RELIABLE REAL-TIME VIDEO COMMUNICATION IN WIRELESS SENSOR NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ORHAN AYRAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2007

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Özgür Barış Akan
Supervisor

Examining Committee Members

Prof. Dr. Semih Bilgen	(METU, EEE)	
Assoc. Prof. Dr. Özgür Barış Akan	(METU, EEE)	
Prof. Dr. Buyurman Baykal	(METU, EEE)	
Assoc. Prof. Dr. Gözde Bozdağı Akar	(METU, EEE)	
Dr. Altan Koçyiğit	(METU, IS)	

I thereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

RELIABLE REAL-TIME VIDEO COMMUNICATION IN WIRELESS SENSOR NETWORKS

Ayran, Orhan

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Özgür Barış Akan

February 2007, 58 pages

Many wireless sensor network (WSN) applications require efficient multimedia communication capabilities. However, the existing communication protocols in the literature mainly aim to achieve energy efficiency and reliability objectives and do not address the multimedia communication challenges in WSN. In this thesis, comprehensive performance evaluation of the existing transport protocols is performed and it has been shown that the existing proposals achieve very poor performance in terms of large set of metrics such as packet delivery rate, end-to-end packet delay, bandwidth and energy efficiency, frame peak signal-to-noise ratio (PSNR), delay-bounded frame PSNR, frame delivery probability, frame end-to-end delay and jitter. Based on these results, an energy-efficient real-time and reliable video sensor communication protocol (VSCP) is introduced for WSN. VSCP estimates video quality perceived by sink using lost segments of video frames and aims to maintain the overall reliability at a given level with minimum energy expenditure. Source data rates are adjusted in a quality adaptable manner according to the network conditions and the overall reliability computed by sink. QSC (quality scalable coding) encoding technique is used to produce a nearly constant quality video at a given maximum data rate during adjustment of source data rates. Performance evaluations show that VSCP protocol significantly outperforms the existing proposals in terms of multimedia communication performance metrics in WSN.

Keywords: Sensor Networks, Transport Protocol, Video Transport, Reliability, Congestion Detection

ÖZ

TELSİZ SENSÖR AĞLARINDA GERÇEK ZAMANLI VE GÜVENİLİR VİDEO İLETİMİ

Ayran, Orhan

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Özgür Barış Akan

Şubat 2007, 58 sayfa

Birçok TSA (Telsiz Sensör Ağı) uygulaması, verimli çoğulortam iletişimine ihtiyaç duymaktadır. Ancak, mevcut iletişim protokolleri temelde enerji verimliliği ve güvenilirlik konularını ele alırken TSA'daki çoğulortam iletişim zorluklarına değinmemektedir. Bu tezde, TSA için tasarlanmış olan taşıma katmanı protokollerinin çoğulortam iletişimindeki başarımları incelenmiş ve mevcut protokollerin, paket aktarım hızı, uçtan uca paket gecikmesi, bant ve enerji kullanım verimliliği, çerçeve sinyal gürültü oranı tepe değeri (PSNR), zaman sınırlı çerçeve PSNR'si, başarılı çerçeve aktarım olasılığı, uçtan uca çerçeve gecikmesi ve seğirmesi gibi başarım metrikleri açısından çok düşük başarım sergiledikleri gözlemlenmiştir. Bu sonuçtan yola çıkarak TSA'da gerçek zamanlı, enerji etkin ve güvenilir video iletimini hedefleyen bir iletişim protokolü (VSCP) tasarlanmıştır. VSCP protokolü, kaybolan çerçeve parçalarını kullanarak, alıcı tarafında elde edilen video kalitesini yaklaşık olarak hesaplamakta, uygulumanın ihtiyacı olan güvenilirliği enerji ve ağ etkin bir biçimde istenilen seviyelerde tutmaya çalışmaktadır. Yaklaşık olarak sabit bir video kalitesi elde etmek ve gönderici veri hızlarını ayarlamak için değişken kaliteli kodlama tekniği kullanılmıştır. Yapılan başarım testleri, çoğulortam başarım metrikleri açısından, VSCP'nin mevcut protokollerden çok daha iyi sonuç verdiğini göstermektedir.

Anahtar Kelimeler: Sensör Ağları, Taşıma Katmanı Protokolü, Video İletimi, Güvenilirlik, Tıkanıklık Sezimlemesi

To my parents,
Nejla and Seyfettin

ACKNOWLEDGMENTS

I am indebted to my advisor Assoc. Prof. Dr. Özgür B. Akan for his great advice, support, help and showing me the right direction when I lost my way at certain times during this study.

I thank to my parents for everything they do for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

CODA	Congestion Detection and Avoidance
EERA	Energy Efficient Rate Adaptation
ESRT	Event-to-Sink Reliable Transport
GOP	Group Of Pictures
MPEG	Moving Pictures Expert Group
PSFQ	Pump Slowly Fetch Quickly
PSNR	Peak Signal To Noise Ratio
QoS	Quality of Service
QP	Quantization Parameter
QSC	Quality Scalable Coding
RFP	Received Frame PSNR
RMST	Reliable Multi-Segment Transport
RT	Real-time
SenTCP	Sensor Transport Control Protocol
TFP	Target Frame PSNR
VSCP	Video Sensor Communication Protocol
WFP	Wait for First Packet
WMSN	Wireless Multimedia Sensor Network
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

Wireless Sensor Networks (WSN) are, in general, multi-hop ad hoc networks composed of small sensor nodes with limited capabilities in terms of power, processing, memory and communication ranges. Recently, considerable amount of research efforts have yielded many promising communication protocols to address the challenges posed by the WSN paradigm. However, these research results mainly address the reliable and energy-efficient communication problems of the WSN applications which primarily require conventional data communications. Nevertheless, there exist many proposed WSN applications such as target tracking, source localization, discovering and following rare animal species, controlling the vehicle traffic in highways and railways which may also involve in collecting information in the form of multimedia such as audio, image, and video; and hence necessitate efficient multimedia communication in WSN [3].

Wireless Multimedia Sensor Network (WMSN) is a sensor network composed of small sensor nodes with multimedia sensing capability such as video and audio. Traditional Wireless Sensor Network (WSN) applications are based on simple scalar measurements of sensor nodes and require low bandwidth due to limitations of sensor nodes in terms of power, bandwidth, processing, memory capabilities. However, recent advances in multimedia hardware such as small and inexpensive CMOS cameras, low-power, high-performance processors, and improvements in embedded device technology promise a wide range of new WSN applications involving area surveillance, environmental monitoring, tracking etc. [11].

