finds their trustworthiness. In the fuzzy trust estimation model, three inputs are used as a fuzzy variable:

- Forwarded packet ratio: Number of packets forwarded by the suspected node / Number of packets forwarded by the other sensors (except suspected node).
- Delay: Delay of the packets forwarded by the suspected node / Expected delay of packets forwarded by the suspected nodes.
- Validity: Number of valid packets forwarded by the suspected node / Number of packets forwarded by the suspected node.

Fuzzy logic based trust estimation model evaluates the packets coming from neighbor sensors with respect to the above fuzzy variables. According to the fuzzification result, malicious nodes are detected and useless packets are dropped to increase the packet delivery rate.

2.1.7. FCE

Saad A. Munir *et al.* [4] propose a QoS architecture which can classify the packets based on three application classes; (i) source-event driven, (ii) source-continuous, and (iii) sink query based data. The queuing model of their QoS architecture takes advantage of a Fuzzy Based Congestion Estimation (FCE) [25] approach for efficient buffer management. To figure out congestion, FCE approach uses two inputs as a fuzzy logic variable:

- Net packet arrival rate: Ratio of incoming packets to outgoing packets at a specific node.
- Buffer occupancy: Percentage of the buffer fullness at a specific node.

Congestion level is calculated periodically via a fuzzy-logic-based calculation. There are three different congestion levels described in [25]. These levels are called as; *slight congestion*, *somewhat congested*, and *very heavy congestion*. According to the congestion level, rate is adapted and fuzzy logic based queuing model is applied to drop packet from buffers.

FCE is the closest work to ours where it uses a fuzzy-logic based congestion control. However, they differ in the parameters used for the fuzzy logic system (FLS) and the congestion mitigation technique. FCE uses buffer occupancy and packet arrival rate values as an input for the FLS. However, SUIT uses buffer occupancy of the parent (next-hop) sensor node, the ratio of incoming to outgoing packets and the number of contenders. As the congestion mitigation technique, both protocols adapt the data sending rate, but SUIT also performs a novel mechanism which decreases the image quality by dropping the relevant packets during images are being transmitted. Additionally, FCE classifies packets into three categories depending on the type of service layer. According to the congestion level, it applies a packet drop policy to provide the best QoS. However, SUIT has only one service which contains only real time video packets and therefore does not classify packets.

2.2. Reliable Transport Protocols

The reliable transport protocols ensure error-free and complete transmission of data among the sensors. There are several transport protocols in the literature with reliable transmission. These protocols differ in the data type which sensors carry, loss detection, and loss recovery mechanisms. In the following subsections, reliable transport protocols are summarized.

2.2.1. DTSN

Distributed Transport for Sensor Networks (DTSN) protocol [32] provides reliable end-to-end communication. It is assumed that there is a reliable MAC layer which provides two way communication between sensor nodes. Like Transmission Control Protocol (TCP), DTSN also starts an end-to-end session before communication. However, this session has a soft state. The soft-state session (connection) is established when the sender sends its first packet, and the connection is closed when the activity timer finishes. In this protocol, sender waits an explicit acknowledgment request (EAR) message after sending DTSN window amount of packets. The EAR message can be ACK or NACK. One other important issue in this protocol is that the intermediate nodes also cache the packets. If they receive any NACK message, firstly they check their local buffers. If they have the missing packets in the buffer, they immediately send them back to the receiver. Otherwise, they resend the NACK message to the parent sensor node.

2.2.2. RSTP

Reliable Synchronous Transport Protocol (RSTP) [33] provides a reliable and synchronous image transmission. It receives images depending on a schedule in order to provide fair bandwidth usage. If there is more than one node with a video sensor in the network, RSTP gets the same amount of data from each sensor. It does not allow any of them to use the network excessively.

For the connection establishment, RSTP uses a three way handshaking similar to TCP [34]. However, the receiver node (sink) is responsible for the first request, not the sender sensor (source). This is a common way for sensor devices, because generally designers want to perform as less effort as possible on sensor nodes. Connection termination is also very similar to the TCP. The sender triggers the termination and the receiver puts a Finish (FIN) flag into the last ACK message in order to inform the sender about the termination. End-to-end loss recovery is performed in RSTP. When the receiver detects a packet loss, it sends a NACK message back. Packet loss

is detected by comparing the last packet number with the expected packet number.

2.2.3. PSFQ

Pump Slowly Fetch Quickly (PSFQ) [35] is a reliable transport protocol for WSNs which provides downstream reliability guarantee. This protocol is suitable for the applications in which the sink disseminates jobs to a group of target sensor nodes periodically. Main idea of PSFQ is sending data from the sink at a slow speed, but fetching the data loss and recovering it immediately on intermediate nodes. It comprises three operations: *pump*, *fetch*, *report* operation. In the pump operation, the sink periodically broadcasts packets to its neighbors. The pump operation is finished when all data fragments have been sent out. The receiver sensor first checks if this packet is duplicated or not. For unduplicated packets, PSFQ sets a schedule to forward them if there is no gap in the sequence number.

The fetch operation is initiated if a packet loss occurs. The lost packet is detected by checking a sequence number gap in a file fragment. If the packet loss is detected, the fetch operation sends a NACK in the reverse path to request a retransmission from neighboring nodes. A neighbor node which receives a NACK message checks its cache for the missing packet. If the missing packet is found in the cache, neighbor node retransmits it. If the lost packet is not available in the cache and the number of NACKs which the intermediate node has received exceed a predefined threshold, the received NACK at an intermediate node is relayed. Otherwise, it is not relayed to avoid message implosion.

In the report operation, the sink collects feedback about the data delivery status. The report message is initially sent by the sink. It is piggybacked to the packet instead of sending many control messages in order to decrease the communication cost. The message travels among nodes in a hop-by-hop manner, and each node can append its own feedback information to the original report message. Finally, it propagates back toward the sink that initially requested the report.

2.2.4. RMST

Reliable Multi-Segment Transport (RMST) [36] is a transport protocol which runs above directed diffusion [37] to guarantee delivery of all packets from each source to the sink. The directed diffusion paradigm which is explained in Section 3.3 is different from traditional networking. RMST supports loss detection and repair. It uses two different methods for loss recovery. One of them is *end-to-end selective request NACK* which is called non-cached mode. The other one is *hop-by-hop selective request NACK* which is called cached mode. RMST uses a timer-driver mechanism to detect data loss. In the non-cached mode only the sink sets up a timer. However in cached mode, each caching node on the reinforced path from source to the sink also sets a timer to detect loss. So, in this mode NACK receivers are responsible for finding the missing packet or forwarding NACK on the path toward the sink, if it fails to find the missing packet. However, in the non-cached mode intermediate NACK receivers just relay the packet through the source.

2.3. Reliable Transport Protocol Providing Congestion Control

In this section, a brief overview of transport protocols that provide both reliability and congestion control is provided.

2.3.1. RCRT

Rate-Controlled Reliable Transport (RCRT) [28] protocol provides central congestion control and reliable end-to-end transmission. For reliable delivery, it uses an end-to-end mechanism which provides NACK based reliability. If the sink detects the missing packets, it explicitly requests them from the sensors by sending NACK messages back. If there is no packet loss, the sink sends a cumulative ACK message to the sensor node. The sink also deals with sorting unordered packets.

RCRT uses a central congestion control instead of a distributed one. The central control takes places at the sink, hence, calculations are performed on the sink side.

For the congestion decision, RCRT protocol uses the round trip time (RTT) and the recovery time of a lost packet. RTT time is calculated while establishing the connection which is performed via three way handshaking like TCP. If the recovery time of the lost packet or RTT is too long, it is assumed that there is a congestion. Depending on the congestion, the sink node estimates the total rate of all flows, rather than the rate of a single flow. The rate estimation is performed in an additive increase multiple decrease (AIMD) manner. If RCRT detects that the network is congested, the rate is decreased multiplicatively. Otherwise, the rate is increased additively. After the total rate is calculated, RCRT assigns rates to each flow according to the rate allocation policy.

2.3.2. ESRT

Event to Sink Reliable Transport (ESRT) [23] protocol provides reliable event detection and congestion control for WSN. ESRT is designed for the event based applications in which sensor nodes report detected events to the sink periodically. Therefore, instead of conventional end-to-end reliability, it just addresses event-to-sink reliability.

ESRT provides event reliability by configuring event reporting frequency of the source nodes with minimum energy consumption and without network congestion. The sink checks the event-to-sink reliability level. If it is lower than the required level, the reporting frequency of source nodes are increased aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, ESRT reduces the reporting frequency in order to save energy.

For congestion control, sensor nodes monitor their local buffers in order to detect congestion. If their local buffer overflows because of the incoming packets, the network can be decided as congested. In case of congestion, a sensor node sets a bit which is called *Congestion Notification Bit* in the header of the packets it transmits. The sink uses this notification while deciding on the new reporting interval.

2.3.3. STCP

Sensor Transmission Control Protocol (STCP) [24] is a generic end-to-end upstream transport protocol which provides both congestion detection and avoidance and a variable degree of reliability based on the application requirement. STCP uses three types of packets: *session initiation packet*, *data packet*, and *ACK packet*. Source sensors declare their variable reliability to the sink while establishing a connection via the session initiation packet. This packet includes the number of total packets originating from the node, the type of data, transmission rate and required reliability. Based on this information, the sink starts to monitor the session. If the reliability goes below the required level, the sink sends a NACK message to the source. Otherwise, it does not send a NACK even if it does not receive any packets within the expected time interval.

Congestion control is performed by using data packets. Each data packet has a *congestion notification bit* in its header. In order to decide on the congestion, the buffer size of the sensor is used. If the buffer size of the intermediate node reaches a threshold t_{high}, the node notifies the sink by setting a congestion notification bit in the data packet headers. The sink which receives a packet with the congestion notification bit set sends an explicit notification to the source. On receiving this notification, the source either decreases the transmission rate or route packets along a different path.

2.4. Summary

Considerable amount of research has been done to implement efficient communication protocols for WSNs. These protocols mainly address reliable transmission and congestion control problems. In Section 2.1, popular congestion control protocols are summarized. Most of the protocols use buffer occupancy, channel condition and packet service time to detect congestion. Only *FCE* uses fuzzy-logic based approach for congestion detection. Majority of the protocols use hop-by-hop congestion notification. Only *Siphon* performs end-to-end congestion notification. For congestion mitigation, a considerable amount of the protocols apply rate adaptation and some of them drop

Table 2.1. Classification of the protocols based on congestion control.

Protocols	**Congestion Detection**	**Notification**	**Congestion Mitigation**
CODA [19]	Buffer Size and Channel Condition	hop-by-hop backpressure and closed-loop multi-source regulation	rate adaptation and packet drop
SenTCP [20]	buffer occupancy and packet arrival rate	open-loop hop-by-hop	rate adaptation
Fusion [21]	queue size and channel sampling	hop-by-hop flow control	rate adaptation and prioritized MAC
CCF [27]	packets service time	hop-by-hop	routing tree based fairness
Siphon [22]	Buffer Size and Channel Condition	hop-by-hop and end-to-end	traffic redirection
FTE [26]	forwarded packet and packet delay	-	drop packets of malicious sensors
FCE [25]	Buffer occupancy and packet arrival rate	-	rate adaptation and packet drop

packets or perform traffic redirection. In Table 2.1, congestion control protocols are summarized.

Protocols which propose reliable transmission are summarized in Section 2.2 and compared in Table 2.2. These protocols support either upstream or downstream traffic. All of the protocols provide packet reliability by using NACK based loss reporting. However, multimedia traffic has different characteristic compared to the conventional data or event-based traffic. Reliability for all packets in the network is not appropriate for multimedia traffic due to the reasons defined in [3, 38]. For loss recovery, generally hop-by-hop loss recovery is applied, but end-to-end approach is also used.

Protocols which provide both congestion control and reliability are summarized

Table 2.2. Classification of the protocols based on reliability.

Protocols	**Trafic Direction**	**Reliability Type**	**Loss report**	**Loss Recovery**
DTSN [32]	two way	packet	NACK	hop-by-hop
RSTP [33]	two way	packet	NACK	end-to-end
PSFQ [35]	downstream	packet	NACK	hop-by-hop
RMST [36]	upstream	packet	NACK	hop-by-hop and end-to-end

Table 2.3. Classification of the protocols based on congestion control and reliability.

Protocols	**CD**	**CN**	**CM**	**Direction**	**Reliability**	**LN**	**LR**
RCRT [28]	RTT	-	yes	upstream	packet	NACK	E2E
ESRT [23]	buffer size	HbH	yes	upstream	event	-	E2E
STCP [24]	buffer size	E2E	yes	upstream	packet and event	NACK	E2E

CD: Congestion Detection, CN: Congestion Notification, CM: Congestion Mitigation, LN: Loss Notification, LR: Loss Recovery, HbH: Hob by Hop, E2E: End to End

in Section 2.3. These protocols are listed in Table 2.3. These protocols provide either packet or event reliability. All of them has upstream traffic and end-to-end loss recovery. They use buffer occupancy or round trip time for congestion detection and use hop-by-hop or end-to-end mechanism for congestion notification.

A common feature of most of these protocols is that they address the communication problems of the WSN applications which require conventional data communications. Firstly, most protocols focus on reliability that targets 100% reliability and this cannot be applied to multimedia data. Secondly, they focus on providing a generic transport layer which is independent from the underlying layers. However, cross layer solutions facilitate to optimize the protocol stack. For example, the transport layer can work with the routing or MAC layer to get more information about the state of the wireless channel. Finally, none of the mentioned protocols address semantics of video encoding standards which can help to optimize transmission performance. Therefore, designing new image transport protocol addressing the discussed criteria for WSNs is highly demanded.

3. ISSUES FOR MULTIMEDIA STREAMING IN WMSNs

WMSN is a subtype of WSN that consists of small video sensors which provide retrieving video and audio streams besides scalar data. WMSNs have become popular because they enable lots of new applications such as multimedia surveillance, health care, environmental monitoring, smart home, automated parking advice, and traffic avoidance and control applications [3, 39]. Multimedia data transmission over WSNs has some challenges compared to the conventional data transmission. In this section, we discuss these challenges and issues for multimedia transmission in WMSNs.

One of the major challenges is bursty high-load traffic arising from multimedia streaming. Because of the limited data rate and buffer capacity of sensor devices, bursty traffic causes congestion and packet loss which degrades the QoS and network performance. Generally, conventional WSN applications do not consider high-load data because they are designed for monitoring low-load environment-related data, such as temperature information, for longer periods. Thus, such applications ignore the high-load data scenarios and focus on other problems. However, multimedia streaming applications have high bandwidth requirements and strict delay constraints. As a result, multimedia data traffic should be processed in a different manner with respect to conventional data traffic.

Another key challenge is the real-time requirement of the video applications. Most of the WMSN applications e.g. multimedia surveillance has strict delay requirements. The video frames should be received in a reasonable amount of time. Therefore, delivering multimedia data under strict delay constraints over WMSN requires different protocol design than the conventional protocols which provide 100% guaranteed delivery of all packets to the destination rather than dealing with strict delay constraints. Generally applications which deal with real-time data cannot deliver all off the data due to limited channel rate. Instead of 100% guaranteed delivery, they provide less reliability for multimedia transmission. As a result, data transmission and packet dropping techniques have become very important in WMSNs.

Other challenges to be considered are low power and computational capabilities of the sensor nodes. As mentioned, sensor devices have limited battery that also limits the network lifetime. Since the network lifetime is one of the fundamental constraints on any WSN application, energy efficiency is the most crucial performance metric. Protocols designed for WSNs must consider the energy limitation of sensor devices at every phase of the design. In WMSNs, there is another important criterion that affects the energy usage. This criterion is the video encoding which consumes considerable amount of energy because coding algorithms which are too complex for sensor devices have to be performed. Conventional video coding techniques are designed with respect to a powerful station (such as digital video broadcaster), and less powerful clients (such as televisions). In our case, the scenario is exactly the opposite. Less powerful sensor nodes encode the video and the powerful sink node performs the decoding. Therefore, an appropriate video coding technique should be selected for WMSN applications.

3.1. Application Layer Issues (Video Encoding)

The application layer should process the collected data to properly send or receive it via the underlying protocol in the network. In WMSNs, most of the data traffic is occupied by video data. Video coding and compression techniques become very critical because processing video data is challenging for sensor nodes which have limited energy, bandwidth and processing capability.

The encoder used in wireless sensors should have high compression capability, low complexity and high error resiliency [3]. However there is a trade off between compression an energy usage while transmitting video over WMSNs: Video coding shall have high compression, but this is a complex operation. High compression reduces the number of transmitted bits, so it provides better bandwidth usage. On the other hand, using a complex compression technique increases the local calculation which consumes more power and causes inefficient energy usage.

There are two different video encoding paradigms used in WMSNs. One of them is predictive video coding and the other one is distributed video coding. Predictive video

coding has been used for years and adopted in most of the available video standards since the end of the 1980s. Especially ITU-T H.26x and ISO/IEC MPEG-x families use this standard [40]. In this type of coding, video is encoded once and clients can decode it anytime. Generally encoders are powerful devices and the decoders are quite less powerful devices with respect to the encoders. Bernd Girod *et al.* [41] show that, in MPEG [10], H.264 [42] or H.263 [9] video coding techniques, encoders are five to ten times more complex than the decoders, because the algorithms used for prediction in the encoder side is very complex. For this reason, video is encoded on the encoder side by powerful machines, and decoded on the clients by using low CPU usage and calculation.

Traditional video compression standards such as MPEG or H.263 require much more computation power for the encoder side. The encoding versus decoding asymmetry mentioned above is suitable for video broadcasting or video-on-demand (VOD) streaming systems where video is compressed once and decoded many times. However for wireless sensor applications, encoding is performed by sensor nodes and decoding is handled by the sink. Hence WMSN application scenarios are different than the video broadcasting or VOD streaming. For this reason, distributed video coding proposes low-complexity encoders and high-complexity decoders. Distributed video coding is one of the video compression paradigm and the theoretical studies were firstly performed by Slepian and Wolf, and Wyner and Ziv in 1970s. Firstly, Slepian and Wolf developed a lossless distributed coding theorem [11]. Shortly after Slepian and Wolf's seminal paper, Wyner and Ziv have extended their work to establish a lossy compression with side information at the decoder [12]. Today most of the distributed video coding algorithms are based on Wyner-Ziv coding. The Wyner-Ziv theory suggests unconventional video coding which encodes individual frames independently and decodes them conditionally. This technique cannot provide as high compression as intercoding technique such as MPEG, but it provides better compression than the intra coding techniques such as motion JPEG (MJPEG).

In summary, predictive video coding provides fairly high data compression which enables better bandwidth usage. On the other hand it consumes more energy due to the

complexity of motion estimation and prediction algorithms. Due to high computation power and energy requirements, predictive encoding is not appropriate to be used by WMSN sensors [43]. Distributed video encoding has less complexity than predictive video encoding, because neither motion estimation nor prediction is required at the encoder side. Because of having less complexity, distributed video coding has attracted a lot of interest. However, there is still a difficulty to find an explicit distributed video coding model for real world images [44]. Lack of explicit models prevents using distributed source coding in practical applications. Therefore WMSN applications generally require alternative coding techniques which provide both compression and a simple encoding algorithm.

JPEG is one of the potential candidates for WMSN applications. It is the current standard for compression and decompression of still, continuous tone, monochrome and color images [45] and it is widely used in WSNs applications. In Section 3.6, the method that we use for transmitting JPEG images over WMSN is explained in detail.