The need for multimedia delivery in WSN brings extra communication challenges which are not sufficiently studied yet. Multimedia applications require strict quality of service (QoS) guarantees of high bandwidth, short inter-packet and end-to-end delays. In addition to limited capabilities of sensor nodes, due to event-based and multi-sender nature of WSN, high bandwidth demand itself brings a set of challenges. Huge amount of traffic generated by multiple sources within an event area increases probability of network congestion. If sources are deployed close to each other in the field, there will be a contention bottleneck around the event area causing source node queues to build-up resulting a congestion event. Congestion not only causes packet losses due to queue overflows, but it also increases delay of packets that are successfully forwarded towards sink. This causes late losses which waste sensor nodes energy since packets having long delays are dropped by application. Another challenge is due to temporally and spatially correlated nature of multimedia data. For example, a wide range of video encoders perform predictive coding which exploits spatial correlation of inter-frames. In such a case, quality of a received frame depends on successful reception of the previous one. Hence, burst of packet losses is another problem which decreases application reliability at the same time wasting sensor node resources.

In addition to challenges of multimedia communication itself, there are other difficulties posed by WSN in multimedia communication. In WSN, information needed by sink is formed by collective effort of a number sensor nodes within an event area. Hence, communication protocols should provide an event-to-sink reliable transport mechanism instead of an end-to-end reliability [2]. There are two main challenges of event-to-sink reliability for multimedia delivery in a WSN;

- In-network processing of data gathered from multiple sensors in order to extract relevant information needed by sink. First, multimedia data such as video is two dimensional and different images captured by sensors at different time instants makes the aggregation process very challenging. Although many research studies have been done such as image registration [13], fusion of images with non-linear, locally dependent geometric distortions is an open issue and needs further research effort. Second, complex in-network processing of multimedia data decreases battery-limited sensor lifetimes due to huge amount of data exchanges between sensor nodes.

- Successful processing of spatially and temporally correlated received data at sink in order to provide reliable results to the end-user. It is inevitable to have powerful algorithms at sink to pre-process multimedia data gathered from multiple sensors to give relevant information required for application. However, challenges discussed for in-network processing are also valid for sink-based processing when exploiting spatial and temporal correlation of multimedia data.

In terms of communication protocol stack, event-to-sink reliability can best be provided by exchanging data between various communication layers in a WSN, thus using a cross-layer approach. Indeed, reliable data transport is the main objective of transport layer. However, reliable collective multimedia delivery depends on various parameters such as different delays, different quality levels of received data from multiple sensors, energy consumption and packet loss rates and the only way to obtain an overall reliability measurement is combining information gathered from different layers of protocol stack such as MAC layer for queuing delays, application layer for quality of decoded multimedia data, network layer to obtain statistics of different data flows and so on.

Clearly, reliable multimedia delivery techniques are imperative to realize efficient multimedia communication in WSN. With this regard, the main focus of this thesis is to comprehensively evaluate the performance of the existing WSN transport layer protocols in multimedia communication scenarios and to design a communication protocol for WSN to achieve reliable and efficient real-time video delivery in terms of application-specific reliability requirements such as aggregated received video quality, delay and energy consumption. To this end, first a wide range of simulations are performed and results are presented to clearly assess the performance and point out the shortcomings of the currently proposed transport protocols in multimedia communication, and then a protocol, VSCP (Video Sensor Communication Protocol) is proposed that overcomes these shortcomings in an efficient way.

The rest of this thesis is organized as follows. In Chapter 2, a brief overview of the existing transport protocols and results of performance evaluation experiments are presented. Design and performance evaluation of the VSCP protocol are given in Chapter 3. Concluding remarks are given in Chapter 4.

CHAPTER 2

PERFORMANCE EVALUATION OF EXISTING TRANSPORT PROTOCOLS

2.1 Overview of Existing Protocols and Multimedia Challenges

Currently, there exist several transport protocols [2, 6, 7, 8, 9, 10] proposed to achieve congestion control and/or energy-efficient reliable data delivery for conventional data communication in WSN. Main properties of these protocols are summarized in Table 2.1. Although these protocols provide congestion detection or reliable transport functionalities, none of them supports real-time (RT) communication or provides quality-of-service (QoS) guarantees in terms of delay and minimum bandwidth requirements. RMST [8] performs strict (i.e., 100%) packet-based reliability which only leads to waste of sensor resources due to loss-tolerant nature of multimedia. ESRT [2] provides reliable event detection, however it neither considers nor addresses the requirements of multimedia delivery in WSN. On the other hand, GARUDA [10] and PSFQ [9] provide energy-efficient reliable delivery of sensing commands and queries over the reverse path, i.e., from the sink to the sensor nodes, which is irrelevant to the problem of multimedia communication from the sensor field to the sink, thus, GARUDA and PSFQ are not included in the experiments.

2.1.1 ESRT (Event-to-Sink Reliable Transport)

ESRT [2] aims to achieve reliable event detection with minimum energy expenditure and congestion resolution. It provides both congestion and reliability con-

Table 2.1: Summary of Existing Transport Protocols for WSN

Name	*Direction*	*Congestion Detection*	*Reliability Support*	*QoS or RT Support*
ESRT [2]	Forward	Passive	Application	-
CODA [6]	Forward	Active	-	-
SenTCP [7]	Forward	Active	-	-
RMST [8]	Forward	-	Packet	-
PSFQ [9]	Reverse	-	Packet	-
GARUDA [10]	Reverse	-	Packet	-

trol schemes. ESRT is a sink-based protocol where the sink computes event reliability (i.e., number of data packets received divided by desired number of data packets expected to be received) periodically at each decision interval. Intermediate sensor nodes monitor their buffer level for congestion detection. When sensor nodes realize that their buffer will overflow in the next decision interval, they notify the sink by setting congestion bit of data packets forwarded towards sink. The sink knowing the current reliability and congestion existence in the whole network, updates the reporting (sending) rates of all source nodes in order to maximize the reliability with minimum energy usage. Under no congestion and low reliability, sending rate of source nodes are increased multiplicatively, while when congestion exist and reliability is low, an exponential decrease policy is used. ESRT provides reliable event detection, however it neither considers nor addresses the requirements of multimedia delivery in WSN.

2.1.2 CODA (Congestion Detection and Avoidance)

CODA [6] aims to provide an energy efficient congestion control scheme for sensor networks. It has three key mechanisms:

- Congestion detection.
- Open-loop, hop-by-hop back-pressure.
- Closed-loop, multi-source regulation.

Intermediate sensor nodes perform buffer level monitoring and channel status sensing in order to detect congestion. Channel status is sampled periodically during back-off to save energy. When the sensed channel load (i.e., number of times channel is busy) exceeds a threshold, a node broadcasts a suppression message as a back-pressure signal and at the same time exercises the local congestion policy. When source nodes receive the suppression messages, they decrease their sending rates. In closed-loop multi-source regulation, if the event rate of a source exceeds some fraction of the maximum theoretical throughput, it triggers sink regulation by setting a regulate bit in the event packets it forwards to the sink which causes sink to send ACKs at certain times. If a source node receives a prescribed number of ACKs during an interval, it maintains its rate, otherwise the source node decreases its sending rate.

CODA is specifically designed for congestion detection/avoidance and does not define how source data rates are decreased/increased for reliable event detection. Since real-time multimedia data packets, especially video packets are highly correlated, packet based multiplicative decrease or additive increase methods decrease final reliability perceived by sink. Secondly, both hop-by-hop back-pressure signalling and closed-loop multi-source regulation require the reverse channel to be available when congestion occurs, which may not be the case when congestion is due to huge amount of multimedia data traffic.

2.1.3 SenTCP (Sensor Transport Control Protocol)

SenTCP [7] is a hop-by-hop congestion control protocol. Each intermediate sensor node monitors packet arrival and service times to detect congestion. It uses hop-by-hop congestion control such that each intermediate sensor node issues feedback signal backward and hop-by-hop. The feedback message, which carries local congestion degree and the buffer occupancy ratio, is used for the neighboring sensor nodes to adjust their sending rate in the transport layer.

SenTCP's congestion control mechanism is reactive, each intermediate sensor node issues a feedback signal backward and hop-by-hop whether or not congestion exists. When there is no congestion, feedback signal is sent periodically while when there is congestion it is sent at each packet arrival which has a side effect of decreasing performance of multimedia communication where there is high data traffic. In addition to contention caused by high data rates of neighbor nodes, periodically sent congestion detection packets increase the network load. SenTCP has no reliability or QoS support which are needed for multimedia communication.

2.1.4 RMST (Reliable Multi-Segment Transport)

RMST [8] aims to provide reliable data delivery along the forward path (i.e., from sources to sink). It is designed to run above Directed Diffusion (to use its discovered path from sensors to sink) in order to provide guaranteed reliability from sensors to sink (delivery and fragmentation/reassembly) for applications. RMST is a selective NACK-based protocol. RMST basically operates as follows. Firstly, RMST uses timer-driver mechanism to detect data loss and sends NACK on the way from detecting node to sources (Cache or non-Cache mode). Secondly, NACK receivers are responsible for looking for the missing packet, or forward NACK on the path towards sink if it fails to find the missing packet or in non-cache mode.

RMST is designed in order to provide 100 % guaranteed data delivery which is not needed for multimedia delivery especially for real-time applications. It has no congestion control, no energy conservation mechanism and application-level reliability.

2.1.5 PSFQ (Pump Slowly Fetch Quickly)

PSFQ [9] aims to distribute data from sink to sensors by pacing data at a relatively slow-speed, but allowing nodes that experience data loss to fetch (recover) any missing segments from immediate neighbors very aggressively (local recovery, ”fetch quickly”). PSFQ provides reliability in the reverse path (i.e., sink-to-sensors) and mainly aims to deliver the queries requested by the sink to the sensor field so it is not suitable for multimedia communication that originates from sources towards sink.

2.1.6 GARUDA

GARUDA [10] aims to provide reliability in the reverse path (i.e., from sink to sources). It has three primary components. Firstly, GARUDA uses WFP (Wait-for First -Packet) pulse transmission to guarantee success of single/first packet delivery, in order to choose and construct Core sensors. Secondly, GARUDA performs core election such that only sensors with hop-count of the form $3 * i$ are allowed to elect themselves as core sensors. GARUDA performs two phase loss recovery (loss recovery for core and non-core sensors) using out-of-sequence NACK. Like PSFQ, GARUDA provides reliability in the reverse path and hence not suitable for multimedia communication in WSN.

2.2 Performance Evaluation

2.2.1 Simulation Environment

Simulation experiments are performed using ns-2 [19]. 50 nodes are randomly placed in 200m X 200m field. 4 source nodes are randomly selected within an event area of radius 70 m. Sink is randomly located within the field. Each node has a queue size of 50, radio range of 40m, 2Mbps 802.11 MAC where link layer ARQ and RTS/CTS exchange mechanisms are disabled. In [14] it is shown that using shorter size packets, a better goodput can be obtained in a sensor network. However, 802.11 RTS/CTS exchange decreases throughput when packet size is small, so basic access mechanism provides better performance. Directed Diffusion [5] is used as the routing protocol. Simulations are performed for 25 times using random deployment and the results are averaged. ESRT [2], CODA [6], SenTCP [7], and RMST [8] (i.e., RMST is only included in video streaming simulations to observe the effect of 100% reliability objective on multimedia delivery performance) are included in the experiments.

2.2.2 Data Traffic With High Load

Traffic originating by a multimedia event occurring in a densely deployed sensor field yields high network load. Number of nodes positioned in the event area that are involved in the communication determines the level of the offered load. Here, each source node is assumed to start transmission with the initial sending rate of 200 packets/sec and a packet size of 200 Bytes. Note that this corresponds to a multimedia delivery scenario where the generated traffic rate is 320 Kbits/sec per source. Total data traffic originating from 4 sources results 1.28 Mbps (800 packets/sec).

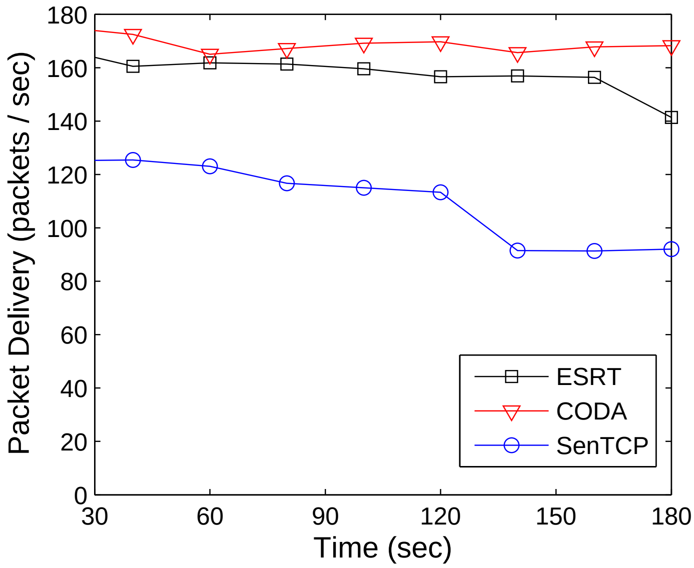


Figure 2.1: Packet delivery rate.