3.2. Transport Layer Issues

Basically, a transport protocol is responsible for delivering the data from the source to the destination. Because of the multihop topology of WSNs, data flow across several intermediate sensor nodes before arriving to the destination. Therefore, transmission takes place between two or more devices in the network. There are one sender, one receiver and possibly some number of intermediate sensors. Traffic direction can be downstream (data flows from the sink to sensor nodes) or upstream (data flows from sensor nodes to the sink) [38]. In WMSN applications, sensor nodes collect video data from environment and send it towards the sink, so the main data flow is in the upstream direction. There can also be downstream traffic due to queries and control commands from the sink to the sensor nodes. Since the dominating traffic is in the upstream direction, transport protocols should perform most of the tasks at the sink side.

Transport protocols used in wireless multihop networks are different than those

in the wired networks [18]. In wired networks, the channels are mostly reliable and packets are not dropped due to the interference. Moreover, topology of the wireless multihop networks can change dynamically because of mobility of devices or dead devices whose battery is exhausted. This may cause unexpected packet losses and delays. Another important point is the complexity of the conventional protocols. Generally these protocols are designed without considering the energy efficiency. They send a lot of control messages in order to manage the network condition and to provide better quality of service. In WSNs, the lifetime is very critical, so sensor activity should be decreased as much as possible. In brief, transport protocols for WSNs should be designed by considering the above mentioned factors.

In addition to the end-to-end data transfer, an efficient transport protocol should handle reliability and congestion control. A reliable transport protocol is able to assure the sender if the packet is received by the receiver or not. Generally if a transport protocol is reliable, it provides 100% guaranteed delivery of all packets to the destination. Including TCP, most of the protocol use ACK/NACK based system to detect packet losses. Some protocols use forward error correction (FEC) or similar error correction mechanism to recover lost packets without the need of retransmission. In WMSNs, where sensor nodes transmit real time video data, 100% guaranteed delivery is not necessary, because detecting a packet loss via ACK/NACK messages increases the newer packets delay, so that video frames may be received by the sink at an unacceptable latency. Also FEC or a similar error correction mechanism brings a packet size and calculation overhead. Since sensor nodes have limited computation power and memory, protocols should be implemented as simple as possible.

In reliable networks, there is a mechanism for recovering lost packets. As it is mentioned above, packet loss probability of WSNs can be quite high with respect to the other reliable wired networks. During the transmission, data can be lost due to the wireless medium unreliability or sensor node failure. Generally missing packets cause retransmission. There are two common methods for retransmitting lost packets. One of them is end-to-end retransmission in which just the sender and the receiver sensors are dealing with the loss recovery. The other one is hop-by-hop retransmission.

In this method, not only the sender and receiver but also all intermediate sensors are responsible for loss recovery. End-to-end manner is easy to implement but if the hop count is high, this method depletes more energy because a retransmission from source to destination is required for each missing packet. On the other hand hop-by-hop manner provides energy efficiency because it can retransmit lost packet via the intermediate sensors instead of the sender sensor. However, this method is difficult to implement and sensor nodes need more memory for buffering the packets which are passing through them.

Congestion is one of the most important issues that should be handled while transporting video data on WMSNs. Congestion occurs when a sensor node carries so much data that deteriorates the QoS. In wired networks, congestion control is usually done in an end-to-end manner, and the congestion decision is made with respect to the packet drops and increased delays. However, in WMSNs the end-to-end approach is not suitable. Because of the multi hop topology of sensor networks, packet drops and increased delays are not the only indicators for the congestion. For example, TCP determines the congestion based on packet loss. When it detects a lost packet, its congestion control mechanism is invoked to enter the slow start and the congestion avoidance mode. In this mode, the transmission window is decreased and the retransmission timeout (RTO) is increased. Although TCP works well at congestion control for wired networks, it is not suitable for wireless networks. Because the wireless medium has higher packet loss ratio than wired networks, so a packet loss does not always infer congestion. In brief, controlling the congestion in WSNs is required for the protocols to ensure the QoS of the network. Since it is a difficult task, there are lots of protocols introduced which only deals with the congestion control [19–21, 25–27].

In Chapter 5, a fuzzy logic based congestion control mechanism is introduced. The concept of Fuzzy Logic (FL) was first proposed by Lotfi Zadeh, in early 1960s. In 1965 he has published a seminal paper [46] on fuzzy sets and this is admitted as the birth of fuzzy logic technology. The fuzzy set theory is different than the traditional set theory. A traditional set contains crisp variables so that the membership of a variable is either 1 or 0 (true or false respectively). However, a fuzzy set contains

uncertain variables which do not have precisely defined criteria of membership. As a result, fuzzy set variables can have membership grade between 0 and 1 instead of a Boolean result [46]. In order to use FL in applications, a great deal of systems have been proposed [47]. A FLS is generally used for mapping uncertain variables from complex real world to a scalar data. Jerry M. Mendel defines the FLS as a nonlinear mapping of an input data (feature) vector into a scalar output in his tutorial [48].

3.3. Routing Layer Issues

The routing layer is responsible for providing energy efficient and stable routes. Therefore, the routing layer is an important factor on QoS for multimedia streaming application in WSNs [18]. Routing is not a subject of this thesis but fundamental issues are explained in this section.

Routing over WSNs is a difficult task due to lossy links, dynamic topology, minimal bandwidth and power constraints [49]. Route selection should minimize the network congestion and the end-to-end packet delay. In WSNs, multihop transmission is used in order to decrease energy usage, because short-range wireless communication depletes less energy. On the other hand, multihop routing requires more processing at intermediate nodes and increases the packet delay due to queuing. Therefore, larger number of hops consumes less energy but it also results with higher end-to-end delay.

Raghunandan *et al.* surveyed the current routing protocols and they classified routing in WSNs into flat-based routing, hierarchical-based routing and location-based routing depending on the network structure. In flat-based routing, all nodes are typically assigned equal roles or functionality. In this type routing, there is no synchronization between nodes which requires contention-based scheduling. Therefore, links are formed on the fly without synchronization. Flat-based routing is a data centric routing where base station sends queries to certain regions and waits for data from the sensors located in the selected regions. On the other hand, the hierarchical routing has two kinds of nodes which differ in the sensing the proximity and sending data. Low-energy nodes can be used to sense proximity of the target and to select cluster heads. Higher

energy nodes can be used for sending the information (routing). This type of routing requires synchronization between nodes and uses reservation-based scheduling. The third category is the location-based protocols. In these protocols, sensor nodes have the information of their location. They can calculate the distance between neighboring nodes via incoming signal strengths or directly by communicating with a satellite, using global positioning system (GPS) with a small low power GPS receiver. This kind of protocols create the links between nodes based on their locations. GPSR [50] is one of the geographic routing protocol which is used in our simulations.

Flooding based routing protocols are used in WSNs to provide guaranteed delivery by routing packets through multiple paths. Flooding based protocols are very easy to implement. However, these kinds of protocols are not suitable for multimedia streaming applications due to redundant data transmission and consuming large amount of energy [51]. To overcome drawbacks of conventional flooding protocols, some studies are performed, e.g. SPIN. For a given amount of energy, SPIN protocols can deliver 60% more data than conventional flooding protocols [52].

Directed Diffusion [37] is another routing scheme which is commonly used in WSNs. In directed diffusion based protocols, each sink needs to register an event. To register an event, sinks inform all neighbors about their interest and make it possible to disseminate them throughout the sensor network. The interest is defined using a list of attribute-value pairs. Sensors which receive a message containing the interest information, create a gradient. Direction is set toward the neighboring node from which the interest is received. When the events occur they start flowing towards the originators of interests along multiple gradient paths. The sensor network reinforces one or a small number of these paths. Since directed diffusion based protocols do not need to maintain the network topology, they provide high energy efficiency. However, these protocols are designed for query-driven applications and therefore they cannot work efficiently for applications that transmit data to the sink continuously [51].

For efficient video transmission, multimedia data should be routed to a dynamically selected node. The next hop node is decided with respect to the energy efficiency

and application QoS requirements. This technique is proposed by some routing protocols which address multimedia streaming over WMSNs [53, 54]. Ciubotaru *et al.* show that their routing protocol, which is Adaptive Routing (AR) based and designed for video streaming, achieves better performance than flooding and directed diffusion based routing protocols [53].

3.4. MAC Layer Issues

Medium access control (MAC) protocols for WSN make it possible for several sensor nodes to communicate with each other over a shared medium. Main goal of these protocols is minimizing the collision in order to prevent two interfering nodes from transmitting data at the same time. MAC protocols for WSN should adapt topology changes easily because sensor nodes may move or may die over time due to lack of battery.

There have been several studies about MAC protocol design for WSNs [55]. Most of the protocols are either contention based or reservation and scheduling based. Reservation and scheduling based (e.g. TDMA) protocols consume less energy because the duty cycle of the radio is reduced and there are no contention and collision overhead. However these kinds of MAC protocols require synchronizing of all sensor nodes. Because of the dynamic topology nature of WSNs, it is very difficult to synchronize all sensor nodes. On the other hand, contention based MAC protocols have an advantage regarding scalability. Since the sensor nodes contend with each other while accessing shared medium, they do not have to be synchronized with other sensors. Contention based protocols are not as efficient as TDMA based protocols in terms of energy consumption. In contention based protocols, a sensor node listens to the medium even if it is in the idle mode in order to receive incoming requests. Compared to TDMA based protocols, energy consumption of contention based protocols is higher due to idle listening.

For the WSN applications such as multimedia surveillance and environmental monitoring, sensor devices send data when they sense an event such as unauthorized

entrance into an area or earthquake. For these applications, sensor nodes stay in idle mode for a long time if no sensing event happens. In order to use the battery efficiently, SMAC [56] tries to reduce the energy consumption of the sensor nodes via a periodic sleep mode. Instead of listening to the medium all the time, it has a sleep and listen period. In the sleep period, it turns off the radio of the sensor node so that it reduces the energy consumption. In the listen mode, it turns on the radio again. There are lots of MAC protocols for WSNs which have the same technique as SMAC to decrease the energy consumption of the network by integrating a kind of sleep-listen schedule, such as MD-SMAC [57], N-MAC [58], T-MAC [59], and Diff-MAC [1].

3.5. Cross Layer Approach

In layered network design, operations are divided into hierarchical layers. Each layer is responsible for performing interrelated functions. These functions together provide a service that allows two systems to exchange data. Basic idea behind this approach is dividing the network communication process into smaller and simpler components. Layered network design has a lot of advantages [60]: It reduces the complexity of the design, development and testing phases because each layer focuses on its own task and it does not have to consider other layers. If one of the layers is changed, other layers are not affected by this. Another advantage is that, layered network allows different types of network hardware and software to communicate with each other. For example, an application which is running on top of a layered model can run any device which supports the related network model. The application does not need to know how data is sent via the underlying layers.

In the conventional layered approach, packets flow from upper layers to lower layers. Each layer can only communicate with the next layer. This behavior brings advantage because of the above mentioned reasons. To explain their advantage, personal computers can be discussed as an example. They provide wired and wireless connections by using Ethernet or Wi-Fi adapters. The standardized communication protocol for personal computers is the TCP/IP suite. With the help of the layered model, the same application can be used regardless of the connection type. It is not needed to

create entirely different network software for each connection type. Instead, we can just edit the network layer (Network Access Layer) to allow compatibility. However, unlike the conventional network application, WMSN application runs on a specific device. Separating layers strictly is not required in case of the different network hardware and software possibility. Moreover, layers may need to share some information with each other to provide better QoS. For example, routing layer may need to know the type of the application while deciding on the next hop to forward data. Similarly an application may require the channel condition which MAC layer can provide to adapt the video rate. As a result, the layered network design is not adequate for WMSNs.

The main goal of the cross-layer approach is increasing the performance of wireless networks by using the interactions between various layers of the protocol stack. However, exchanging information over two or more layers does not mean that there is no layered architecture. Breaking all of the layers can destroy modularity and leads to spaghetti design which results in various negative consequences [61]. As stated before, one of the main advantages of the layered architecture is the low complexity. It enables designers to split the overall stack into logically related operations. While designers are working on a particular layer, they do not have to worry about the rest of the stack. This advantage is no longer available in spaghetti designs, which can make it difficult to develop, debug and improve for further innovation. Therefore, the interaction of the layers is very important and a well designed cross layer architecture should consider these aforementioned drawbacks. There are several studies which propose cross layer optimization techniques and explain why a cross layer solution is needed for sensor networks [61–64].

3.6. Image Transmission Over WMSNs

A real-time streaming application is more challenging than data sensing applications for WSNs. Multimedia data transmission requires more bandwidth and shorter delay than the other data. In order to use the bandwidth efficiently and transmit the video frames as fast as possible, video coding/compression techniques play an important role on the efficiency of the application. As stated in Section 3.1, in WMSN

applications video compression should produce a video stream of the small size and should tolerate packet loss. On the other hand, it should have low complexity and consume as low power as possible due to the battery limitations of the sensor nodes.

The most popular video coding techniques are MPEGx or H.26x series which are defined by the IS0 (Interactional Standardization Organization) and the ITU (International Telecommunication Union) standards. Encoding a video with MPEGx or H.26x based encoding provides good compression but video sensors struggle to handle encoding complexity which is arising from motion estimation and compensation based predictive coding technique [65]. Moreover, predictive coding techniques deplete pretty high amount of energy. For these reasons, video sensors typically use simpler coding schemes such as JPEG and JPEG2000, instead of conventional MPEGx or H.26x video formats which have complicated compression algorithms. Since sensor nodes are always in the shortage of battery, memory and processing capacity, there are several studies that use JPEG images in their WMSN application [13–16].

The JPEG is the dominant image compression technique for multimedia sensor nodes. Wallace *et al.* explain JPEG standard in detail in [66]. JPEG has two different compression methods as *lossless* and *lossy*. The lossy compression technique is based on the Discrete Cosine Transform (DCT) and known as the *baseline* method. This method is commonly used by the applications which deal with JPEG images. The compression used in this thesis will be treated as the baseline unless specifically declared as lossless or lossy. A baseline JPEG image can be encoded with different parameters. Encoding parameters lead to a JPEG structure variety. In the following sections, three different types of JPEG images which are encoded in a different manner are explained. These types are: *Sequential DCT based JPEG*, *Progressive DCT based JPEG*, *Sequential JPEG with restart marker*.

3.6.1. Sequential DCT based JPEG

Sequential encoding is the most popular encoding mode for DCT based JPEG. Most of the images used in the Internet are Sequential JPEG (SJPEG). In SJPEG, each

image component is encoded from left to right and top to bottom route. Therefore, while downloading a large image with slow connection, the image is displayed line by line. In order to see the whole image, the file should be downloaded completely. DCT based JPEG use variable length codes (VLC) instead of fixed-length codes (FLC). The advantage of VLC is reducing the bit rate, but they have one major disadvantage. Because of using the variable length coding, they are very sensitive to channel errors. A single bit error can lead to a complete loss of synchronism at the decoder which results in a completely unusable picture. There have been numerous studies in the literature [67–69] in order to overcome this disadvantage. As a simple solution, restart markers which are defined in JPEG standard can be used. While transmitting data over an unreliable channel it provides resynchronization of the decoders.

3.6.2. Progressive DCT based JPEG

If the network bandwidth is low and the size of the image is high, it takes several minutes to transmit the image. As stated in 3.6.1, the receiver (generally the sink) receives top-to-bottom display of image gradually and it has to wait for a long time in order to get a full-quality picture. To overcome this drawback, progressive encoding is used. In progressive DCT-based JPEG (PJPEG), image is encoded into a series of scans. Therefore an image can be sent in layers rather than scan lines. When the first layer is received, a blurred but recognizable image can be seen. Whenever the subsequent layers arrive, the image quality gradually increase and more precise views can be achieved.

The advantage of PJPEG is that an image can be viewed on-the-fly while it is being transmitted. The client can see a low quality version of the whole image very quickly with gradual improvement of quality. For WMSNs, PJPEG is more suitable than SJPEG which provides a slow top-to-bottom display of the image. On the other, hand displaying each scan requires the same amount of computation to display a whole SJPEG file. Therefore it brings an extra overhead to the client device. In WMSNs, generally the sink node acts as an image viewer device. Since the sink has relatively higher computational capability than the sensor nodes, PJPEG does not affect the



(a) Original PJPEG (b) Some layers are corrupted (c) More layers are corrupted

Figure 3.1. Packet loss effects on progressive JPEG.

performance of the overall system. As a result, if a WMSN application needs to start displaying an image on the fly (without waiting for all the data to be received) progressive DCT-based JPEG is the most appropriate one.

Although the progressive feature has been defined as a *not required to be supported* extension in JPEG standard, for some specific applications it is very useful. For instance, if an application sends big JPEG images over a low bandwidth network, the receiver can display the image with increasing quality instead of waiting for all data to be received. Especially for WMSN applications, the end-to-end delay of an image frame would be very high due to the multihop topology and low bandwidth of the network. For this reason, in this thesis using PJPEG is proposed as the most appropriate technique. In Figure 3.1a, the packet loss effect on PJPEG is depicted. Some layers (subframes) of the JPEG image is dropped and the resulting image is given in Figure 3.1b. In Figure 3.1c more layers of the original image is dropped and the resulting image is shown.

Another important advantage of PJPEG is its error resilience capability. Unlike SJPEG, PJPEG consists of several layers which compose a JPEG image frame. In case of a data loss or error in any layer, the whole frame is not corrupted. Only the related layer is corrupted but JPEG frame can still be reconstructed from the remaining layers. By using error resilience advantage of PJPEG, a novel congestion mitigation method is proposed in Chapter 5. If the network is congested, the image quality can be decreased

on the fly by dropping one layer or layers from the PJPEG. Only PJPEG makes it possible to decrease the image quality on intermediate sensors, because of its layered structure. In Section 5.2, the method of decreasing the image quality on the fly is explained in detail.

3.6.3. DCT based JPEG with Restart Markers

As stated in Section 3.6.1, in case of bit error(s) or loss of packet(s) on the DCT encoded lossy compressed JPEG, image may be corrupted due to VLC. To overcome this problem JPEG has a restart marker feature. The restart marker is used in JPEG to divide data segments into independent blocks. It is mostly used on unreliable networks. The advantage of the restart marker is its high packet loss resilience. If there is a bit error or data loss inside the marked block, just the related block become useless rather than the whole frame. The restart marker is used both for sequential and progressive DCT based JPEGs. However, in this thesis, it is only applied for sequential JPEG.

Restart markers add a small amount of overhead to the image size because each data block has additional 2 bytes. The size of data blocks is configurable. Generally JPEG editing tools allow using restart marker after each 16x16 pixel area or after each row of blocks which has 16 pixel heights. Since restart marker brings extra overhead to the file size, using restart marker for each row is decided to be more reasonable for WMSNs rather than marking each 16x16 pixel area. Therefore, in this thesis, all of



(a) Original SJPEG-RM (b) One block is corrupted (c) Two blocks are corrupted

Figure 3.2. Packet loss effects on sequential JPEG with restart marker.

the restart markers are used after each row of blocks.

In Figure 3.2, packet loss effects on sequential JPEG with restart marker (SJPEG-RM) is depicted. Since the marker is applied after each row of blocks, any packet loss or bit error in the block causes to lose the related row completely. However, the other blocks are not affected. In Figure 3.2b the 5^{th} row, in Figure 3.2c 15^{th} and 25^{th} rows are corrupted and the resulting images are depicted.

3.7. Summary

In conclusion, SJPEG is used frequently in WMSN applications [13,15,16]. However, it is highly vulnerable to packet loss and bit errors. If any data is lost or corrupted, the whole frame may be useless. To overcome this drawback, the restart marker property of JPEG can be used. This method is less sensitive to packet loss and bit errors, but it uses extra two bytes to mark each block. Using the restart marker increases the file size of the image but it can be ignored since the difference is very small. In Chapter 6.1 different types of JPEG encoding methods are compared with respect to the file size. SJPEG with and without restart markers have almost the same size as shown in Table 6.1. Therefore, using restart markers does not bring an extra overhead for the WMSN applications regarding the file size. Another encoding type which is mentioned in this chapter is the progressive encoding. Progressive encoding usually gives better compression when compared to sequential encoding due to the ability to use different Huffman tables for different frequencies on each *scan* or *pass*. However, the file size difference is not too large. As shown in Table 6.1, PJPEG has slightly lower file size than the SJPEG but they are very close to each other. Except for the ability to mark a block of data or provide progressive display, all of the encoding types are basically identical, and it is possible convert any of them to any other without quality loss.