As shown in figures 2.1 and 2.2 number of dropped packets are always greater than number of delivered packets for each protocol. SenTCP does not wait for congestion to happen in order to notify the sources, since it does this periodically even if no congestion occurs in network. There is always an increase in packet drop rate and the protocol can not cope with congestion. In SenTCP when buffer ratio of a sensor node exceeds some value, it tries to send feedback signal at each packet arrival, which does not work under such multimedia scenarios where network is congested because of high number of packets injected to the network. While sources are already trying to send high number of data packets to the sink, intermediate sensor nodes are injecting the same number of packets to the reverse channel which makes the situation worse. As a result, number of dropped packets increases with time, up to a value more than 375 packets/sec.

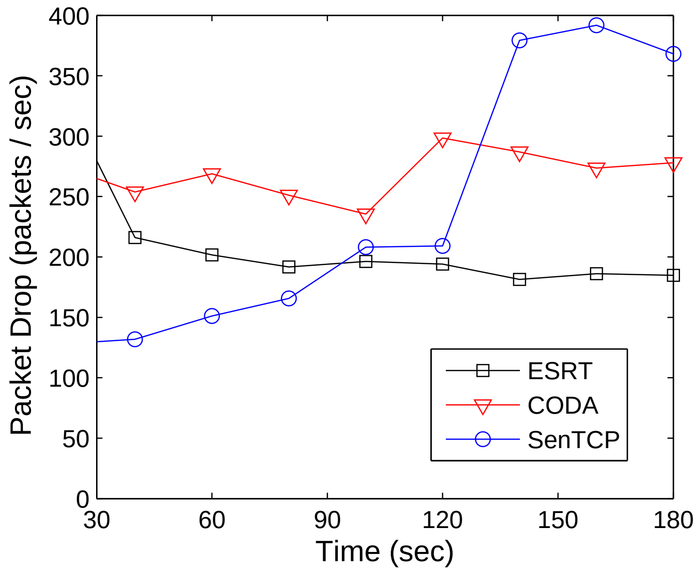


Figure 2.2: Packet drop rate.

CODA's open-loop hop-by-hop back-pressure mechanism needs the intermediate sensor nodes to detect congestion and notify the sources, but not only this mechanism does not work when sources are not so close to each other but also under congestion, reverse channel is not always available to forward these suppression messages towards sources. Number of dropped packets are in the order of 250 packets/sec up to 100 sec, while it increases up to 300 packets/sec after on. Neither open-loop nor closed-loop control mechanisms of CODA does not help with decreasing the packet drop rate of the network to an acceptable value.

In ESRT, packet drop rate is at about 200 packets/sec and does not change much until end of the simulation. First, ESRT does not need intermediate sensor nodes to send extra messages (i.e., it uses the headers of packets that are to be forwarded along sink) for congestion detection, not causing extra waste of energy at these nodes, secondly all control mechanisms are provided by sink.

As shown in data traffic simulation results, network is congested and works at the full capacity when the data traffic load is high as will be the case with multimedia delivery. It is observed that the number of dropped packets are significantly greater than the received packets for all protocols. This shows that sensor nodes waste most

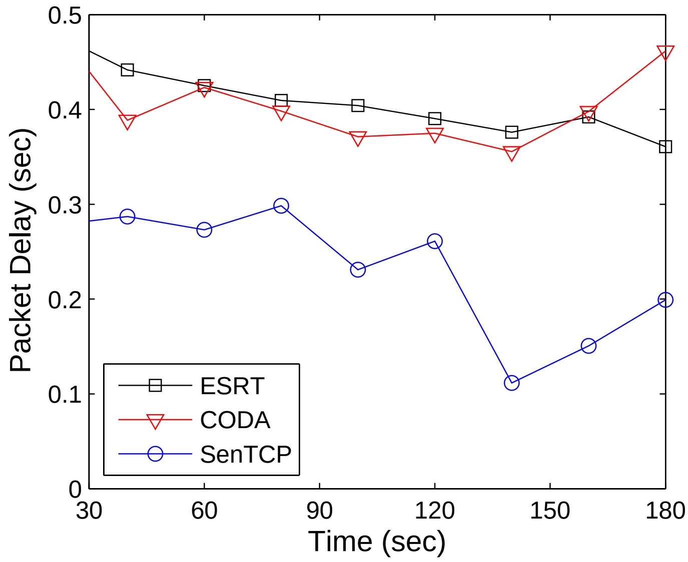


Figure 2.3: End-to-end packet delay.

of their power in the scenarios where the data delivery requires high bandwidth. As shown in Figure 2.3 end-to-end packet delay performances are not satisfactory especially for WSN applications that require effective interaction of sink and sources. Consequently, results obtained indicate that these protocols do not achieve efficient communication when the offered network load is high which will be the case in multimedia communication scenarios.

2.2.3 Real-time Video (MPEG-4) Streaming

A sample MPEG-4 [16] video sequence (i.e., Waterfall [23] encoded with *ffmpeg* [21] encoder) of 250 frames and 1850 Bytes average frame length is used for the simulations of real-time video streaming in WSN. Average frame PSNR is 38 dB, GOP size is 12 with IP...P pattern. The initial frame rate is 30 frames/sec and the frames are packetized into 100 Bytes. When the protocol reduces the sending rate, the source omits the frame which has a sending time less than that of the next frame to maintain the real-time delivery objectives.

Results are extracted using a video quality evaluation tool named Evalvid [20] providing the received packets at the sink as an input to the tool. The sample video is first streamed with UDP (i.e., MTU=100 Bytes) between a client-server over a real network (local area network) and sent packets are recorded to a file. This file is used as an input to ns-2 such that each source within the event area reads this file (i.e., packet id, length) and sends packets accordingly. Received packets are recorded to a file by sink. The received trace-file, sent trace-file and sample video are given to Evalvid as an input and reconstructed video, PSNR, end-to-end frame delay, cumulative jitter are get as output. Since each packet trace has a time-stamp, late losses are also handled by the tool applying a default static play-out buffer size strategy. Lost frames/packets are zero padded during reconstruction of the MPEG video, however, when decoding the video into raw image sequence for performance evaluation, lost sections are concealed from the previously reconstructed frames.

2.2.3.1 Energy and Bandwidth Efficiency

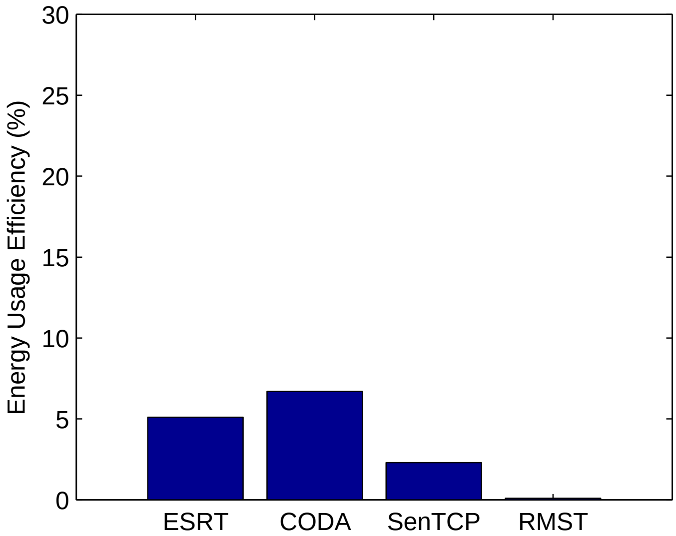


Figure 2.4: Energy usage efficiency.

Fraction of lost packets is an indication of waste of energy in WSN. Energy efficiency can be approximated as the fraction of number of packets received at sink to the total number of packets sent by source nodes as shown in Figure 2.4. The energy efficiency of protocols is very low due to huge amount of packet loss. CODA and ESRT provides better energy efficiency than SenTCP and RMST since their congestion detection mechanisms (i.e., although congestion is not resolved at all) detects congestion at certain times and source rates are decreased. Since less number of packets are injected to the network, this causes the percentage of dropped packets to decrease.

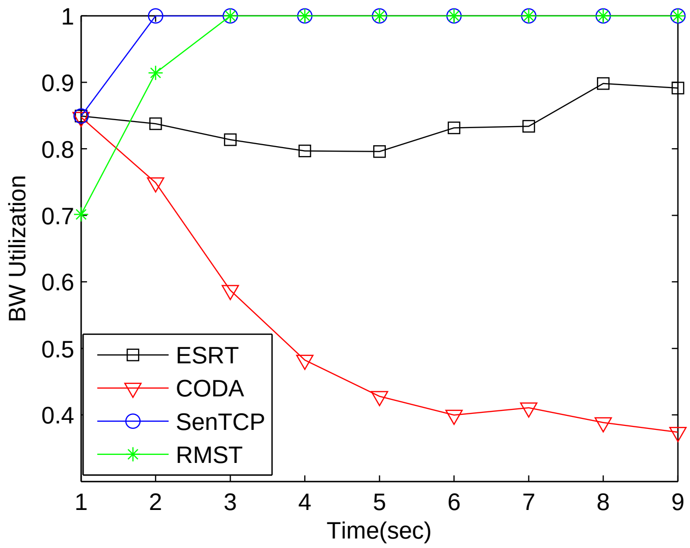


Figure 2.5: Normalized bandwidth utilization.

The normalized bandwidth utilization, i.e., the ratio of source sending rate to the required sending (initial) rate, is shown in Figure 2.5. SenTCP and RMST preserve the initial source sending rates while ESRT and CODA try to regulate the source sending rates to decrease the congestion level. Although SenTCP has congestion detection scheme, it does not work when amount of traffic injected by multiple sources are very high. Congestion control schemes of ESRT, CODA and SenTCP do not consider the multimedia requirements and hence cannot meet the expected delivery rate. RMST on the other hand, congests the network by injecting extra retransmission packets resulting large amount of loss due to congestion. Therefore, RMST has the least energy efficiency which is nearly zero.

2.2.3.2 Probability of Successful Frame Delivery

The probability of successful frame transmission of each frame is shown in Figure 2.6. None of the transport protocols provide a reasonable frame delivery rate. RMST can only get the first frame across at the sink, since it loses the remaining frames while trying to recover all of the lost packets of the first frame with retransmissions. Note also that for all protocols, the probability of successful frame delivery

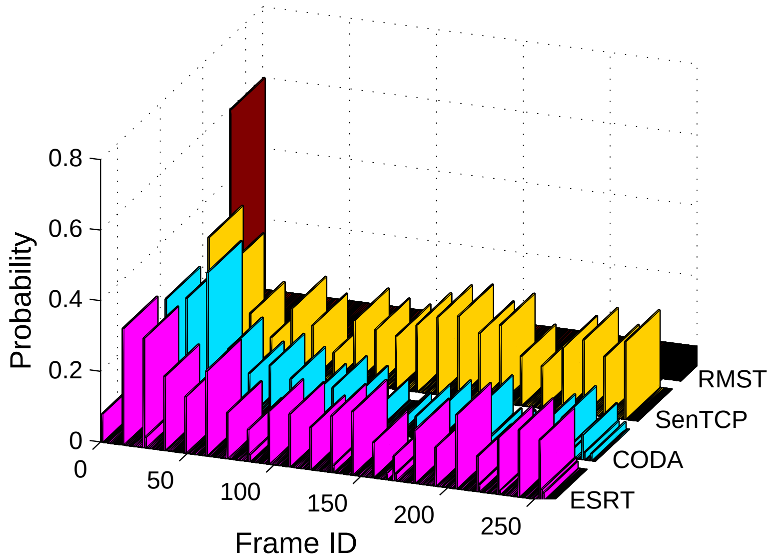


Figure 2.6: Successful frame delivery probability.

of initial frames is generally greater than the others as the network becomes severely congested as more frames are injected to the network. It is to be noted that even if data rate is regulated as a result of congestion resolution, due to packet based source rate regulation of existing protocols which is not suitable for multimedia communication where packets following one another is highly correlated, a reasonable frame delivery success can not be provided.

2.2.3.3 Frame End-to-End Delay and Cumulative Jitter

Frame end-to-end delay results in Figure 2.7 indicate that the observed delay for successfully delivered frames are very high, e.g., more than 1 second.

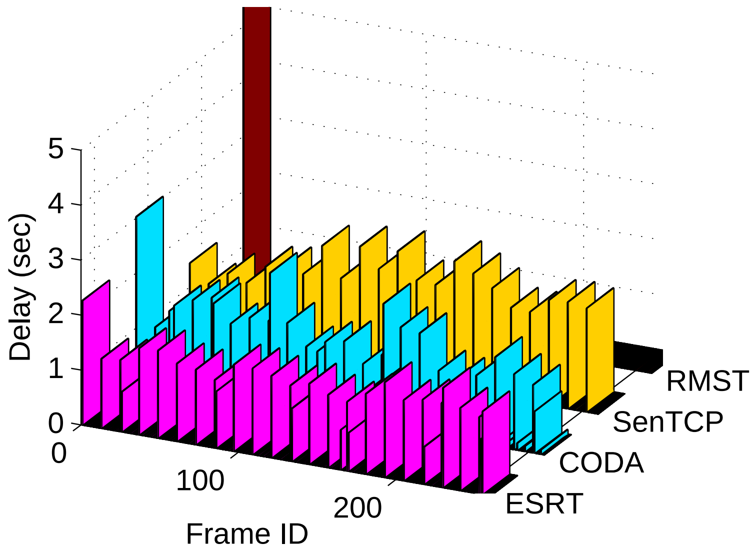


Figure 2.7: Frame end-to-end delay.