4. SENSOR FUZZY-BASED IMAGE TRANSPORT (SUIT)

Transmitting images over WSNs is not a new concept. There are several protocols proposed for image transmission in sensor networks [70–72]. In this chapter, a new cross-layer connectionless image transport protocol is introduced.

SUIT is an image transport protocol which provides a fuzzy logic based congestion control mechanism. The main idea behind SUIT is transmitting the maximum number of frames to the sink by decreasing the frame quality to an acceptable level in case of congestion. With this method, it is possible to transmit frames with lower quality instead of dropping them. Thus, SUIT increases the average frame delivery which is an important QoS metric for video streaming applications. Basic properties of SUIT are listed as follows:

- Connectionless Communication
- Packet Prioritization
- Cross-Layer Data Exchange among Layers
- Congestion Detection and Mitigation

SUIT is developed for applications which have the main goal of transmitting non-continuous video from source sensor nodes to the sink as fast as possible. A good example for this kind of application is a video surveillance application. Video surveillance systems monitor the region of interest (ROI) and send the important data to the base station. If the sensor detects a target in its field of view (FoV), it streams the captured video data to the sink. Generally, the target is a moving object such as a thief, a vehicle, or an animal. Because of target's mobility capability, it enters the sensor's FoV and leaves within a short period.

SUIT streams a sequence of JPEG images instead of conventional MPEGx or H.26x based video coding techniques which have the drawbacks explained in Section 3.6. Basically they have complex compression algorithms which are not appropriate

for the sensor nodes due to their limited capabilities in terms of power, processing, and memory. Therefore SUIT supports the JPEG codec. Although it is capable of streaming both sequential and progressive JPEG, the congestion control module is designed for PJPEG. Performance of SUIT is evaluated for different types of JPEG encoding (SJPEG, SJPEG-RM and PJPEG). It provides the best performance by using progressively encoded images as shown in Section 6.2.

The most important feature that SUIT provides is the congestion control mechanism. It uses a fuzzy logic based congestion control in order to estimate the congestion and then mitigate it. As explained in Section 3.2, the congestion control is a difficult task for WMSN protocols because there are several factors that can lead to congestion. Therefore, fuzzy logic is used for mapping different metrics into a single value. For congestion mitigation, this thesis proposes a novel idea which allows intermediate sensor nodes to decrease the image frame quality on the fly. In Chapter 5, the congestion control mechanism is explained in detail.

SUIT performs a connectionless transmission. Like UDP [73], it does not establish a connection before sending data. The source sensor transmits the image frames as a burst and does not wait any response from the sink. Therefore, the result of transmission is not known by the sender. So the proposed protocol in this thesis is not a reliable protocol either. The necessity of connectionless communication and the other details are denoted in Section 4.1.

Another important feature of SUIT is that it is a cross layer protocol. Upper layers use the necessary information from the lower layers. For example, the transport layer uses the next hop sensor information from the routing layer and the application layer uses the congestion level information from the transport layer. The proposed cross layer design in this thesis is defined in Section 4.4.

4.1. Connectionless Communication

In computer networking, most of the multimedia streaming applications use a UDP based protocol instead of TCP [74], because TCP congestion control and retransmission mechanisms affect the streaming performance. Like UDP, SUIT provides connectionless communication. Thus it does not establish a session before sending data. The basic idea behind connectionless transmission is delivering the video stream as fast as possible. SUIT is designed for transporting non-continuous short video data. When a sensor is ready to send data, it should immediately start transmitting because the connection establishment may take long time in WMSNs. For example, if the sensor attempts to establish a session when a target enters its FoV, target may leave the FoV before establishing a session.

SUIT does not require confirmation message of a receiver about the sent packet. Therefore, it provides unreliable transmission. The receiver (the sink node) can detect the missing packets by monitoring the sequence number of the incoming packets but it does not inform the sender about the transmission result. Generally reliable transport protocols send ACK/NACK based feedback packet to the source. However, for streaming data traffic, this method is not appropriate. The source sensor cannot wait for the response of the sink before sending packets or frames. For this reason, packet based reliability cannot be applied for multimedia streaming. Instead of providing guaranteed delivery of all packets, connection between the sender and receiver can be monitored via control packets. For continuous video streaming, status of the connection should be known to detect errors during streaming and to provide better QoS. For example, video conferencing applications check the connection speed in order to adapt incoming and outgoing video resolution. However, SUIT does not monitor the connection either. There are two reasons for not taking into consideration the status of the connection:

- It is assumed that the predominant captured video period is very short so that all of the data may be completely transmitted from the source before getting a control message from the sink.
- A novel congestion mitigation technique which is proposed in Section 5.2 makes

it possible to decrease the video quality on the fly.

In summary, SUIT sends all the video data as a burst when the target is detected in the FoV. It provides connectionless communication and does not provide packet reliability. In case of the congestion, image frame quality is decreased on the intermediate sensors by the proposed congestion mitigation technique explained in Section 5.2.

4.2. Packet Ordering

In WSN applications, packets may be received out of order due to multihop topology of the network. Packets of the same sensor may have different paths because of the routing layer decision. Therefore out of order delivery should be handled for WMSN applications. SUIT is a connectionless protocol, however, unlike UDP, it has a mechanism for reordering packets.

In order to display an image, SUIT waits for all packets of a frame for sequentially encoded JPEG images. If the JPEG is progressively encoded, the application can

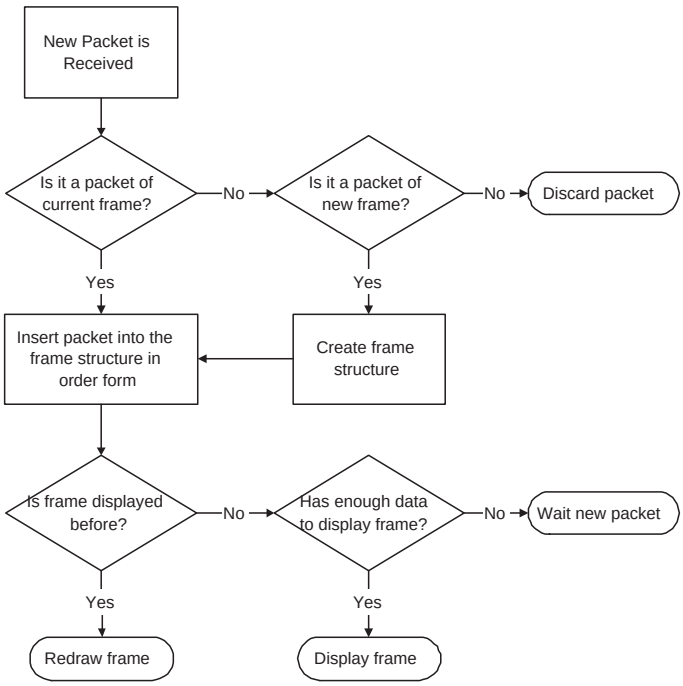


Figure 4.1. SUIT's out of order packet handling.

display each received sub-frame gradually. Thus a frame or sub-frame can only be displayed after the whole packets of the relevant frame or sub-frame are received.

Every SUIT packet has a source id, frame number and a sequence number which is increased by one for each created packet. In the application layer, SUIT tracks the sequence number of the incoming packets. When all of the packets of a frame (or sub-frame if image is progressive) are received, the packets are ordered by the application and then displayed. If the packet of a previous frame is received after processing a new frame, it is discarded because a new frame is already displayed and this packet becomes useless. In Figure 4.1, the out of order packet handling mechanism of SUIT is depicted.

4.3. Packet Prioritization

The buffer management in transport layer is an important issue in WMSNs because it has a considerable impact on the overall network efficiency and QoS which is dependent to the application requirements. For video surveillance applications, the primary requirement is delivering the video frames as fast as possible. SUIT has only one type of packet (video fragment packet) so there is only one queue at the transport layer instead of having a couple of prioritized queues. SUIT assigns a score for each packet for prioritizing them to improve QoS performance. When the transport layer buffer is full, the packets which have the lowest score are dropped. Score of the packet is calculated by using the following parameters:

- *Current hop count of the packet* is the number of sensor nodes that the packet has visited between the source node and the current sensor node.
- *Average delay of the packet* is equal to total delay divided by the current hop count. The total delay is the period elapsed between the arrival time of the packet from the source and the received time of the packet by the current sensor.
- *Frame index of the packet* determines the frame sequence number. Whenever the target is detected by the sensor, sequence number starts with zero and increases by one.

SUIT calculates a *weight* for each parameter by using Equations 4.1, 4.2, and 4.3. SUIT also allows disabling the buffer management. If it is disabled and the queue is full, all incoming packets are dropped regardless of the information they are carrying.

Current hop count value is embedded to the packet header. This value is assigned as zero when the packet is created at the source. At each hop, the receiver node increases the current hop value by one. Thus intermediate nodes are aware of the number of nodes which are visited by the packet before. The packet which visits more intermediate nodes until it reaches the sink has higher probability of drop-out than the packet which visits fewer sensors between the source and the sink. Therefore SUIT gives a higher priority to the packets whose current hop count is higher than the others. Weight of current hop count (W_h) can be 1, 2, 3 or 4. Depending on the maximum hop count that the topology allows, the possible hop-count value space is divided into four intervals using the points h_1, h_2, and h_3 where $h_1 < h_2 < h_3$. The (W_h) is calculated with the following formula:

$$W_h = \begin{cases} 1 & \text{if } hop \leq h_1 \\ 2 & \text{if } h_1 < hop \leq h_2 \\ 3 & \text{if } h_2 < hop \leq h_3 \\ 4 & \text{if } h_3 < hop \end{cases} \tag{4.1}$$

Similar to the hop count, total delay of the packet is also embedded into the packet header. Before sending a packet, sensor nodes calculate the interval between the time it is received and the time it is ready to send. Then sensor nodes add the calculated time period to the predefined total delay. As a result, each packet includes the total time that it spends on the intermediate sensors. The average delay can be calculated easily by dividing the total delay by the current hop count. The delay is an important metric for the overall efficiency of the network. The packet which has less average delay provides better performance for video surveillance applications. If the network is congested and the queue is full, transmitting the packets with low delay instead of the packets with high delay decrease the overall latency. Therefore SUIT

gives a higher priority to the packets whose average delay is less than the other packets. Weight of the average delay (W_d) can be 0, 1, 2 or 3. Depending on the maximum bandwidth of the sensor nodes, the possible average delay value space is divided into four intervals by using the points d_1, d_2, and d_3 where $d_1 < d_2 < d_3$. Then (W_d) is calculated with the following formula:

$$W_d = \begin{cases} 3 & \text{if } average\ delay \leq d_1 \\ 2 & \text{if } d_1 < average\ delay \leq d_2 \\ 1 & \text{if } d_2 < average\ delay \leq d_3 \\ 0 & \text{if } d_3 < average\ delay \end{cases} \tag{4.2}$$

For surveillance applications, Durmus *et al.* show that the initial frames of an event are more valuable, so they assign higher priority to the initial frames [75]. Therefore, we take into consideration to use frame index as another QoS metric. In order to distinguish packets based on their frame index, the source sensor embeds the frame index into the packet header. SUIT gives higher priority to the packets which is a fragment of the first frames of the detected target. The weight of frame index (W_f) can be one or two. While calculating W_f, a threshold F_{th} which can be changed depending

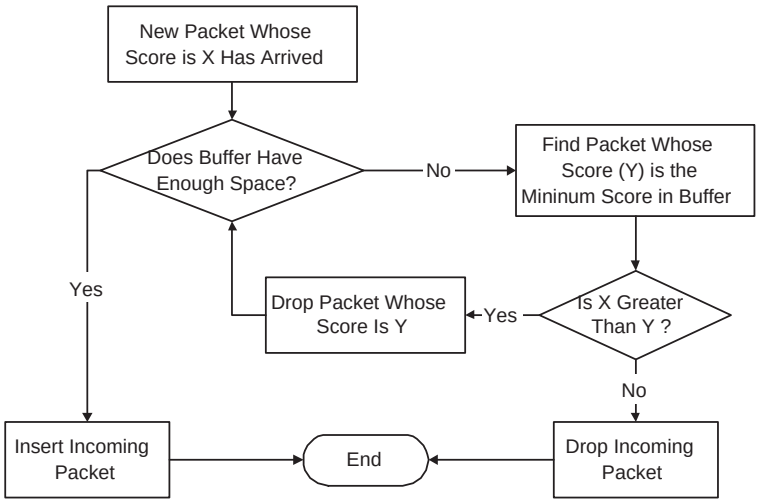


Figure 4.2. Buffer management in SUIT.

on the application requirements is used. W_f is calculated with the following formula:

$$W_f = \begin{cases} 2 & \text{if } frame\ index < F_{th} \\ 1 & \text{if otherwise} \end{cases} \tag{4.3}$$

To assign a score to a packet, weights of each parameter are calculated and then multiplied with each other. The packet which has the minimum score is dropped when the queue is full. The overall process is illustrated in the flowchart given in Figure 4.2.

4.4. Cross-Layer Data Exchange among Layers

SUIT uses a cross layer approach to exchange information among layers. However, layering is not completely left, so that the advantages of the layered design which are discussed in Section 3.5 are still preserved. SUIT has its own application layer and transport layer but it does not provide a MAC and routing layer. Any MAC and routing layer can be used under SUIT by extending them via a messaging module. SUIT can work with such protocols as long as they provide the necessary information which is mentioned later in this section.

The application layer is responsible for generating and viewing images. Images can be both SJPEG and PJPEG with or without restart markers. The source sensor is capable of fragmenting the image and the sink is capable of defragmenting it. In conventional layered designs, the application layer does not request information from the lower layers but in SUIT, the application layer gets the congestion level from the transport layer. According to the congestion level, it adapts the video rate as explained in Section 5.2.

One of the required features of the MAC layer is sharing and collecting sensor specific information in order to predict the status of the medium more accurately. In SUIT, the MAC layer should piggyback the fullness of the transport layer queue into each outgoing packet including RTS and CTS packets. The receiver sensor should

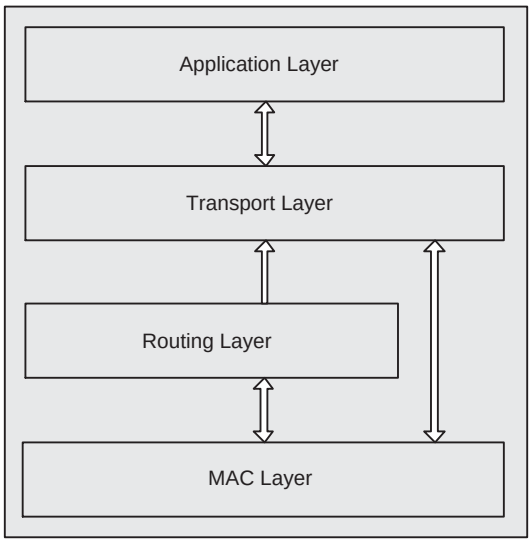


Figure 4.3. Cross-layer design of SUIT.

read the packet header at the MAC layer and generate a table which stores the buffer occupancy of the neighbors. During the fuzzy logic based congestion estimation, the transport layer uses the elements of this table as explained in Section 5.1. Moreover, the transport layer requests the ID of the next hop sensor node from the routing layer during the process of congestion estimation. As a result, the transport layer uses data from both MAC and routing layers.

In summary, the cross-layer design of SUIT creates interactions across layers by enabling flow of information on both sides. The main aim of this design is to achieve gains in the overall system performance without corrupting modularity. This does not leave the philosophy of layered designs but provides a messaging system among layers. In Figure 4.3, the block diagram of the proposed cross-layer design is illustrated.

5. CONGESTION CONTROL MECHANISM FOR SUIT

Congestion is one of the major problems in WMSNs. Since congestion causes buffer overflow that may lead to larger queuing delays and higher packet loss, it directly affects QoS [76]. Moreover it can lead to the transmission collision on MAC layer which requires retransmission of the colliding packets. Since wireless sensor nodes have limited battery, retransmission wastes the limited energy. Because of these drawbacks, congestion in WMSNs should be controlled efficiently.

The congestion control performs three basic operations which are *congestion detection*, *congestion notification* and *congestion mitigation* [76]. Among these, the most difficult operation is the congestion detection. An efficient congestion control mechanism should detect congestion immediately when it occurs. After detecting it, relevant sensor nodes should be notified to make them aware of the congestion. SUIT does not need a notification step because the detection is performed on each sensor instead of a central unit. Therefore SUIT performs only the congestion mitigation step after deciding that the network is congested.

SUIT proposes a congestion control mechanism which applies a fuzzy-logic based approach for the decision of the congestion level and then reacting to mitigate the congestion. In Section 5.1, the proposed fuzzy logic based congestion detection mechanism is described and then in Section 5.2, a novel congestion mitigation method is explained.

5.1. Fuzzy Logic Based Congestion Detection

In wired networks, the congestion detection is usually performed in an end-to-end manner. For these networks, the most common indicators of the congestion are packet delay and packet loss. However, the end-to-end approach and mentioned indicators are not appropriate for WMSNs. Radio interference or competing access to the radio medium may also cause packet loss in the wireless medium [8]. Therefore, a packet loss cannot be a definite reason for the congestion. Packet delay is calculated based on

the ACK message coming from the sink. Because of the multihop topology of sensor networks, the feedback of the sink cannot be received by the source sensor immediately. Thus, the source sensor is unable to catch instantaneous congestions by using the packet delay metric. As a result, using the packet delay information for congestion estimation is not practical either.

Because of not having an exact indicator of the congestion, it is a difficult task to detect congestion immediately when it occurs in WMSNs. The most commonly used indicator for WSNs is high buffer occupancy. However, monitoring a sensor node's queue may not be sufficient since multimedia data produces bursty high-load traffic in WSNs [3]. Especially in surveillance applications, video transmission causes local buffer to change instantaneously. In the performance evaluation of SUIT, it is observed that the buffer occupancy may change momentarily while transmitting a video of 3 to 5-seconds length over sensor nodes. As a result, only the fullness of buffer at a time cannot always give the correct congestion level.

SUIT proposes three indicators to be used for the congestion decision. These indicators are:

- *Ratio of incoming to outgoing packets within a time window* seems like the buffer occupancy metric, but it is more accurate than the buffer occupancy. The buffer size at a time can change instantaneously which may cause inaccurate results. Here, the traffic rate is calculated based on a sliding window. Hence, it overcomes the momentary changes in traffic and leads to more stable results than the instantaneous buffer occupancy.
- *Number of contenders* stands for the number of neighbors which have packets in their queue waiting to be sent. If there are too many contenders, all senders should compete with each other. In this case, the collision probability is higher. Therefore, the number of contenders is an effective indicator for congestion.
- *Buffer occupancy of parent (next-hop) sensor node* is an important metric because of providing information about the environment. If the buffer fullness of the next hop sensor node is high, the congestion probability will also be high.