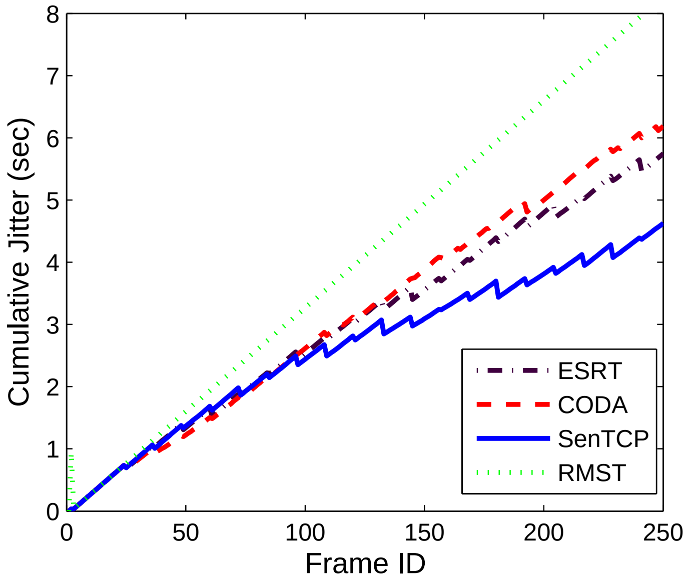


Figure 2.8: Cumulative frame jitter.

Note also that the delay of the first frame delivered by RMST is unacceptably high, (i.e., 170 seconds), because of the extra time spent while retransmitting the lost packets.

Cumulative frame jitter is defined as the variance of the inter-frame time. In Figure 2.8, it is shown that only the initial frames have an acceptable jitter while the jitter increases linearly along other frames. The frame loss gets bursty with time since the congestion in the network increases as more frames are injected, which increases the time difference between successfully delivered frames.

2.2.3.4 PSNR and Delay-Bounded PSNR

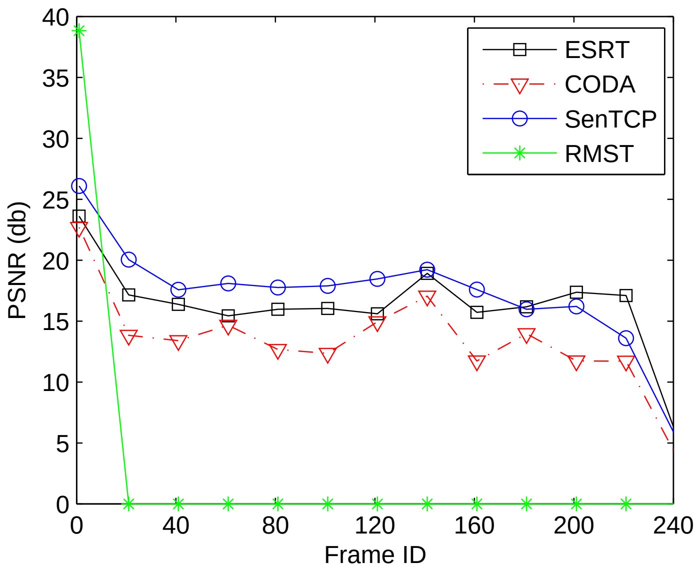


Figure 2.9: Frame PSNR.

An important video quality measurement which indicates the actual video quality perceived by the end-user is the peak signal-to-noise ratio (PSNR) of received frames. Typical PSNR of a frame should be more than 30 dB to provide a reasonable quality to the end-user. In Figure 2.9 the resulting PSNR of the received frames are shown. The PSNR levels change between 5 and 20 dB, which indicates poor received video quality.

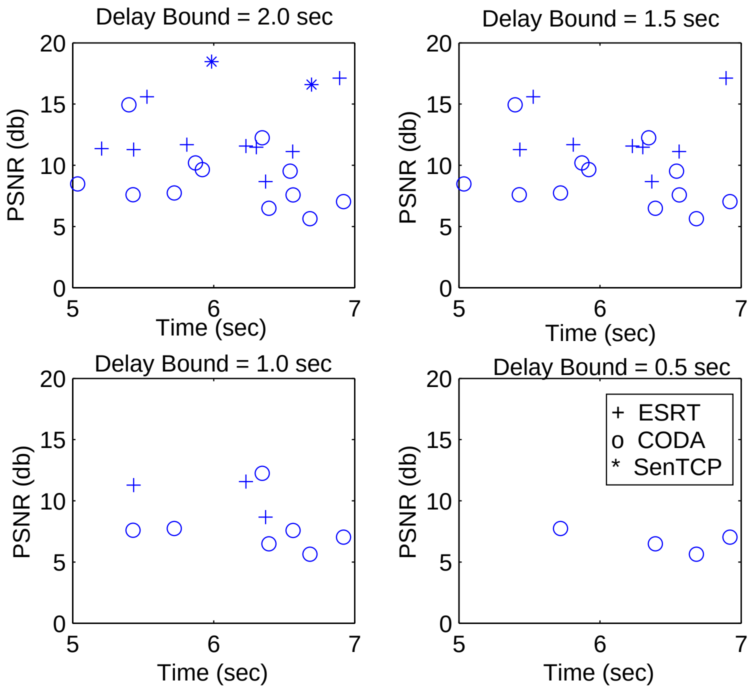


Figure 2.10: Delay-bounded frame PSNR (i.e. each point represents a received frame).

Delay-bounded PSNR is also measured to observe the effect of real-time delay bounds on the reliable communication objective for applications such as real-time video target tracking. Here, PSNR of the received frame should be at a certain level within a certain delay bound. In Figure 2.10, delay-bounded PSNR is shown for various delay bounds for a duration of 60 frame transmission. When delay bound is 2 seconds, most of the received frames satisfy the delay bound, so a certain level of PSNR (10 dB) is achieved. Number of received frames with required reliability level decreases with the application-specific delay bound. For a delay bound of 0.5 seconds, none of the protocols achieves delay-bounded PSNR objective.

In this chapter, simulation experimental results are presented for the performance of the existing transport protocols developed for WSN in multimedia communication. It is shown that the existing transport protocols provide very poor performance for real-time multimedia delivery in WSN. High bandwidth demand, energy-efficient reliable multimedia delivery, and real-time delay requirements are the most important challenges that need to be addressed by the new multimedia communication protocols for WSN. In the next chapter, a communication protocol (VSCP) is proposed that aims to overcome challenges of real-time multimedia delivery and provide reliable real-time video communication in WSN.

CHAPTER 3

VSCP: A REAL-TIME RELIABLE VIDEO COMMUNICATION PROTOCOL FOR WSN

In Chapter 2, shortcomings of existing WSN transport protocols in multimedia communication were presented in detail [1]. In this chapter, VSCP (Video Sensor Communication Protocol) is introduced which aims energy-efficient and reliable real-time video communication in wireless sensor networks. To the best of our knowledge, VSCP is the first proposal in this area that aims to achieve the objective of providing a complete transport solution for real-time video delivery in wireless sensor networks.

3.1 VSCP Overview

VSCP is a sink-based protocol with the main objective of; *providing reliable real-time video delivery from sensor nodes towards the sink in terms of required application reliability (i.e., aggregate multimedia reliability in terms of received video qualities and end-to-end frame delays as explained in Section 3.5.1) with minimum energy expenditure.*

In VSCP, when an event is detected by a number of source nodes, each source node starts to stream its own captured video towards sink. In the first decision interval (I_1) sink determines and broadcasts target frame PSNR (p^s) of each source node i, i.e., p_i^s to achieve desired aggregate multimedia reliability (R^*) in an energy-efficient way. Source node i upon receiving its target frame PSNR, i.e., p_i^s, encodes and sends new frames at that p_i^s. Each intermediate sensor node monitors its buffer level and queuing delay to detect congestion and notify the sink with the identifiers of source nodes causing congestion. In each decision interval sink measures the aggregate multimedia reliability (R), performs congestion decision based on the feedback from the intermediate nodes and updates p^s of each source node to achieve R^*.

In a WSN, information needed by sink is formed by collective effort of a number of sensor nodes within an event area. Hence, communication protocols should provide an event-to-sink reliable transport mechanism instead of an end-to-end reliability [2]. From this point of view, a reliability definition is needed by sink that covers application-specific requirements such as aggregate video quality, delay, energy usage and other parameters if any. However, this reliability definition is application specific. For example, an application may need to receive a low quality video from maximum number of sources, a limited number of but high quality video from some of the sources, or a target average video quality. Although definition of that reliability is application-specific, VSCP operates on the basis of the aggregate multimedia reliability (R) given in Section 3.5 where

$$1 - \epsilon < R^* < 1 + \epsilon \text{ where } 0 < \epsilon << 1 \qquad (3.1)$$

is an indication of reliable communication. In each decision interval (I_i), sink calculates R and takes necessary actions to achieve R^* (i.e., keep R close to 1).

For a video communication scenario, it is clear that aggregate multimedia reliability depends on the per source received video quality at least. Hence, to compute R at sink, a received video quality estimation is needed. During frame transmission some packets are lost causing a decrease in the received frame PSNR (p^r). During decoding process which is performed at sink, p^r is estimated from lost segments of that frame and target frame PSNR (p^s) sent by a source. Details of this estimation is given in Section 3.3 where basic idea is to exploit the decoder side information of lost pixels and decoder-specific error concealment process performed on lost pixels.

In VSCP, video encoding method used in sources (Section 3.2) is based on hybrid constant quality and constant bitrate coding. Each source node starts encoding the raw video to have a constant frame PSNR of p_e (i.e., $p^s = p_e$) which is the required frame PSNR of the application per each received video stream. In each decision interval sink estimates average PSNR of received frames ($\bar{p^r}$) of each video stream and if R^* can not be provided at current p^s of source nodes, it decreases or increases p^s according to network conditions. When any frame that is encoded at p^s results a data rate greater than maximum allowed data rate ($U(p^s)$), source node performs constant bitrate encoding at $U(p^s)$. In such a case, average p^s is kept constant at the expense of increased standard deviation.

One of the major functionalities of a transport protocol is to detect and resolve network congestion. In addition to huge amounts of data traffic generated by multimedia communication, multi-sender nature of sensor networks makes this feature more important. VSCP performs passive congestion detection based on queue level monitoring and average end-to-end frame delay, determines which sources are involved in congestion and decreases p^s of these source nodes in a controlled manner until congestion is resolved. When congestion is resolved, it starts to increase p^s again in a controlled manner until R^* is achieved. Details of this procedure is given in Section 3.4.

Efficient energy usage is a critical issue for battery limited sensor nodes. Percentage of dropped packets is an indication of energy usage efficiency since power consumed for lost packets is wasted. For real-time multimedia streaming applications packet loss may be caused by queue overflows due to network congestion, late losses due to strict delay constraints or channel loss due to wireless channel conditions. In a densely deployed sensor field, as the number of source nodes involved in event detection increases, the packet traffic generated by source nodes generally increases as packets are forwarded towards distant nodes (i.e., far from sink) to closer ones (i.e., close to sink) as a result of multi-sender one-receiver nature of communication. In the first decision interval, VSCP sets p^s of distant source nodes to a smaller value than closer source nodes. Main advantages of this method for multimedia communication in WSN can be summarized as;

- Decreasing congestion probability in the event area when traffic generated by distant sources are forwarded along near sources (i.e., distant sources are parent nodes of near sources).

- Balancing end-to-end frame delays of distant and near sources, thus providing a better performance in terms of delay variance of different flows.
- Providing the target aggregate reliability with decreased energy expenditure since total end-to-end energy usage of closer nodes are less than distant ones.

VSCP provides the above functionality with *Energy Efficient Rate Adaptation (EERA)* which is explained in Section 3.5.2.

3.2 Quality Adaptive Source Coding

As explained in Section 3.1, a combination of constant PSNR and constant bitrate [12] encoding is used at sources to control R by adapting p^s of source nodes according to network conditions. Any type of source coding can be used at source nodes as long as constant or nearly constant PSNR can be achieved at each frame. Since we use MPEG streaming for the performance evaluation, we obtained a nearly constant PSNR for each frame using a constant quantization parameter (QP), in order to obtain better results (i.e., less PSNR standard deviation), quantization parameters should be changed dynamically during encoding process. For each target frame PSNR p^s, there is a predefined maximum bit rate ($U(p^s)$). This causes each sent frame having a nearly constant PSNR (unless the resulting bit rate is greater than $U(p^s)$) while causing the resulting data rate fluctuating between 0 and $U(p^s)$.

In Table 3.1, coding parameters used throughout the simulations are shown. The sample video (Foreman [23]) is a 300 frame, 176x144 size QCIF video which is encoded with Berkeley MPEG encoder [22] at frame rate of 25 fps, 15 frame GOP size, IP...P GOP pattern. Different PSNR levels are obtained changing the I and P frame quality factors. Although quantization parameters should be changed dynamically during encoding process, as it can be seen from the table, even when the quality factors are chosen constant throughout the encoding process, a nearly constant PSNR which has a tolerable standard deviation can be achieved.