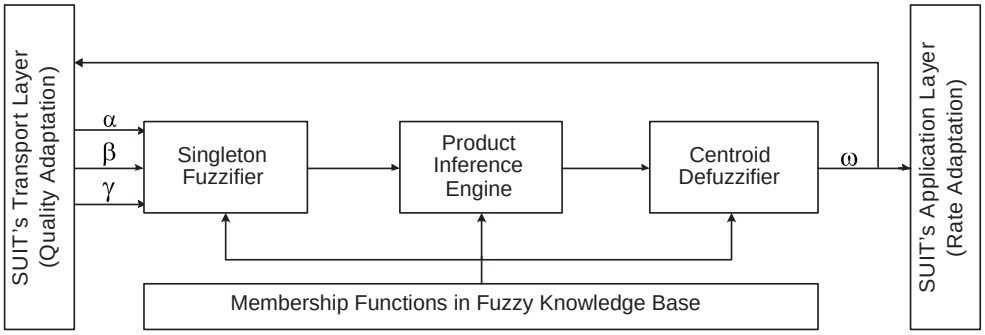


Figure 5.1. Fuzzy inference system of SUIT.

SUIT uses a FLS, in order to map these three congestion metrics into a single value. FLS is generally used for mapping uncertain variables from complex real world to a scalar data [48]. The components of the proposed FLS are singleton fuzzifier, product inference engine and centroid defuzzifier. In Figure 5.1, these components are depicted.

Since FLS uses three variables, the fuzzy set consists of three fuzzy variables as:

$$A = (\alpha, \beta, \gamma) \tag{5.1}$$

where α is the number of contenders, β is the buffer occupancy percentage of the parent (next-hop) sensor node and γ is the fuzzy variable term for the traffic load. The traffic load is the ratio of incoming packets to outgoing packets, and is therefore defined as:

$$\gamma = \frac{\gamma_{in}}{\gamma_{out}} \tag{5.2}$$

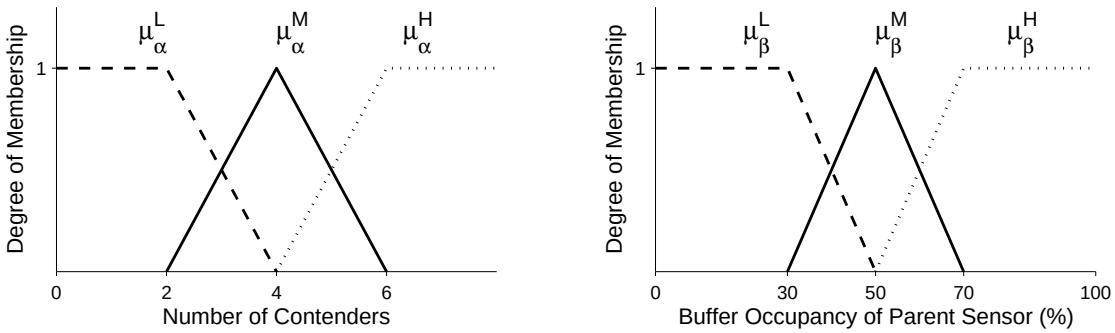
The ratio of incoming packets to outgoing packets is provided by the MAC layer. It is widely used for the congestion estimation in WMSNs [77]. However, it only provides node-level information. Variables from other nodes should be considered for better estimation. Therefore, SUIT uses the buffer occupancy rate of the next-hop sensor and the number of contenders which facilitate to have better awareness of the congestion in the network level. The buffer occupancy percentage information of the

neighbors is stored at the MAC layer in a table called Neighbor Buffer Occupancy Table (NBOT). In order to share the current buffer occupancy condition, the MAC layer gets this information from the transport layer and inserts it to each packet as a piggyback. When any of the sensor nodes hears a packet, it updates the corresponding entry in its NBOT with the new value regardless of the received packet's destination. By using NBOT, each sensor node can monitor the buffer occupancy levels of its next-hop node and the number of contenders (number of neighbors which have packets to send).

Inputs and outputs of the FLS are non-numeric linguistic variables. Instead of using numerical values, FLS uses variables from natural language. Our FLS has three linguistic variables named LOW (L), $MEDIUM$ (M), and $HIGH$ (H) which determine the congestion level of the system.

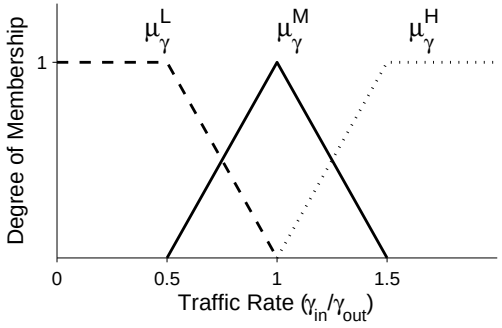
Membership functions are defined to be used in fuzzification and defuzzification steps. We use membership functions to quantify a linguistic term. For each fuzzy variable, a set of membership functions are defined, and this set consists of different functions for each linguistic term. In our case, we have three membership function sets and each of the sets has a different function for L, M and H linguistic variables. Membership functions can be in different forms such as triangular, trapezoidal, piecewise linear, Gaussian, or singleton [48]. We are using triangular shapes due to the ease of computation. In Figure 5.2, membership functions for the fuzzy variables α, β and γ are depicted.

The first step of our FLS is the fuzzification step. The fuzzification is the process of transforming crisp values into fuzzy values by using membership functions which associate a grade to each linguistic term. The proposed FLS uses a singleton fuzzifier to map each crisp value to a fuzzy value using membership functions. Since FLS has three linguistic variables, the fuzzifier defines three fuzzy classes named L, M, and H



(a) Membership Function for α

(b) Membership Function for β



(c) Membership Function for γ

Figure 5.2. Membership functions for the fuzzy variables α, β and γ.

for each fuzzy variable:

$$F : (\alpha, \beta, \gamma) \rightarrow (\alpha^f, \beta^f, \gamma^f) \tag{5.3}$$

$$\chi^f = \left[\mu_\chi^L(\chi), \mu_\chi^M(\chi), \mu_\chi^H(\chi)\right] \tag{5.4}$$

where χ represents the variables α, β and γ. Fuzzy values are calculated via membership functions of the fuzzifier which are presented in Figure 5.2. For example, if the number of contenders, α, of the sensor node is 5, then the corresponding α^f is [0,0.5,0.5] since $\mu_\alpha^L(5)$=0, $\mu_\alpha^M(5)$=0.5 and $\mu_\alpha^H(5)$=0.5 as can be seen in Figure 5.2a. If the buffer occupancy of the next-hop sensor node, γ, is 40, then the corresponding γ^f is [0.5,0.5,0] since $\mu_\gamma^L(45)$=0.5, $\mu_\gamma^M(45)$=0.5 and $\mu_\gamma^H(45)$=0 as can be seen in Figure 5.2b. If the traffic rate , β, of the sensor node is 1, then the corresponding β^f is [0,1,0] since $\mu_\beta^L(2)$=0, $\mu_\beta^M(2)$=1 and $\mu_\beta^H(2)$=0 as can be seen in Figure 5.2c.

The second and the most important step of our FLS is the inference step. Inference is used to combine fuzzy rules from the fuzzy rule base which is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion. Since there are three fuzzy variables and three fuzzy classes for each fuzzy variable, the rule base contains 3^3=27 different rules. The output of each rule is calculated by the rule evaluation method (REM) and results are mapped to a consequence class c. The proposed REM is similar to the one in [54] except for prioritizing the fuzzy variables. In [54], a weighted voting mechanism is used to prioritize fuzzy variables. In the proposed REM, all of the variables have the same effect for congestion decision. Therefore, all of the fuzzy variables have the same importance. The weights of the fuzzy classes for α, β and γ are set as 1 for L, 2 for M and 3 for H. The output of a rule r, θ_r, is calculated by summing up the weight of f_α^r, f_β^r and f_γ^r.

Table 5.1 shows the rule base of our fuzzy inference system (FIS), where the θ column corresponds to the output values of the rules which are calculated using the REM and the c column corresponds to the consequence fuzzy classes determined by the unique values of θ. As shown in Table 5.1, θ value can be a number between 3 and 9, hence there are 7 consequence fuzzy classes. In Rule 1, all of the fuzzy variables α, β and γ are L, so the output of Rule 1 is assigned as 3. In Rule 27, all of the fuzzy variables α, β and γ are H, so the output of a Rule 27 is assigned as 9.

We use the product-inference rule in our FIS, since it allows a mathematical simplification in the defuzzification process and is commonly used in practice. Hence,

$$\mu_r = \mu_\alpha^{f_\alpha^r}(\alpha) \cdot \mu_\beta^{f_\beta^r}(\beta) \cdot \mu_\gamma^{f_\gamma^r}(\gamma) \cdot \mu_c^{f_c^r}(\theta_r) \tag{5.5}$$

where $\mu_c^{f_c^r}$ is the membership functions of the consequence fuzzy classes defined in Figure 5.3 and $\mu_\alpha^{f_\alpha^r}$, $\mu_\beta^{f_\beta^r}$ and $\mu_\gamma^{f_\gamma^r}$ are the membership functions in the rule r corresponding to the fuzzy classes of the fuzzy variables for α, β and γ respectively. For example, α is determined as L, β is determined as M, and γ is determined as H for Rule 6 in Table 5.1. Thus for Rule 6, $\mu_\alpha^{f_\alpha^r}(\alpha)$ is equal to $\mu_\alpha^L(\alpha)$, $\mu_\beta^{f_\beta^r}(\beta)$ is equal to $\mu_\beta^M(\beta)$, and $\mu_\gamma^{f_\gamma^r}(\gamma)$ is equal to $\mu_\gamma^H(\gamma)$.

Table 5.1. Fuzzy rules in the knowledge base.

Index	α	β	γ	c	θ
1	L	L	L	1	3
2	L	L	M	2	4
3	L	L	H	3	5
4	L	M	L	2	4
5	L	M	M	3	5
6	L	M	H	4	6
7	L	H	L	3	5
8	L	H	M	4	6
9	L	H	H	5	7
10	M	L	L	2	4
11	M	L	M	3	5
12	M	L	H	4	6
13	M	M	L	3	5
14	M	M	M	4	6
15	M	M	H	5	7
16	M	H	L	4	6
17	M	H	M	5	7
18	M	H	H	6	8
19	H	L	L	3	5
20	H	L	M	4	6
21	H	L	H	5	7
22	H	M	L	4	6
23	H	M	M	5	7
24	H	M	H	6	8
25	H	H	L	5	7
26	H	H	M	6	8
27	H	H	H	7	9

The last step of the proposed FLS is the defuzzification step. Defuzzification is a process of transforming fuzzy output inferred from the fuzzy inference system (FIS) to crisp value. We use a centroid defuzzifier in the defuzzification step. The centroid defuzzifier returns the center of the inferred fuzzy outputs. The centroid calculation is the most popular defuzzification method which returns the center of the area under the curve [78,79]. However, calculating the center of the gravity of a trapezoid is a complex process and not appropriate for the sensor devices due to their limited processing capability and battery. In order to decrease the complexity of the defuzification step, we determine the membership function of each consequence fuzzy class as an isosceles triangle whose abscissa value of its apex is equal to the θ value of the corresponding

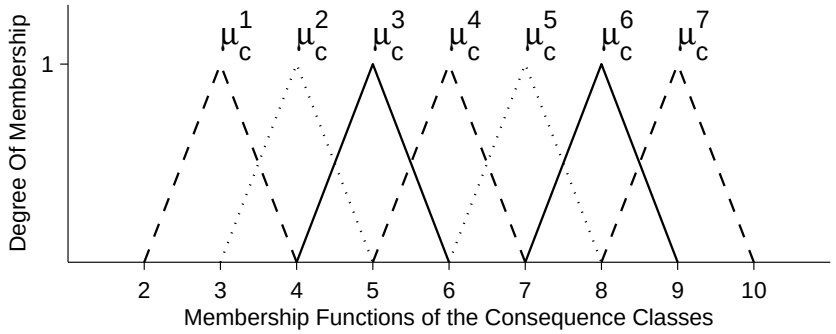


Figure 5.3. Membership functions of the consequence classes.

consequence class. In Figure 5.3 membership functions of the consequence classes are depicted.

Output of the centroid defuzzifier can be calculated easily by using Equation 5.6.

$$\omega = \sum_{r=1}^{27} \theta_r \mu_r \Big/ \sum_{r=1}^{27} \mu_r \tag{5.6}$$

where θ_r is the output of the rule r and μ_r is the output product-inference engine. Defuzzification is the final step in FLS. After applying the centroid defuzzifier, the output of the FLS, ω, becomes a crisp value between 3 and 9.

5.2. Congestion Mitigation

SUIT classifies ω into three classes as *Not Congested* (NC), *Slightly Congested* (SC), *Fairly Congested* (FC) according to Equation 5.7.

$$L = \begin{cases} Not\ Congested & \text{if } 3 \leq \omega < 5 \\ Slightly\ Congested & \text{if } 5 \leq \omega \leq 7 \\ Fairly\ Congested & \text{if } 7 < \omega \leq 9 \end{cases} \tag{5.7}$$

SUIT does not apply congestion mitigation for the NC level because in this case sensor nodes have enough capacity to handle the network traffic. SUIT congestion

mitigation algorithm is activated if the network is SC or FC. Our mitigation algorithm has two major procedures to decrease the congestion. These procedures are rate adaptation and quality adaptation.

Rate adaptation takes place in the source sensor which has a target in its FoV and generates image frames. The source sensor generates video in the application layer based on the congestion level which is calculated in the transport layer. Since SUIT is a cross layer protocol, the application layer can request the required data from underlying layers. Application layer can define three different video rates according to three congestion level classes defined in SUIT. Depending on the sensors' data rate and encoded image size, the choice of rates should be configured by the application. A generic function for the rate adaptation is given in Equation 5.8.

$$r = \begin{cases} r_1 & \text{if } L = NC \\ r_2 & \text{if } L = SC \\ r_2 & \text{if } L = FC \end{cases} \tag{5.8}$$

where L is the congestion level and r is the video rate of application. r_1, r_2, and r_3 are the video rates defined by the application and their values should be as $r_1 > r_2 > r_3$.

Quality adaptation takes place on the intermediate nodes on the fly. To the best of our knowledge, this technique has been firstly proposed by SUIT. As far as we know, the quality adaptation on the fly is not applied for WMSNs before. We use PJPEG's structural advantage in this procedure. As explained in Section 3.6, PJPEG comprises a number of scans (subframes). If some packets of a sub-frame are dropped or corrupted, only the corresponding sub-frame becomes useless instead of the whole image. In our novel quality adaptation technique, we intentionally drop the packets of some subframes of PJPEG. Depending on the congestion level, SUIT drops packets of a randomly selected sub-frame. In order to distinguish subframes of the image, the SUIT packet has two types of information in its packet headers. These data are the number of total subframes that the related frame has, and the sub-frame index that the packet belongs to.

SUIT can perform quality adaptation even if the image is not completely received. So it does not need to have all packets of the image before decreasing the quality. In Figure 5.4, effects of quality reduction are illustrated with respect to the size and the PSNR metrics. In the comparison, an example PJPEG which has 10 subframes is used. To decrease the quality, the subframes from 5 to 10 are dropped. As shown in Figure 5.4c, the resulting image size is decreased about 73% compared to the original image. Predictably the resulting image quality also decreases as shown in Figure 5.4d. However the client can still view a low quality image which is better than losing the whole image frame.

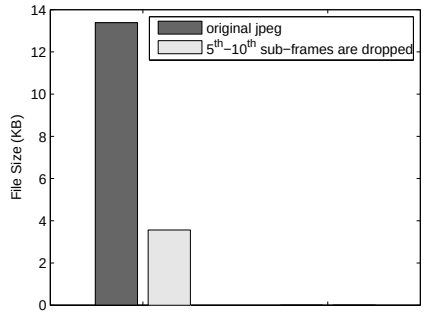
The quality adaptation takes place on the intermediate nodes on the fly. Decreasing the quality is the responsibility of the transport layer. Thus, the application layer always generates full-quality images. In our quality adaptation technique, all PJPEG images are processed starting from the lowest score to highest score (scoring was ex-



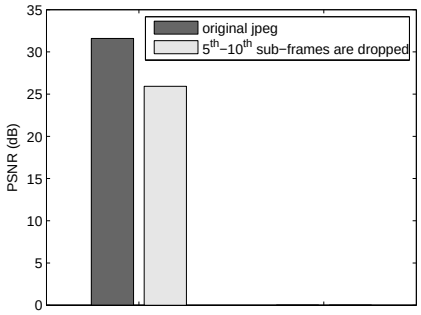
(a) Original PJPEG



(b) 5^{th} to 10^{th} subframes are dropped



(c) Quality Reduction Effects on File Size



(d) Quality Reduction Effects on PSNR

Figure 5.4. Quality reduction effects on file size and PSNR.

plained in Section 4.3). In this process, subframes of the PJPEG images are randomly selected and dropped. According to the application requirements, less subframes are dropped for the SC case while more subframes are dropped for the FC case. However, the quality cannot be decreased more than a threshold defined for the application. This quality adaptation process is ended when adequate space in the buffer becomes available. If there is still not enough space in the buffer after all images are processed, images are dropped according to the packet prioritization policy.

In summary, in order to mitigate congestion, SUIT performs rate adaptation at the source node and quality adaptation on the intermediate nodes. The rate adaptation is commonly used by WSNs applications, but quality adaptation is a novel technique which can be applied for PJPEG images while they are being transmitted. The overall congestion mitigation process is illustrated in the flowchart given in Figure 5.5.

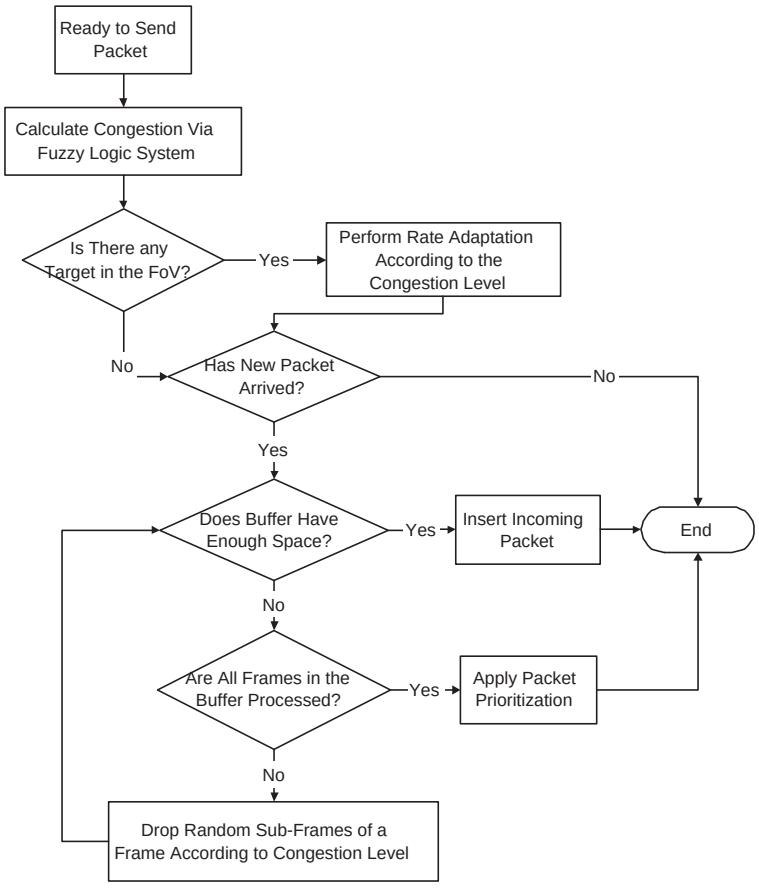


Figure 5.5. Operation of congestion mitigation algorithm.

6. PERFORMANCE EVALUATION OF SUIT

In this section, the performance of the SUIT protocol is evaluated. The performance evaluation is composed of two sections. Firstly, JPEG encoding techniques are compared to each other and the best encoding technique is determined for SUIT. Secondly, performance of SUIT's congestion control is evaluated with extensive simulations and it is compared to two different congestion detection and mitigation techniques. These competitors use ESRT [23] and FCE [25] based congestion detection and rate adaptation as the congestion mitigation technique.

6.1. System Model and Simulation Parameters

OPNET [80] network simulation tool is used for the implementation and simulation of the SUIT protocol. We set up an intrusion detection application scenario for the simulations. In this scenario, sensors have a camera extension which enables the detection of moving targets. Targets are detected with the model depicted in Figure 6.1 [1]. A sensor node creates video frames while the target is moving between the points A and B, FoV. The time spent between A and B is called the dwell time and can be calculated as:

$$T_{dwell} = \frac{d_{AB}}{\vec{V}} \tag{6.1}$$

where d_{AB} is the distance between A and B, and $\vec{V}$ is the velocity of the target. Thus, a camera working with K frames per second (fps) creates a video stream of $K \cdot T_{dwell}$ frames in case of an intrusion.

In the simulations, video sensors are uniformly deployed in a square shaped terrain. There are multiple targets in the environment, which are moving according to the random waypoint mobility model [81]. The target is assumed to be a pedestrian whose velocity is 2 m/s. Sensor nodes use a binary detection model to detect targets.

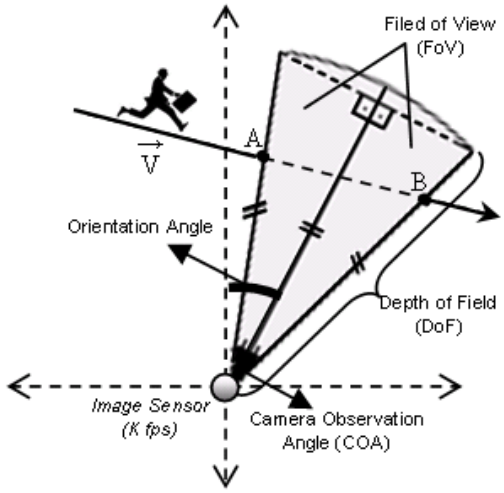


Figure 6.1. Camera detection model, taken from [1].

According to the binary detection model, the probability of sensing a target is equal to 1 if the target is in the FoV of the sensor node. When the target enters to the FoV, sensor node's camera starts generating video. The video used in the simulations is a sequence of Sub Quarter Common Intermediate Format (SQCIF) JPEG images which have a resolution of 128x96. For each sensor node, a sequence of prepared JPEG images is stored. If sensor nodes sense intrusion, they generate a JPEG image by picking one of the stored images.

JPEG images used in the simulation are generated by using the GNU Image Manipulation Program (GIMP) [82]. First off all, different bitmap images are prepared for the simulations and then they are converted to the JPEG images via GIMP. Three different encoding techniques are used in the conversion step. These techniques are (i) sequential JPEG (SJPEG), (ii) sequential JPEG with restart marker (SJPEG-RM) and (iii) progressive JPEG (PJPEG). File sizes of the images which are encoded with different parameters are very similar. The average file sizes and other features of the images are shown in Table 6.1.

JPEG segments are the basis of SUIT protocol. SUIT implements a JPEG parser in order to extract each segment properly. In Figure 6.2a, segments of the SJPEG are depicted. The segments which are shown in dark gray are critical segments. These

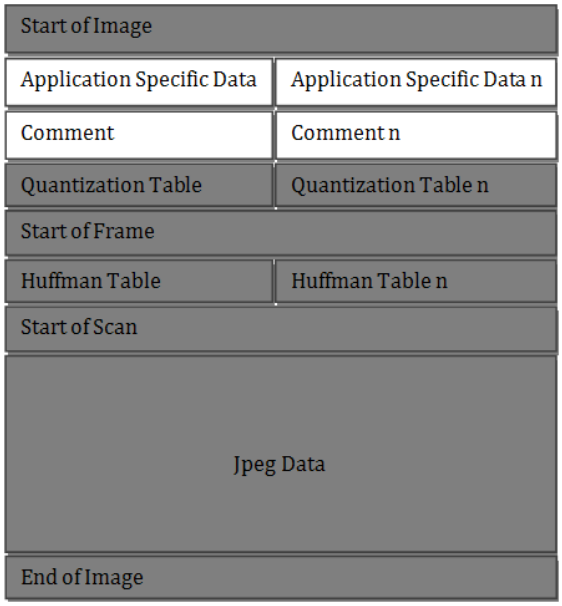
Table 6.1. Properties of the JPEG files used in the simulation.

Type	**Resolution**	**Average size**	**PSNR**	**# of frames**	**RM position**
SJPEG	128x96	9.2764 KB	35.9221	1	-
SJPEG-RM	128x96	9.2946 KB	35.9221	1	end of each row
PJPEG	128x96	9.1487 KB	35.9221	10	-

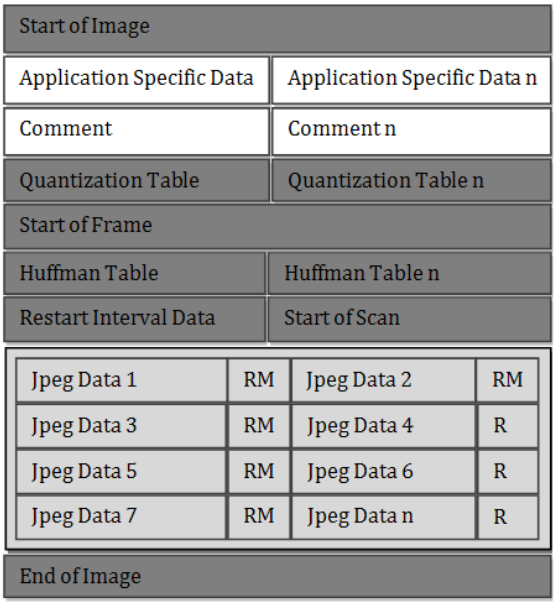
segments should not be corrupted. Otherwise, the whole frame is lost. A JPEG image can include a variable size application specific data segment and a comment segment. These segments are shown in white boxes. The application data segment includes metadata of the encoder application such as the thumbnail image. The comment segment includes comments of the user or encoder application. Since both of these segments do not have an effect on the image quality; they are ignored by the image parser and therefore they are not transmitted. In Figure 6.2b, segments of the SJPEG-RM are depicted. In this type of JPEG, there are still some critical segments. However, the biggest part of the image is not critical. Segments shown in the light gray are not critical. If one of these blocks is corrupted, the image is corrupted up to the next restart marker. In Figure 6.2c, segments of the PJPEG are depicted. Unlike the SJPEG, PJPEG has a good packet loss resilience. If any part of the image data inside the sub-frame block is lost, only relevant sub-frame is unusable instead of the whole image.

After extracting the segments, SUIT generates packets which have a maximum size of 1024 bytes. Therefore, the segments whose sizes are bigger than 1024 bytes are fragmented into packets. Also the segments whose total sizes are smaller than 1024 bytes are merged into a single packet. However, SUIT merges the packets regarding their importance level. The packets which have the same importance level can be merged. The importance level is used in the MAC layer for deciding the number of retransmissions in case of a transmission failure. There are three importance levels defined in the protocol:

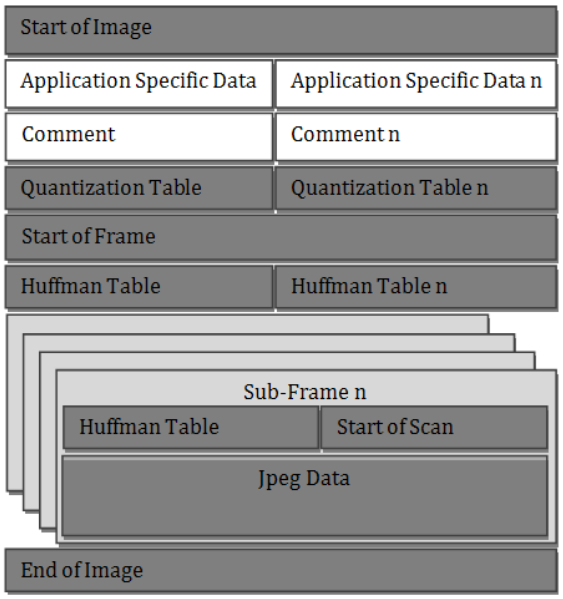
- Critical segments of the JPEG image which are not located in subframes have



(a) SJPEG Segments



(b) SJPEG-RM Segments



(c) PJPEG Segments

Figure 6.2. Segments of the JPEG images.

the highest importance

- Critical segments of the subframes have the intermediate level importance.
- Segments of the remaining JPEG data have the lowest level importance.

Another important task of the SUIT is preparing packet groups for message

passing on the MAC layer. In order to decrease the amount of contention in the MAC layer, the medium is reserved for a group of packets instead of every single packet. SUIT prepares a group of packets for each JPEG image. Thus, in the simulations a JPEG image generates one group which includes several packets.

Diff-MAC [1] protocol is used as the MAC protocol. In the simulations, GPSR [50] is used to handle the network layer operations on the CSMA/CA based Diff-MAC access with RTS/CTS and acknowledgments. It provides a sleep-listen schedule to conserve energy and does not require any synchronization between the neighboring sensor nodes because each node follows its own independent schedule. Diff-MAC is a QoS-aware and priority-based MAC protocol for WMSNs. However in the simulations, the QoS awareness of the Diff-MAC is not used. All SUIT packets are considered as video packets. In order to use Diff-MAC with SUIT, the following modifications are applied:

- An inter-layer communication module is implemented in order to exchange information with other layers.
- A module is implemented in order to construct neighbor buffer occupancy table (NBOT). This module piggybacks the buffer occupancy information into the outgoing packets header and reads this information from each packet that the sensor can hear regardless of the packet destination.
- A message passing module is implemented in order to classify the group of packets which are specified by SUIT while generating packets.
- A module is implemented in order to distinguish SUIT packets with respect to their importance. It chooses different number of retries according to the importance level.

IEEE 802.15.4 compliant radio has low cost and low power attributes. Thus, these radios are widely used for the WSN implementation [83]. In the market, there are several radio modules which are compliant with IEEE 802.15.4 standard. Most of the radios of Atmel [84], Chipcon [85] and Microchip [86] provide 250 kbps data rate [87]. However, for multimedia transmission 250 kbps data rate is not adequate. Chulsung Park *et al.* designed a high data-rate wireless sensor node named as eCAM [88]. Their

sensor node provides 1 Mbps data rate and their camera module can operate as a JPEG compressed still camera. Also Atmel designed an IEEE 802.15.4 compliant single chip ATmega128RFA1 [89] which can provide up to 2 Mbps data rate. Since high data-rate transmission should be used for multimedia streaming applications, we determine the data rate as 2 Mbps in our simulations according to the capabilities of these example radios.

In order to simulate real life more accurately, a realistic link model proposed in [90] is used in the simulations. As the radio propagation model, Zuniga *et al.* use the log-normal shadowing model proposed in [91]. This model is given by:

$$PL(d) = PL(d_0) + 10n \log_{10}(d/d_0) + X_\sigma \tag{6.2}$$

where d is the distance between the transmitter and the receiver, d_0 is the reference distance, n is the path loss exponent, and X_σ is a zero-mean Gaussian random variable (in dB) with standard deviation σ. Based on the observed wireless channel features in [92], we used the most appropriate values for our simulation as $n = 3.0$, $\sigma = 8.4dB$, $d_0 = 1m$ and $PL(d_0) = 34.8$. Zuniga *et al.* used the Signal-to-Noise Ratio (SNR) at a distance d as:

$$SNR(d)_{\ dB} = P_{t\ dB} - PL(d)_{\ dB} - P_{n\ dB} \tag{6.3}$$

where P_t is the output power of the transmitter and P_n is the noise floor. The noise floor, which depends on both the radio and the environment, is studied in [92] and for MICA2 [93] motes, it is calculated as -105 dBm assuming an ambient temperature of 27 °C and no interference. Finally, the probability of successfully receiving a packet $p(d)$ at a distance d is given by the following formula:

$$p(d) = (1 - \frac{1}{2}\exp^{-\frac{SNR(d)}{2}\frac{1}{0.64}})^{8f} \tag{6.4}$$

where f is the packet size in bytes.

In simulations, 400x400 meters square-shaped environment where 256 video sensors are uniformly deployed is used. The communication range of the sensors is 80 meters and these sensors have a camera module whose detection range is 30 meters and observation angle is 52 degrees. There are one to seven targets which are moving in the environment according to the random waypoint mobility model. Other significant parameters used in our simulations are listed in Table 6.2.

In order to evaluate the performance of SUIT on a congested WMSN, we study the scenarios where congestion is generated on the relay nodes. In these scenarios, we make sure that the generated data traffic does not exceed the capacity of the source nodes. Therefore, the data load on the relay nodes is increased without dropping

Table 6.2. Simulation parameters for the evaluation of SUIT.

Parameter	**Value**
Number of Sensors	256
Area	400 m x 400 m
Sink Location	(0,400)
Number of Targets	1 to 7
Target Velocity	2 m/s
Target Mobility Model	Random Waypoint
Detection Model	Binary FoV
Camera Observation Angle	52 deg.
Depth of Field	30 m
Video Encoding	SJPEG, SJPEG-RM, PJPEG
Rate for NC case	2 fps
Rate for SC case	4 fps
Rate for FC case	6 fps
Average Packet Size	1 KB
Average JPEG image Size	9.2 KB
Buffer Size	100 KB
Channel Rate	2 Mbps
Communication Range	80 m
MAC Layer	Diff-MAC
Routing Algorithm	GPSR

packets at the source node. To achieve this, we increase the number of intruders on the deployment area and create video traffic over relay nodes from multiple source nodes.

6.2. Performance Evaluation of JPEG Encoding Techniques

SUIT protocol is designed for transmitting JPEG images over WMSNs. In this section, performance of the JPEG encoding techniques are evaluated and three encoding techniques are compared to each other. These techniques are SJPEG, SJPEG-RM and PJPEG. Properties of the files used in the simulations are given in Table 6.1. All of the files have the same quality and approximately the same size. Thus, the generated images at source sensor can be considered as equal in size and quality.

SUIT congestion mitigation approach provides dynamic frame rate with respect to the congestion level. If the congestion level is NC, the maximum possible frame rate is assigned for the source sensors. In the simulations, frame rate for NC level is assigned as 6 fps. For SC and FC levels, frame rates are assigned as 4 and 2 fps respectively.

The number of received frames at the sink is an important metric for QoS. In Figure 6.3, the average received frame rate at the sink is illustrated. The number of intruders (shown in the x-axis) is varied to increase the load, hence create congestion, in the network. The average received frame rate is calculated by dividing the total received frames to the simulation duration in seconds. As expected, SJPEG has the lowest performance due to its low packet loss resilience. If one of the packets of SJPEG image is lost, the relevant image is also lost. On the other hand, SJPEG-RM and PJPEG images can still be usable in case of data packet loss. Although SJPEG-RM and PJPEG have both high packet loss resilience, PJPEG has better performance due to the congestion mitigation technique of SUIT which is used for only PJPEG images. By applying this technique, the size of the images can be decreased and thus more PJPEG images can be transmitted with lower quality. Especially, when the network load is increased, SUIT can transmit more PJPEG frames than other encoding techniques. As shown in Figure 6.3, for the scenario which has six intruders, SUIT provides 29%

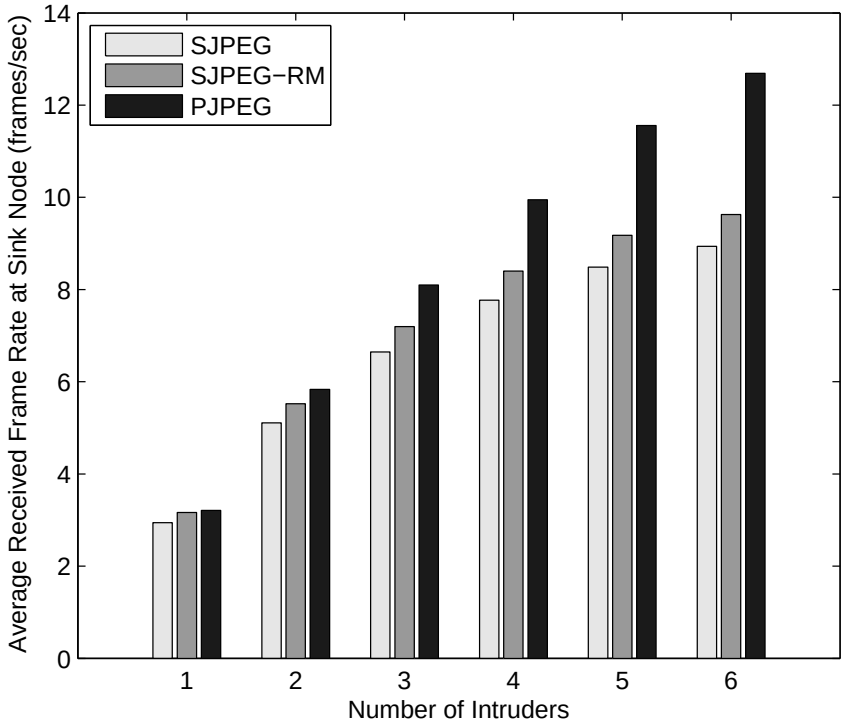


Figure 6.3. The average received frame rate at the sink.

more PJPEG delivery than SJPEG and 23% more PJPEG delivery than SJPEG-RM.

In Figure 6.4, the average frame delivery performance is depicted. When the network load is increased, the packet delivery ratio is decreased. For all of the network load cases, PJPEG provides more efficient frame delivery than SJPEG and SJPEG-RM. SJPEG provides the worst frame delivery performance because it has the lowest packet loss resilience. Therefore, it can be asserted that SJPEG is not appropriate for WMSN which does not provide a reliable medium. SJPEG-RM provides better packet loss resilience than the SJPEG, but it still provides poor performance compared to PJPEG. Figure 6.4 also shows the advantages of SUIT's quality adaptation technique on PJPEG. If the network is congested, SUIT decreases the quality of the PJPEG frames and therefore creates an opportunity for the congested nodes to drain their queues. If other encoding techniques are used, SUIT drops the incoming packets at the congested sensors. As a result PJPEG provides more efficient frame delivery than SJPEG and SJPEG-RM.

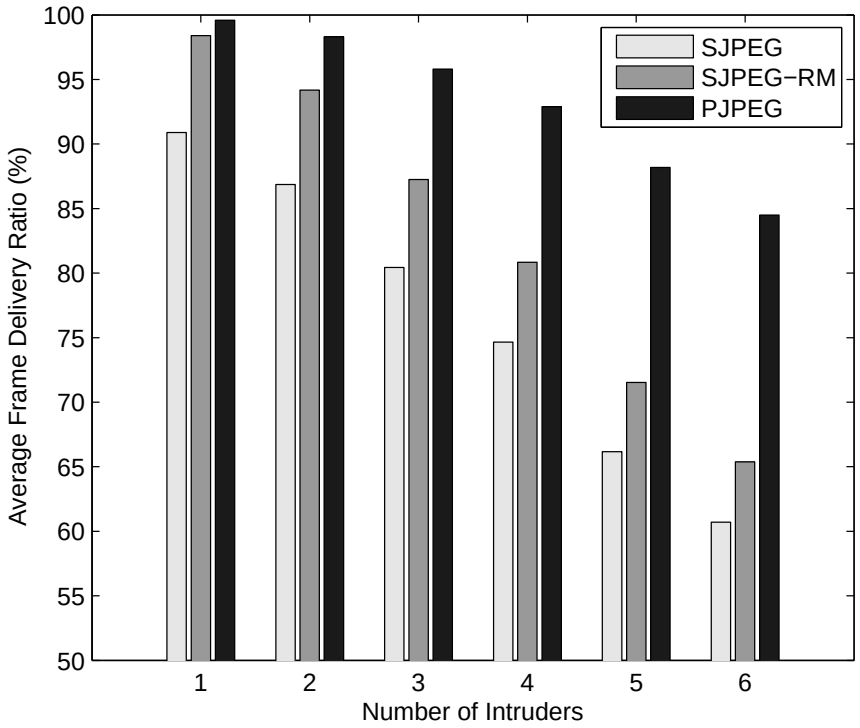


Figure 6.4. Average frame delivery ratio.

The average received frame latency results of the JPEG encoding techniques are provided in Figure 6.5. If the network load is low, the frame latency performances are similar to each other. However if there are more than two intruders, SUIT provides the best frame latency performance with PJPEG. Since the size of PJPEG images becomes smaller due to the SUIT congestion mitigation technique, images are transmitted in shorter periods. When it comes to SJPEG and SJPEG-RM, the average latency performance of SJPEG is better than SJPEG-RM because SUIT transmits a lower number of SJPEG images than SJPEG-RM images. While using SJPEG images, most of the received images which have missing packets are discarded. As a result, SUIT ignores these SJPEG images and this yields to a better latency performance for SJPEG.

The quality of SJPEG, SJPEG-RM and PJPEG are compared with respect to the peak signal-to-noise ratio (PSNR) values. PSNR is a well-known image quality metric and widely used. For two grey-level (8 bits) images f and g whose dimensions

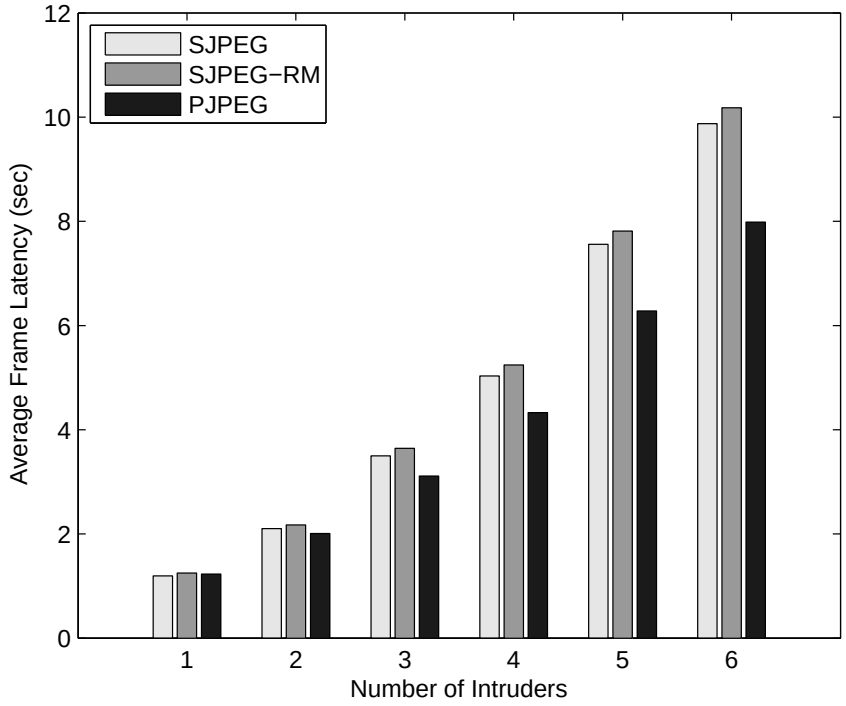


Figure 6.5. Average frame latency.

are $X \times Y$, PSNR calculation is defined by the following formula [94]:

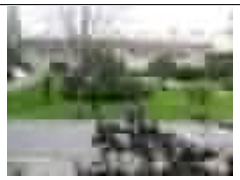
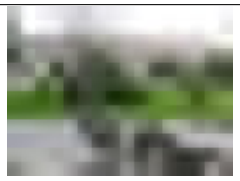
$$PSNR(f,g) = 10 \log_{10} \left(\frac{255^2}{MSE(f,g)} \right) \tag{6.5}$$

where

$$MSE(f,g) = \frac{1}{X \cdot Y} \sum_{i=1}^{X} \sum_{i=1}^{Y} (f_{ij} - g_{ij})^2 \tag{6.6}$$

In Table 6.3 and Table 6.4, visual examples of PSNR values are given for PJPEG and SJPEG-RM respectively. The images shown in the tables are selected randomly from the image set which is received by the sink sensor. If the sub-frame of PJPEG is corrupted, the image becomes blurry. On the other hand, if the restart marker data block of SJPEG-RM is corrupted, the corresponding line of the image becomes invisible. According to Table 6.3, the PJPEG images whose PSNR value is higher than

Table 6.3. PSNR values for PJPEG images.

Original Image	Received Image	PSNR
		34.1312
		31.4390
		23.7759
		21.4659
		18.4814

20 dB can be used for object detection algorithms at the sink side because the intruder can still be identified. Therefore these images are acceptable for the sink application. However, the images whose PSNR value is lower than 20 dB have poor quality and the intruder cannot be recognized. For SJPEG-RM, PSNR value of the image decreases to 20 dB even if a single line of the image is corrupted. Unlike the PJPEG case, the application cannot decide if the received SJPEG-RM image is acceptable or not. Because it cannot be known if the corrupted line contains picture of intruders. So there is a trade-off between PJPEG and SJPEG-RM. In case of packet loss, PJPEG provides the whole image with lower quality, but SJPEG-RM provides high quality image which has missing lines on it.

In Figure 6.6, the average PSNR value of the received images are depicted. Since

Table 6.4. PSNR values for SJPEG-RM images.

Original Image	Received Image	PSNR
		20.3143
		17.3836
		15.4684
		12.2377

all of the received SJPEG images do not have any packet loss, SJPEG has the highest PSNR value which is around 35 dB for the image set used in the simulations. PSNR value of SJPEG-RM is observed as approximately 34 dB for all cases whereas PSNR value of PJPEG is observed as starting from 34.5 dB and it decreases as the network load increases. However, for the worst case (if there are six intruders), the average PSNR value of PJPEG images is observed as higher than 33 dB which is still a good quality for the application.

In summary, the proposed congestion mitigation technique can transmit higher amount of images more efficiently by decreasing the quality on the fly. This technique can be applied with PJPEG images due to PJPEG's structure. Therefore, PJPEG comes out on top with respect to the frame sending rate, the frame receiving rate, the frame delivery and the frame latency performance. For the received image quality performance, PJPEG can transmit lower quality images in case of congestion. It is shown in Figure 6.3 that the PSNR value which is higher than 20 dB is an acceptable

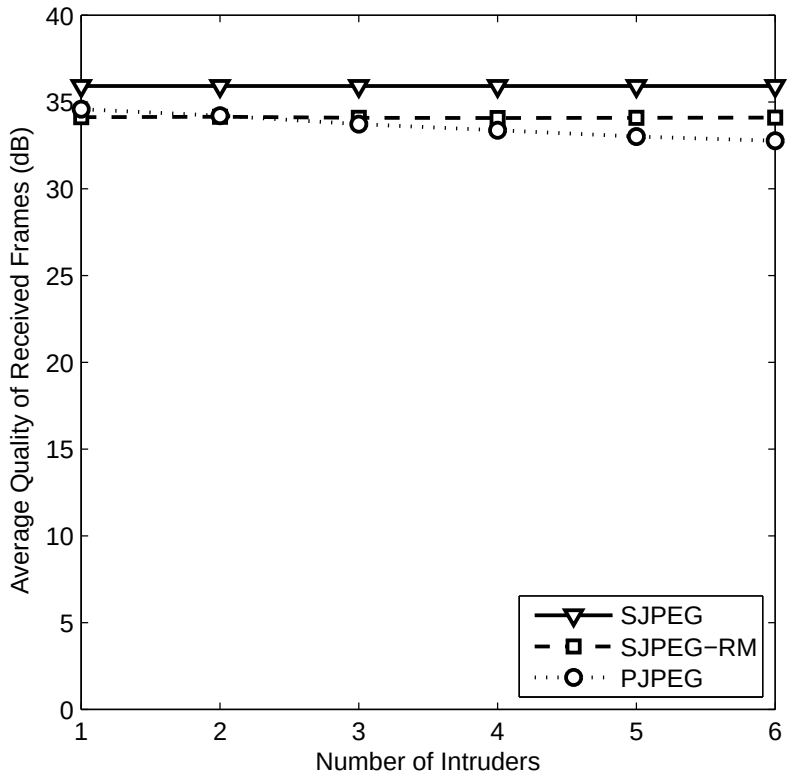


Figure 6.6. The average quality of received frames at the sink.

quality. Thus the average quality of PJPEG images is acceptable. As a result, the progressive encoding is recommended for the SUIT protocol and the performance of SUIT is compared to the other protocols by using PJPEG images in Section 6.3.

6.3. Performance Evaluation of Congestion Mitigation Techniques

The performance of SUIT's congestion control mechanism is evaluated with extensive simulations and it is compared to two different congestion detection and mitigation techniques. These competitors are listed below:

- ESRT [23] based congestion detection and rate adaptation as congestion mitigation.
- FCE [25] based congestion detection and rate adaptation as congestion mitigation.

Basically, SUIT's congestion control provides fuzzy logic based congestion detection and two congestion mitigation techniques. One of the mitigation techniques is data rate adaptation and the other one is quality adaptation while packets are being transmitted over the sensor nodes. Since quality adaptation is used with PJPEG, we select PJPEG encoding for the simulations.

For the first competitor, ESRT [23] based congestion detection mechanism is selected. ESRT monitors the local buffer occupancy in order to estimate congestion. It uses a threshold to determine if the network is congested or not. However, we used two thresholds in order to adapt ESRT congestion detection to SUIT's rate adaptation process. We use the following rule for assigning congestion levels:

$$L = \begin{cases} \textit{Not Congested} & \text{if buffer occupancy} \leq 0.4 \\ \textit{Slightly Congested} & \text{if } 0.4 < \text{buffer occupancy} \leq 0.6 \\ \textit{Fairly Congested} & \text{if buffer occupancy} > 0.6 \end{cases} \tag{6.7}$$

FCE [25] uses a fuzzy logic based congestion estimation mechanism as explained in Subsection 2.1.7. FCE uses two fuzzy variables which are *net packet arrival rate* p, and *buffer occupancy* s. It assigns a grade value to its fuzzy set variables p and s via Equation 6.8 and Equation 6.9.

$$p = \begin{cases} 0 & \text{for } P_{net} < 0.5 \\ 0.5 & \text{for } 0.5 \leq P_{net} < 1 \\ 1 & \text{for } P_{net} \geq 1 \end{cases} \tag{6.8}$$

where P_{net} is the ratio of incoming packets to outgoing packets at a particular node.

$$s = \begin{cases} 0 & \text{for } S = 0 \\ 0.5 & \text{for } S \leq 0.6 \\ 1 & \text{for } 0.6 < S \leq 1 \end{cases} \tag{6.9}$$

where S corresponds to the percentage level of buffer fullness at a particular instance.

Table 6.5. Fuzzy logic table of FCE.

p \ s	**0**	**0.5**	**1**
0	0	0.5	1
0.5	0.5	0.5	1
1	1	1	1

FCE uses the outer-max product in the fuzzy inference step. Therefore, the fuzzy table for the estimation of congestion considering the current packet arrival rate and the buffer size are given in Table 6.5.

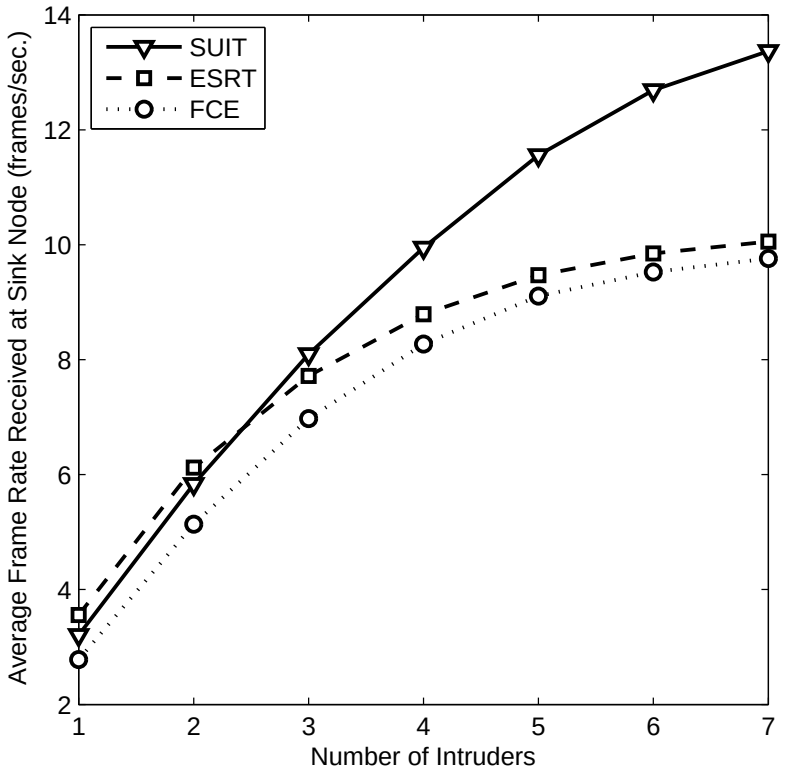


Figure 6.7. Average received frame rate at the sink.

We adapt FCE congestion estimation capability to SUIT's rate adaptation by using the following formula:

$$L = \begin{cases} Not\ Congested & \text{if FCE result} = 0 \\ Slightly\ Congested & \text{if FCE result} = 0.5 \\ Fairly\ Congested & \text{if FCE result} = 1 \end{cases} \tag{6.10}$$

One of the most important metrics for application QoS is the average received frame rate at the sink. In Figure 6.7, comparison of the average received frame rate is depicted. X-axis shows the number of targets which is varied to increase the load in the network. If the network is not congested, all protocols provide a similar frame rate, i.e. with 1-3 intruders. However when the network is getting congested, the SUIT protocol can transmit more frames than the other protocols. The reason for this result is the congestion mitigation technique of SUIT which performs quality adaptation on

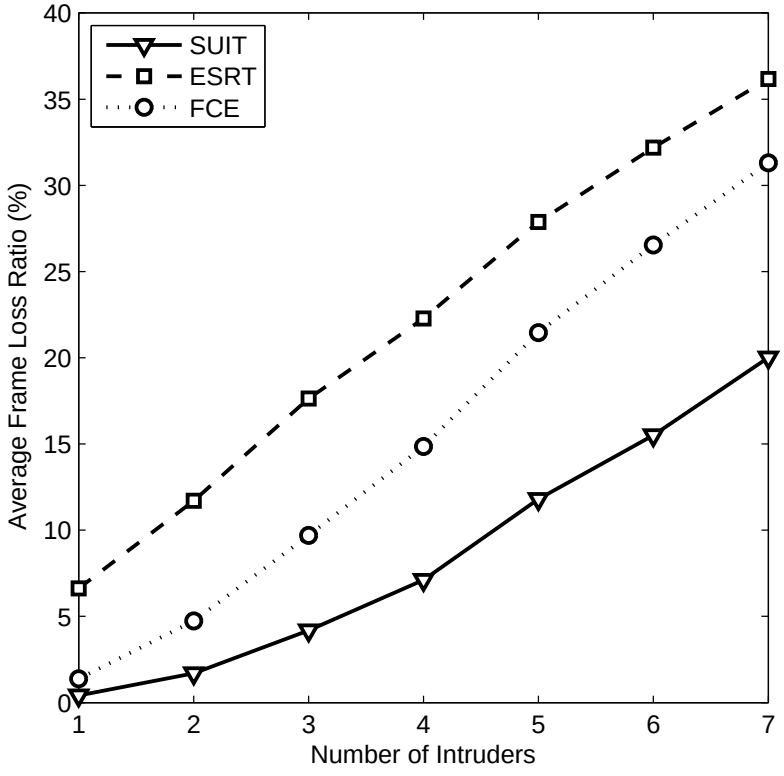


Figure 6.8. Average frame loss.

the fly. In a congested network, SUIT can decrease the buffer fullness by decreasing the quality of the images. However, other protocols have to drop incoming frames. As shown in Figure 6.7, without quality adaptation, the maximum frame rate that can be received by the sink is 10 fps, whereas SUIT allows the sink to receive more than 13 fps.

A protocol can transmit more frames if it generates excessive number of frames, but number of dropped frames should also be considered. So the frame loss ratio is another important QoS metric. In Figure 6.8, the frame loss performances of the protocols are given. For all of the cases, SUIT has better frame loss performance than ESRT and FCE based congestion detection protocols. ESRT has worse performance than FCE, because it calculates the congestion level in an optimistic manner and generates more frames than FCE. Therefore, compared to FCE, ESRT provides a better average received frame rate at the sink, but worse frame loss performance.

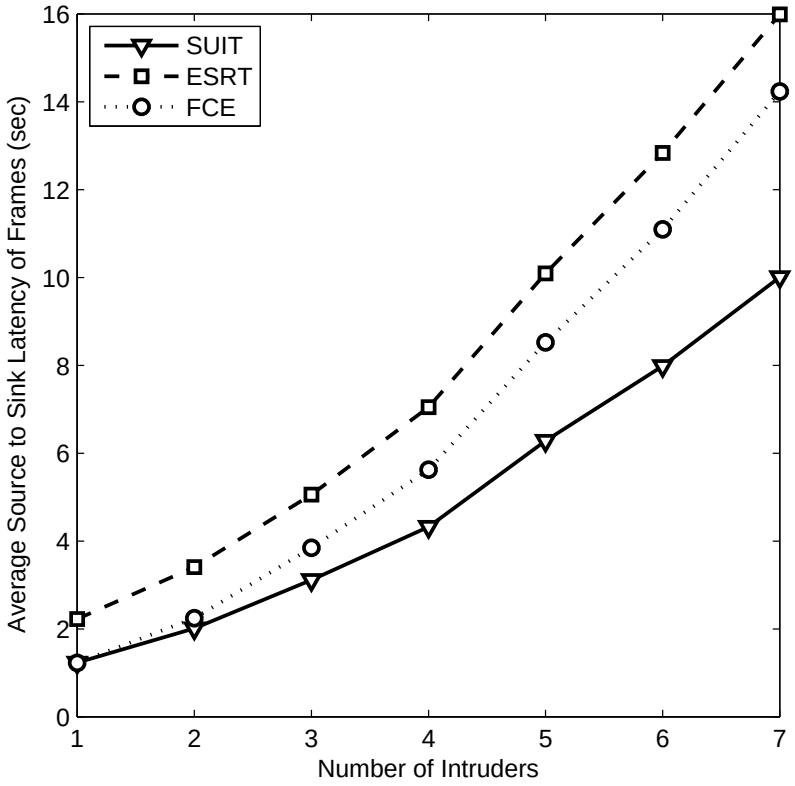


Figure 6.9. The average source to the sink latency of frames.

We compare the performance of the protocols in terms of the mean delay of frames successfully delivered to the sink in Figure 6.9. The latency difference between ESRT and FCE based protocols is nearly 1 second regardless of the congestion level. When the load is light, SUIT has a similar performance with FCE. However, as the load increases, the performance gap between SUIT and FCE widens up to 28%. The main reason for the performance gap between SUIT and the other protocols is the average size of the frames. If the network congestion increases, SUIT decreases the frame quality, so the size of the frames decreases. Since small sized frames are transmitted faster than the origial frames, SUIT provides the best frame latency performance.

The performances of the protocols are compared in terms of average energy consumption per sensor node in Figure 6.10. The average energy expenditure is related to the number of generated frames at source nodes. As shown in Figure 6.12, ESRT always generates more frames than SUIT and FCE so it consumes more energy than the

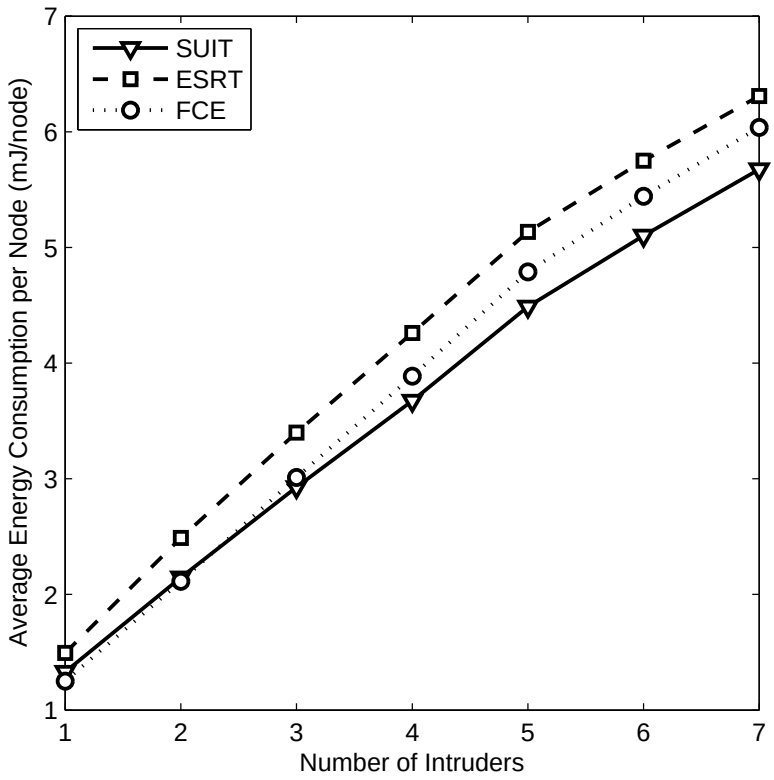


Figure 6.10. Energy expenditure per sensor node.

other protocols. SUIT produces more frames than FCE and therefore it consumes more energy than FCE if the load is light (if the number of intruders is lower than three). However, when the network is getting congested, energy expenditure of SUIT provides better results and the gap between the SUIT and other protocols increases. Since the SUIT protocol can shrink the size of frames in congested networks, it transmits lower number of packets and exhibits better performance in terms of energy consumption in highly loaded networks.

For the image quality performance evaluation of the protocols, the average PSNR values of the received images are used and the result is shown in Figure 6.11. PSNR calculation is performed by using Equation 6.5. Average quality of the received frame for ESRT and FCE are too close to each other. As expected, SUIT provides lower quality than other protocols. The main goal of SUIT is delivering more frames with lower delay and lower energy consumption by decreasing the image quality to an ac-

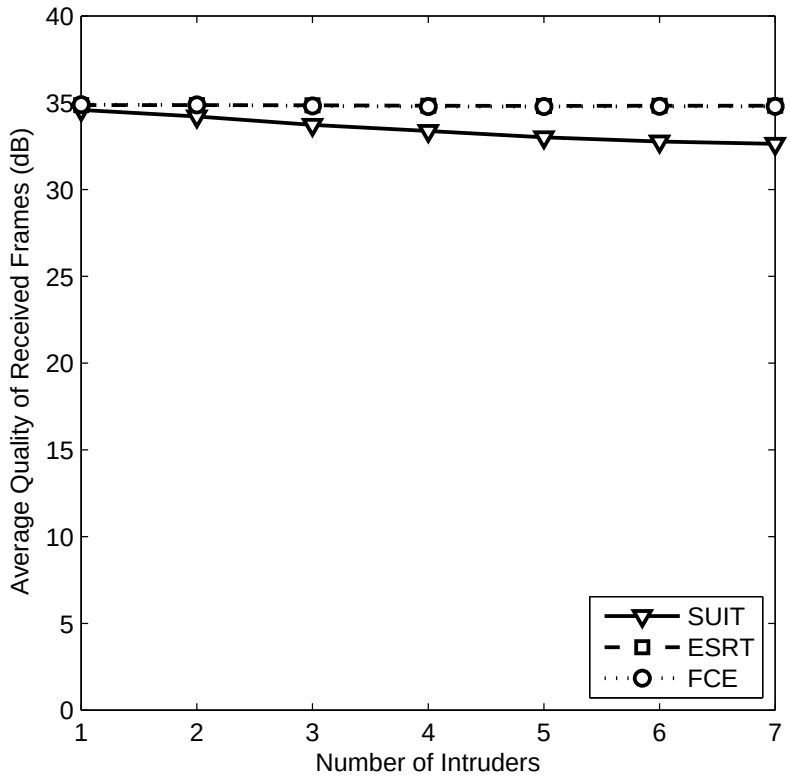


Figure 6.11. Average quality of received frames.

ceptable level. Therefore, it is not surprising that the average PSNR value of the images transmitted with the SUIT protocol is the lowest one. If the network is getting congested, the image quality performance of SUIT decreases. The average PSNR value for the worst case (when there are seven intruders in the area) is observed as 32.5 dB. In the simulations, PSNR value of images varies between 18 dB to 35 dB. According to Table 6.3, images whose PSNR value is higher than 20 dB can be used for object detection algorithms at the sink side. Therefore, images with 32.5 dB still have fairly good quality.

The frame rates at source sensor nodes are calculated in a different manner for each protocol. SUIT and FCE use fuzzy logic based approach whereas ESRT uses local buffer occupancy for deciding on the congestion level. According to the congestion level, the frame rate is assigned dynamically while generating images. In Figure 6.12, the average calculated frame rate at the source sensor nodes is illustrated. All of the

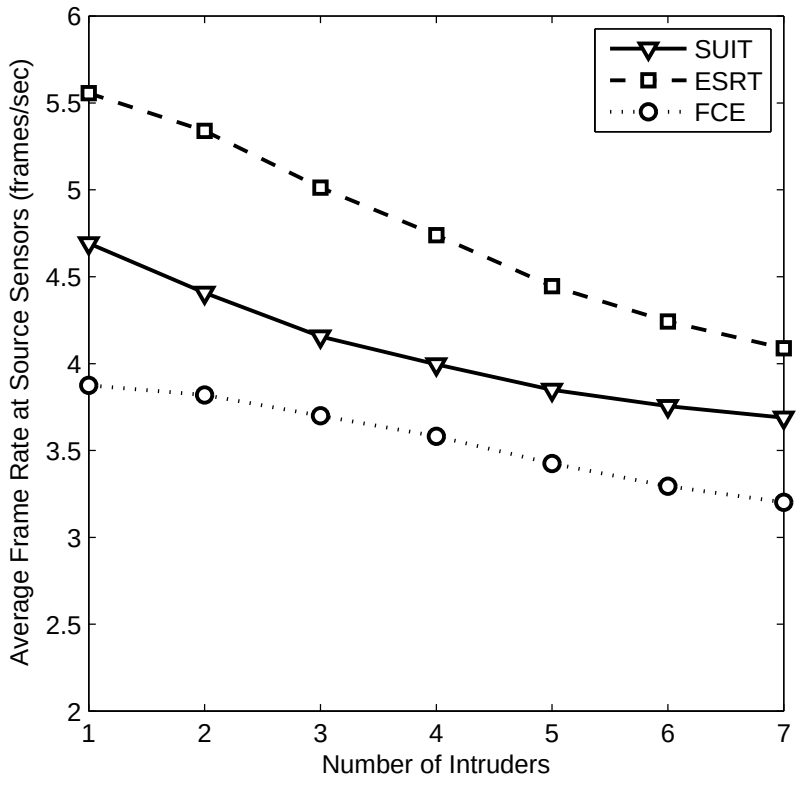


Figure 6.12. Average frame rate at source nodes.

protocols decrease the frame rate as the network is getting congested. ESRT estimates the congestion level optimistically and therefore calculates higher frame rates than the others. On the other hand, FCE estimates the congestion in a pessimistic manner and calculates lower frame rates than the others. SUIT provides a balanced congestion level via its efficient fuzzy logic based congestion estimation technique. Therefore, SUIT provides better performance than its competitors in all QoS metrics, except of the average received frame quality.

6.4. Comparison of Images Transmission with SUIT and Constant Rate

In this section, SUIT's rate adaptation technique is evaluated by comparing it with the image transmission at a constant frame rate. As explained in Section 5.2, SUIT dynamically changes the image sending rate based on the congestion level. In order to stream the images at a constant frame rate, we modify SUIT protocol by implementing a new image transmitting module which generates the frames at a predefined rate. In

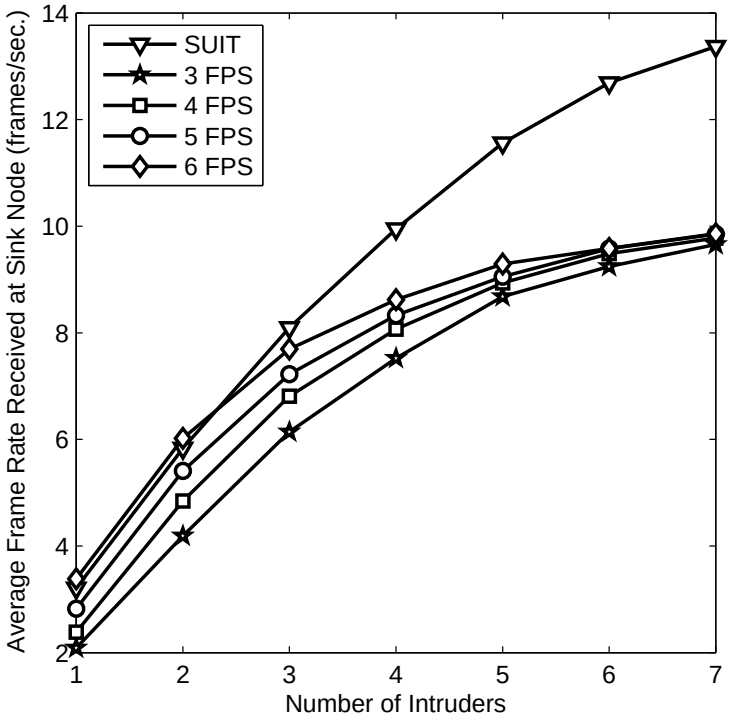


Figure 6.13. Average received frame rate at the sink.

this module, the congestion detection mechanism is not needed to be applied because a constant frame rate is used instead of an adaptive frame rate. Since the level of congestion is not calculated, the quality adaptation technique is not used either. As a competitor, four different frame rates are selected which are 3 fps, 4 fps, 5 fps and 6 fps. Basically, these competitors are using SUIT protocol without congestion detection and mitigation techniques.

In Figure 6.13, the performance of average frame rate received at the sink node is compared. If there are less than or equal to two intruders in the area, transmitting images at 6 fps provides better performance than SUIT. However when there are more than two intruders, transmitting images at 6 fps starts to lose too many frames. On the other hand, SUIT can decrease congestion by decreasing the quality of the images and the frame rate of the sensors. As shown in Figure 6.13, all competitors saturate at 10 fps whereas SUIT can reach 13 fps. As a result, SUIT can transmit more frames to the sink than the competitors in congested networks.

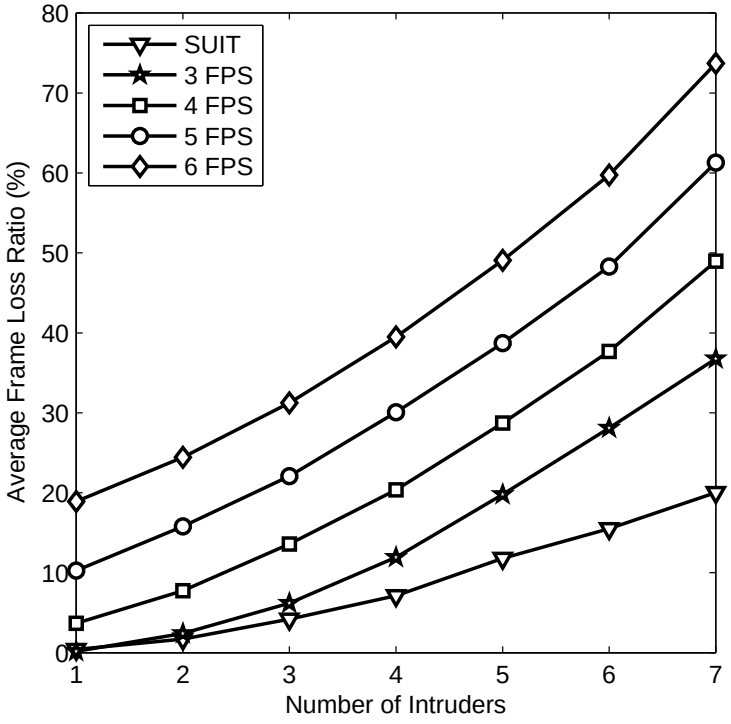


Figure 6.14. Average frame loss.

In Figure 6.14, the average frame loss performances are evaluated. For all of the cases, SUIT provides better frame loss performance than the competitors. Since only SUIT estimates the congestion level and applies the subsequent congestion mitigation techniques, it can prevent dropping frames by decreasing the congestion level as well as the buffer fullness. On the other hand, while streaming images at a constant rate, the competitors have to drop some frames when the buffer of the sensor node is full.

In Figure 6.15, the average frame latency performances are compared. Regardless of the congestion level, SUIT provides better frame latency performance than image transmission at 4, 5 and 6 fps. Sending images at 3 fps in lightly loaded networks provides better average frame latency than the SUIT protocol because 3 fps does not generate too much frames to cause congestion in the network. However if there are more than four intruders in the area, network becomes congested even if the frame rate is 3 fps. Since SUIT can decrease the size of the frames by decreasing the image quality in congested networks, it can transmit images faster than the competitors.

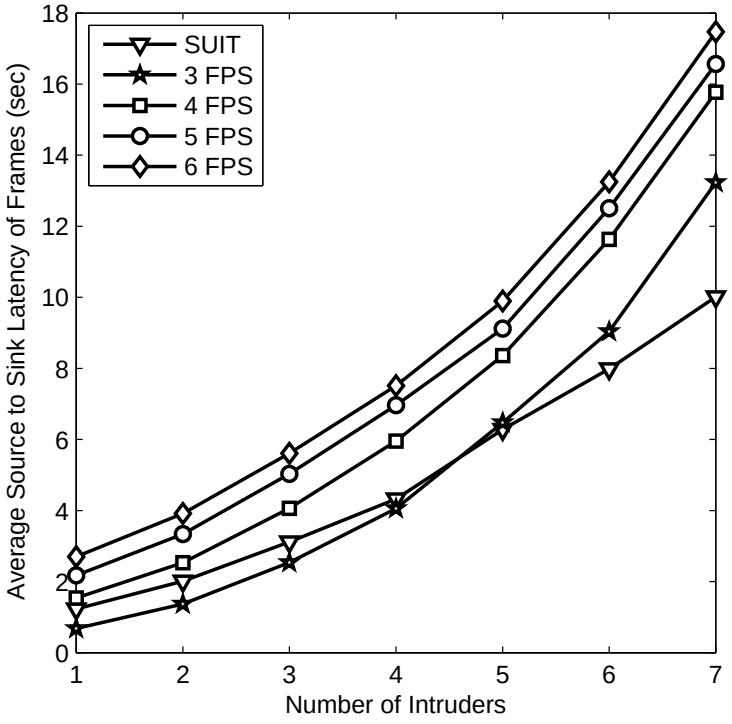


Figure 6.15. The average source to the sink latency of frames.

We compare the performances in terms of the average energy consumption per sensor node and the results of the comparison is shown in Figure 6.15. The energy consumption is calculated according to the number of sent and received packets. If the number of the transmitted images over the network increases, the number of packets on the network also increases and therefore the energy consumption of the sensor nodes becomes high. As a result, streaming images with lower frame rate reduces the energy expenditure. Since SUIT provides variable frame rate and image size by the help of its congestion mitigation techniques, energy expenditure of SUIT protocol may change based on the congestion level. As shown in Figure 6.15, transmitting images with 3 and 4 fps consumes lower energy than SUIT in lightly loaded networks. However, when there are three intruders in the area, SUIT provides better energy consumption performance than the image transmission at 4 fps. If there are more than 5 intruders, SUIT outperforms the image transmission at 3 fps and exhibits better performance than the competitors in terms of the average energy consumption in highly loaded networks.

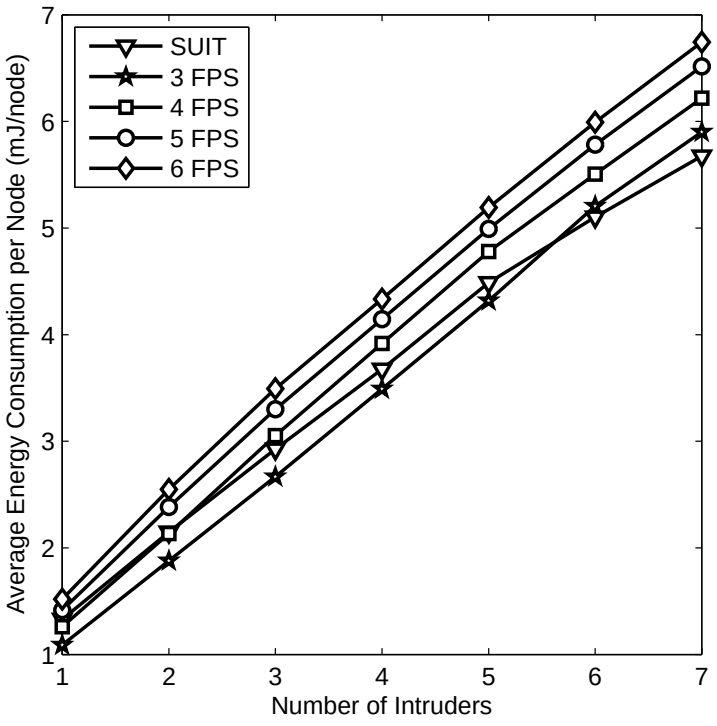


Figure 6.16. Energy expenditure per sensor node.

In this section, SUIT's rate adaptation technique is compared to the image transmission at a constant frame rate, which are 3, 4, 5 and 6 fps, with extensive number of simulations. In the light of these simulation results, for congested networks it can be concluded that the image transmission by using the rate adaptation technique of SUIT provides better performance than the image transmission at a constant frame rate with respect to the average frame rate received at the sink, the average frame loss, the average frame latency and the average energy consumption per sensor node.

7. CONCLUSIONS

In this thesis, we explored the transport protocols in the literature which provide reliable transmission and congestion control for WSNs. Then the issues of multimedia transmission are explained in terms of application, transport, routing and MAC layers. Motivated by existing image transport protocols, we proposed a new protocol, SUIT, based on the necessities of efficient multimedia transmission over WMSNs. SUIT is a cross-layer image transport protocol which provides a fuzzy-logic approach for estimating the congestion efficiently and then reacting to mitigate the congestion. To the best of our knowledge, mitigating congestion by decreasing the image quality on-the-fly is a new technique proposed by SUIT.

SUIT is developed for the applications which require transmitting real-time video to the base station. Therefore it is designed as a connectionless protocol which does not require establishing a connection before streaming real-time video from source to the sink immediately. Moreover, it provides out-of-order packet delivery, as well as it prioritizes packets according to their hop count, average delay per hop and frame index.

JPEG image format is commonly used by sensor nodes due to requiring low encoding complexity and providing enough compression for sensor nodes. SUIT supports JPEG transmission including SJPEG, SJPEG-RM and PJPEG encodings. Extensive simulation results show that SUIT gives better performance with PJPEG in terms of the average frame delay, the average frame loss and the total number of received frames. Therefore, PJPEG is the recommended encoding type for the SUIT protocol.

Congestion is the major problem in WMSN because it degrades the overall performance of the network QoS of the application. In WMSNs, the probability of congestion is higher than WSNs because multimedia traffic produces bursty high-load traffic in the network. Therefore, congestion control techniques used for WMSNs should be different than the ones proposed for traditional WSNs. SUIT has an efficient congestion

detection mechanism. Instead of using a specific congestion indicator, it proposes a fuzzy-logic based congestion detection technique which results in better performance. Congestion indicators of SUIT are; (i) the buffer occupancy of the next-hop sensor node, (ii) the number of contenders and (iii) the ratio of incoming to outgoing packets. Since these indicators are stored at different layers of the network stack, SUIT uses a cross-layer approach for sharing information among layers.

For congestion mitigation, SUIT provides two different techniques. The first technique is adapting the video frame rate at source sensor nodes. The second one is a novel congestion mitigation technique which can adapt the quality of images on-the-fly. The main goal of this technique is increasing the frame delivery performance by decreasing the quality while frames are being transmitted on the intermediate sensor nodes. Instead of dropping a whole frame, SUIT drops some packets of the frame and thus transmits it to the sink with lower, but acceptable quality. By this way SUIT improves the continuity of the video streaming.

A surveillance application scenario is implemented and extensive simulations are conducted in OPNET to verify our approach. The SUIT protocol is compared to ESRT based and FCE based congestion control protocols. Simulation results show that, SUIT works better than other protocols under high network traffic which is created by increasing the number of targets in the monitored area. Especially for the congested network cases, SUIT outperforms other protocols in all aspects except of quality. In the congested network case, SUIT provides the best average received frame rate performance at the sink. With the SUIT protocol, the sink sensor can receive up to 13 fps on average whereas with ESRT and FCE, the sink sensor can receive up to 10 fps. When it comes to the frame loss and latency performance, SUIT comes out on top regardless of the congestion level of the network. As a result, SUIT can transmit more images than other protocols by providing lower delay and loss. Another metric that SUIT is ahead of its competitors is the energy consumption. If there are more than two intruders in the area, SUIT consumes less battery than its competitors. As we mentioned before, the main goal of the SUIT protocol is providing better performance at aforementioned metrics by decreasing the quality to an acceptable level. So the the

quality of received image performance of SUIT is lower than the other protocols as expected. However, the gap between SUIT and the other protocols is not very high, the average image quality of the SUIT protocol is still satisfactory.

As the future work, we want to apply fuzzy-logic based packet prioritization while dropping packets from the local queue. In our current implementation, packets are dropped according to their weights which are calculated based on the predefined rules. However, fuzzy-logic based approach can provide more fairness on queue management. In addition, we want to improve the target detection process by implementing a communication module between neighboring sensor nodes. The sensor nodes whose camera is monitoring the same area, can communicate to each other in order to prevent transmitting the same image of the target for better application performance.

REFERENCES

1. Yigitel, M. A., O. D. Incel and C. Ersoy, "Design and Implementation of A QoS-Aware MAC Protocol for Wireless Multimedia Sensor Networks", *Computer Communications*, Vol. 34, No. 16, pp. 1991–2001, 2011.

2. Boluk, P. S., S. Baydere and A. E. Harmanci, "Robust Image Transmission Over Wireless Sensor Networks", *Mobile Networks and Applications*, Vol. 16, pp. 149–170, 2011.

3. Akyildiz, I. F., T. Melodia and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks.", *Computer Networks*, Vol. 51, No. 4, pp. 921–960, 2007.

4. Poojary, S. and M. Pai, "Multipath Data Transfer in Wireless Multimedia Sensor Network", *International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, pp. 379–383, 2010.

5. Akyildiz, I., T. Melodia and K. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds", *Proceedings of the IEEE*, Vol. 96, No. 10, pp. 1588–1605, 2008.

6. Grieco, L., G. Boggia, S. Sicari and P. Colombo, "Secure Wireless Multimedia Sensor Networks: A Survey", *Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, pp. 194–201, 2009.

7. Kulkarni, R., A. Fö andrster and G. Venayagamoorthy, "Computational Intelligence in Wireless Sensor Networks: A Survey", *IEEE Communications Surveys Tutorials*, Vol. 13, No. 1, pp. 68–96, 2011.

8. Maimour, M., C. Pham and D. B. Hoang, "A Congestion Control Framework for Handling Video Surveillance Traffics on WSN", *CSE (2)*, pp. 943–948, IEEE Computer Society, 2009.

9. Rijkse, K., "H.263: Video Coding for Low-Bit-Rate Communication", *IEEE Communications Magazine*, Vol. 34, No. 12, pp. 42–45, 1996.

10. Sikora, T., "MPEG Digital Video-Coding Standards", *IEEE Signal Processing Magazine*, Vol. 14, No. 5, pp. 82–100, 1997.

11. Slepian, D. and J. Wolf, "Noiseless Coding of Correlated Information Sources", *IEEE Transactions on Information Theory*, Vol. 19, No. 4, pp. 471–480, 1973.

12. Wyner, A. and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder", *IEEE Transactions on Information Theory*, Vol. 22, No. 1, pp. 1–10, 1976.

13. Yu, W., Z. Sahinoglu and A. Vetro, "Energy Efficient JPEG 2000 Image Transmission Over Wireless Sensor Networks", *IEEE Global Telecommunications Conference (GLOBECOM)*, Vol. 5, pp. 2738–2743, 2004.

14. Boukerche, A., Y. Du, J. Feng and R. Pazzi, "A Reliable Synchronous Transport Protocol for Wireless Image Sensor Networks", *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1083–1089, 2008.

15. Mammeri, A., A. Khoumsi, D. Ziou and B. Hadjou, "Energy-Efficient Transmission Scheme of JPEG Images Over Visual Sensor Networks", *33rd IEEE Conference on Local Computer Networks LCN*, pp. 639–647, 2008.

16. Agueh, M. and H. Soude, "Optimal Layer-Based Unequal Error Protection for Robust JPEG 2000 Images and Video Transmission over Wireless Channels", *First International Conference on Advances in Multimedia (MMEDIA)*, pp. 104–109, 2009.

17. Wallace, G., "The JPEG Still Picture Compression Standard", *IEEE Transactions on Consumer Electronics*, Vol. 38, pp. 18–34, 1992.

18. Misra, S., M. Reisslein and G. Xue, "A Survey of Multimedia Streaming in Wireless Sensor Networks", *IEEE Communications Surveys Tutorials*, Vol. 10, No. 4, pp. 18–39, 2008.

19. Wan, C.-Y., S. B. Eisenman and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", *Proceedings of the international conference on Embedded networked sensor systems (SenSys)*, pp. 266–279, 2003.

20. Wang, C., K. Sohraby and B. Li, "SenTCP: A Hop-by-Hop Congestion Control Protocol for Wireless Sensor Networks", *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2005.

21. Hull, B., K. Jamieson and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks", *Proceedings of the international conference on Embedded networked sensor systems (SenSys)*, pp. 134–147, 2004.

22. Wan, C.-Y., S. B. Eisenman, A. T. Campbell and J. Crowcroft, "Siphon: Overload Traffic Management Using Multi-Radio Virtual Sinks in Sensor Networks", *Proceedings of the international conference on Embedded networked sensor systems (SenSys)*, pp. 116–129, 2005.

23. Sankarasubramaniam, Y., O. Akan and I. Akyildiz, "ESRT: Event-to-SInk Reliable Transport in Wireless Sensor Networks", *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp. 177–188, 2003.

24. Iyer, Y., S. Gandham and S. Venkatesan, "STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks", *14th International Conference on Computer Communications and Networks (ICCCN)*, pp. 449–454, 2005.

25. Munir, S., Y. W. Bin, R. Biao and M. Man, "Fuzzy Logic Based Congestion Estimation for QoS in Wireless Sensor Network", *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 4336–4341, 2007.

26. Zarei, M., A. Rahmani, A. Sasan and M. Teshnehlab, "Fuzzy Based Trust Estimation for Congestion Control in Wireless Sensor Networks", *International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, pp. 233–236, 2009.

27. Ee, C. T. and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks", *Proceedings of the international conference on Embedded networked sensor systems (SenSys)*, pp. 148–161, 2004.

28. Paek, J. and R. Govindan, "RCRT: Rate-Controlled Reliable Transport Protocol for Wireless Sensor Networks", *ACM Transactions on Sensor Networks (TOSN)*, Vol. 7, No. 3, 2010.

29. Wang, C., K. Sohraby, V. B. Lawrence, B. Li and Y. Hu, "Priority-based Congestion Control in Wireless Sensor Networks", *SUTC (1)*, pp. 22–31, IEEE Computer Society, 2006.

30. Chaudhry, M., Z. Asad, A. Sprintson and J. Hu, "Efficient Rerouting Algorithms for Congestion Mitigation", *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 43–48, 2009.

31. He, T., J. Stankovic, C. Lu and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks", *23rd International Conference on Distributed Computing Systems*, pp. 46–55, 2003.

32. Marchi, B., A. Grilo and M. Nunes, "DTSN: Distributed Transport for Sensor Networks", *12th IEEE Symposium on Computers and Communications (ISCC)*, pp. 165–172, 2007.

33. Boukerche, A., Y. Du, J. Feng and R. Pazzi, "A Reliable Synchronous Transport Protocol for Wireless Image Sensor Networks", *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1083–1089, 2008.

34. Postel, J., *Transmission Control Protocol*, RFC 793, Internet Engineering Task Force, 1981.

35. Wan, C.-Y., A. Campbell and L. Krishnamurthy, "Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks", *Selected Areas in Communications, IEEE Journal on*, Vol. 23, No. 4, pp. 862–872, 2005.

36. Stann, F. and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks", *IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102–112, 2003.

37. Intanagonwiwat, C., R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed Diffusion for Wireless Sensor Networking", *IEEE/ACM Transactions on Networking*, Vol. 11, pp. 2–16, 2003.

38. E. Gurses, O. B. A., "Multimedia Communication in Wireless Sensor Networks", *Annals of Telecommunications*, Vol. 60, No. 7-8, pp. 799–827, 2005.

39. AlNuaimi, M., F. Sallabi and K. Shuaib, "A Survey of Wireless Multimedia Sensor Networks Challenges and Solutions", *International Conference on Innovations in Information Technology (IIT)*, pp. 191–196, 2011.

40. Pier Luigi Dragotti, M. G., *Distributed Source Coding: Theory, Algorithms and Applications*, Academic Press, 1 edition edn., 2009.

41. Girod, B., A. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed Video Coding", *Proceedings of the IEEE*, Vol. 93, No. 1, pp. 71 –83, 2005.

42. *ITU-T Recommendation H.264 : Advanced Video Coding for Generic Audiovisual Services*, http://www.itu.int/rec/T-REC-H.264-200711-I/en, accessed at May 2012.

43. Ahmad, J., H. Khan and S. Khayam, "Energy Efficient Video Compression for

Wireless Sensor Networks", *43rd Annual Conference on Information Sciences and Systems (CISS)*, pp. 629–634, 2009.

44. Charfi, Y., N. Wakamiya and M. Murata, "Network-Adaptive Image and Video Transmission in Camera-Based Wireless Sensor Networks", *First ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 336–343, 2007.

45. In, J., S. Shirani and F. Kossentini, "JPEG Compliant Efficient Progressive Image Coding", *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 5, pp. 2633–2636, 1998.

46. Zadeh, L. A., "Fuzzy Sets", *Information and Control*, Vol. 8, pp. 338–353, 1965.

47. Feng, G., "A Survey on Analysis and Design of Model-Based Fuzzy Control Systems", *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 5, pp. 676–697, 2006.

48. Mendel, J. M., "Fuzzy Logic Systems for Engineering: A Tutorial", *Proceedings of the IEEE*, Vol. 83, pp. 345–377, 1995.

49. Baghyalakshmi, D., J. Ebenezer and S. Satyamurty, "Low Latency and Energy Efficient Routing Protocols for Wireless Sensor Networks", *International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, pp. 1–6, 2010.

50. Karp, B. and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, 2000.

51. Akkaya, K. and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", *Ad Hoc Networks*, Vol. 3, pp. 325–349, 2005.

52. Heinzelman, W. R., J. Kulik and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proceedings of the 5th annual*

ACM/IEEE international conference on Mobile computing and networking (MobiCom), pp. 174–185, 1999.

53. Ciubotaru, B., D. Pescaru and D. Todinca, "Performances Analysis on Video Transmission in a Wireless Sensor Network", *4th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 183–186, 2007.

54. Sinan Isik, M. Y. D. and C. Ersoy, "Cross Layer Load Balanced Forwarding Schemes for Video Sensor Networks", *Ad Hoc Networks*, Vol. 9, No. 3, pp. 265–284, 2011.

55. Demirkol, I., C. Ersoy and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: A Survey", *IEEE Communications Magazine*, Vol. 44, No. 4, pp. 115–121, 2006.

56. Song, W.-M., Y.-M. Liu and S.-E. Zhang, "Research on SMAC Protocol for WSN", *4th International Conference on Wireless Communications, Networking and Mobile Computing(WiCOM)*, pp. 1–4, 2008.

57. Hameed, S., E. Shaaban, H. Faheem and M. Ghoniemy, "Mobility-Aware MAC Protocol for Delay-Sensitive Wireless Sensor Networks", *International Conference on Ultra Modern Telecommunications Workshops (ICUMT)*, pp. 1–8, 2009.

58. Sun, P., X. Zhang, Z. Dong and Y. Zhang, "A Novel Energy Efficient Wireless Sensor MAC Protocol", *Fourth International Conference on Networked Computing and Advanced Information Management (NCM)*, Vol. 1, pp. 68–72, 2008.

59. Nawab, F., K. Jamshaid, B. Shihada and P.-H. Ho, "TMAC: Timestamp-Ordered MAC for CSMA/CA Wireless Mesh Networks", *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, 2011.

60. Day, J. and H. Zimmermann, "The OSI Reference Model", *Proceedings of the IEEE*, Vol. 71, No. 12, pp. 1334–1340, 1983.

61. Kawadia, V. and P. Kumar, "A Cautionary Perspective on Cross Layer Design", *IEEE Wireless Communications*, Vol. 12, No. 1, pp. 3–11, 2005.

62. van Der Schaar, M. and S. S. N, "Cross-Layer Wireless Multimedia Transmission: Challenges, Principles, and New Paradigms", *IEEE Wireless Communications*, Vol. 12, No. 4, pp. 50–58, 2005.

63. Zhang, Y. and L. Cheng, "Cross-Layer Optimization for Sensor Networks", *Proceedings of the New York Metro Area Networking Workshop*, 2003.

64. Su, W. and T. Lim, "Cross-Layer Design and Optimization for Wireless Sensor Networks", *Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD)*, pp. 278–284, 2006.

65. Chiasserini, C.-F. and E. Magli, "Energy Consumption and Image Quality in Wireless Video-Surveillance Networks", *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 5, pp. 2357–2361, 2002.

66. Wallace, G., "The JPEG Still Picture Compression Standard", *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 1, pp. 18–34, 1992.

67. Lam, W.-M. and A. Reibman, "Self-Synchronizing Variable-Length Codes for Image Transmission", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 3, pp. 477–480, 1992.

68. Han, Y.-H. and J.-J. Leou, "Detection and Correction of Transmission Errors in JPEG Images", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 2, pp. 221–231, 1998.

69. Yoo, Y. and A. Ortega, "Constrained Bit Allocation for Error Resilient JPEG Coding", *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems Computers*, Vol. 2, pp. 985–989, 1997.

70. Mo Kim Cheng, A. and F. Shang, "Priority-Driven Coding of Progressive JPEG Images for Transmission in Real-Time Applications", *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 129–134, 2005.

71. Boukerche, A., Y. Huang, J. Feng and R. Pazzi, "Progressive Image Mosaicking in Wireless Image Sensor Networks", *Proceedings of the 6th ACM international symposium on Mobility management and wireless access (MobiWac)*, pp. 103–110, 2008.

72. Iren, S., P. D. Amer and P. T. Conrad, "Network-Conscious Compressed Images over Wireless Networks", *In 5th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS)*, 1998.

73. Postel, J., *User Datagram Protocol*, RFC 768, Internet Engineering Task Force, 1980.

74. Chen, C. J., R. G. Chang and C. W. Hsueh, "Wireless Smooth Data Streaming on Application Layer", *9th International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 384 –389, 2011.

75. Durmus, Y., A. Ozgovde and C. Ersoy, "Distributed and Online Fair Resource Management in Video Surveillance Sensor Networks", *IEEE Transactions on Mobile Computing*, Vol. 11, pp. 835–848, 2012.

76. Wang, C., K. Sohraby, B. Li, M. Daneshmand and Y. Hu, "A Survey of Transport Protocols for Wireless Sensor Networks.", *IEEE Network*, Vol. 20, No. 3, pp. 34–40, 2006.

77. Munir, S., Y. W. Bin, R. Biao and M. Man, "Fuzzy Logic based Congestion Estimation for QoS in Wireless Sensor Network", *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 4336–4341, 2007.

78. Jain, L. C. and N. M. Martin, *Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications*, CRC Press, Inc., Boca Raton, FL, USA, 1998.

79. Mokarram, M., A. Khoei, K. Hadidi and K. Gheysari, "Implementation of Centroid Defuzzifier Block Using CMOS Circuits", *4th International IEEE Conference on Intelligent Systems*, Vol. 1, pp. 2–29, 2008.

80. *OPNET Modeler*, http://www.opnet.com/products/modeler/home.html, accessed at May 2012.

81. Bettstetter, C., G. Resta and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 3, pp. 257–269, 2003.

82. *GNU Image Manipulation Program (GIMP)*, http://www.gimp.org/, accessed at May 2012.

83. Carlos F. Garcia-Hernandez, J. G.-h., Pablo H. Ibarguengoytia-Gonzalez and J. A.Perez-Diaz, "Wireless Sensor Networks and Applications : A Survey", *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 7, 2007.

84. *Atmel Corporation*, http://www.atmel.com, accessed at May 2012.

85. *Texas Instruments Incorporated*, http://www.ti.com/corp/docs/chipcon, accessed at May 2012.

86. *Microchip Technology*, http://www.microchip.com, accessed at May 2012.

87. Abdul Abdul Hamid, A. H. F., R. A. Rashid, N. Fisal and S. K. Syed Yusof, "Development of IEEE802.15.4 Based Wireless Sensor Network Platform for Image Transmission", *International Journal of Engineering and Technology (IJET)*, Vol. 9, pp. 112–118, 2009.

88. Park, C. and P. H. Chou, "eCAM: Ultra Compact, High Data-Rate Wireless Sensor Node With a Miniature Camera", *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys)*, pp. 359–360, 2006.

89. Atmel Corporation, *8-bit AVR Microcontroller with Low Power2.4GHz Transceiver for ZigBee and IEEE 802.15.4*, rev. 8266cs-mcu wireless-08/11 edn., 2011.

90. Zuniga, M. and B. Krishnamachari, "Analyzing the Transitional Region in Low Power Wireless Links", *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (SECON)*, pp. 517–526, 2004.

91. Sohrabi, K., B. Manriquez and G. Pottie, "Near Ground Wideband Channel Measurement in 800-1000 MHz", *IEEE 49th Vehicular Technology Conference*, Vol. 1, pp. 571–574, 1999.

92. Rappapport, T., *Wireless Communications: Principles and Practice*, Prentice Hall, 2001.

93. *Crossbow Technology*, http://www.xbow.com, accessed at May 2012.

94. Hore and, A. and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM", *20th International Conference on Pattern Recognition (ICPR)*, pp. 2366–2369, 2010.