Table 3.1: Video Coding Parameters for Foreman [23] Sequence

QI, QP	$U(p^s)$ (Kbps)	Average Bit Rate (Kbps)	Average p^s (dB)	p^s Std. Deviation
31, 31	90	50	23.65	2.28
26, 24	95	55	24.02	2.38
20, 18	115	75	25.00	2.55
15, 13	150	107	26.12	2.69
12, 11	180	134	26.88	2.82
12, 10	190	147	27.22	2.73
9, 9	215	175	27.94	3.00
9, 8	240	199	28.43	2.86
9, 7	270	231	28.87	2.72
9, 6	315	275	29.54	2.53
7, 6	335	285	29.87	2.85
6, 6	340	291	30.08	3.05

When p^s is increased or decreased in a controlled manner, resulting source data rate will increase and decrease also, providing the sink to adjust R at the same time controlling the traffic injected to the network. In VSCP, that feature is used by sink while performing source rate adaptation which is explained in Section 3.5.2.

3.3 Estimation of Received Video Quality at Sink

In VSCP, target frame PSNR (p^s) of each source node is determined by sink. For each received frame a PSNR estimation is performed at sink using p^s and lost segments of that frame. In each decision interval average received frame PSNR ($\bar{p^r}$) of each video stream i, i.e., $\bar{p_i^r}$ is approximated and aggregate multimedia reliability (R) is computed.

Frame PSNR can be estimated at sink using number of lost pixels per frame and p^s of that frame sent by source node. Decoder performs an error concealment process to estimate pixel values of lost pixels which leads a difference between expected reconstructed pixel values and estimated values of those pixels. In [17] a general class of MPEG error concealment algorithms are provided. The structure of MPEG implies that if an error occurs within I-picture data, it will propagate through all frames in the Group of Pictures (GOP). Similarly, an error in a P-picture will affect the related P- and B-pictures, while B-picture errors will be isolated. Therefore, effective concealment of lost data in I-pictures and P-pictures to avoid error propagation effects is of critical importance in MPEG. 3 types of concealment algorithms are proposed and compared for one-tier and two-tier systems in [17].

These studies show that approximation of concealment error is decoder-specific and needs further research effort to obtain more accurate results. However, effect of this error on the final quality perceived by the end-user can be formulated as follows.

Target frame PSNR (p^s) of a source node is given by:

$$p^s = 20 \log_{10}(\frac{C}{\sqrt{T}}) \tag{3.2}$$

where C is a constant and T is total error between raw video frame (F) pixels and encoded source frame (S) pixels.

$$T = \sum_{i=1}^{X} \sum_{j=1}^{Y} (F_{i,j} - S_{i,j})^2 \tag{3.3}$$

where X and Y are number of rows and columns of frame and,

$$C = \frac{2^k - 1}{\sqrt{\frac{1}{XY}}} \tag{3.4}$$

where one pixel is represented by k bits.

If it is assumed that T is equally distributed over each sent pixel which is denoted by e,

$$e = T/(XY) \tag{3.5}$$

and if $\lambda_{i,j}$ is defined as the difference between $(i,j)^{th}$ sent and received pixel respectively,

$$\lambda_{i,j} = \begin{cases} 0 & \text{if pixel is correctly received} \\ S_{i,j} - S'_{i,j} & \text{if pixel is lost (concealment error)} \end{cases} \tag{3.6}$$

then total per pixel error of received frame, i.e., T' can be formulated as:

$$T' = \sum_{i=1}^{X} \sum_{j=1}^{Y} (F_{i,j} - S'_{i,j})^2 = T + \sum_{i=1}^{X} \sum_{j=1}^{Y} \lambda_{i,j}^2 + 2\lambda_{i,j}\sqrt{e} \tag{3.7}$$

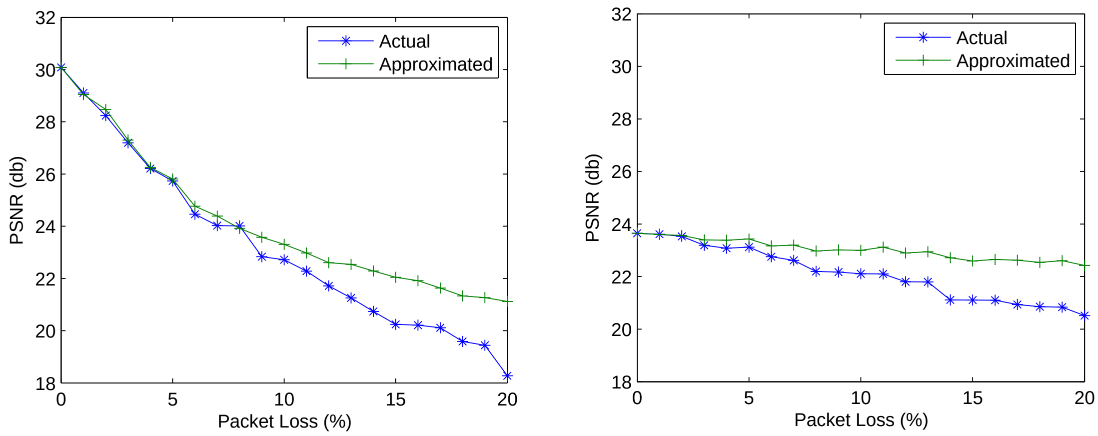
p^s is the target frame PSNR which is known to the sink and if *errors caused by concealment of lost pixels ($\lambda_{i,j}$) are approximately chosen*, final frame PSNR perceived by the end-user can be approximated as:

$$p^r = p^s - 10\log_{10}(1 + \frac{\alpha}{T}) \tag{3.8}$$

where α is the total pixel error caused by concealment process:

$$\alpha = \sum_{i=1}^{X}\sum_{j=1}^{Y} \lambda_{i,j}^2 + 2\lambda_{i,j}\sqrt{e} \tag{3.9}$$

Using (3.8) and (3.9), in figures 3.1 (a) and 3.1 (b) receiver side average PSNR approximation of a 300 frame length sample video sequence (Foreman) is shown for different packet loss rates when a constant concealment error selection (i.e., 10) and MPEG coding is used. To see the effects of estimation process on different quality videos, sample videos used are coded at 30.08 and 23.65 dB respectively.



(a) (b)

Figure 3.1: PSNR approximation with constant $\lambda_{i,j}$ (i.e., $\lambda_{i,j} = 10$) for 30 dB (a) 23 dB (b) sample video.

In figures 3.2 (a) and 3.2 (b) approximate concealment error is increased linearly with number of pixels lost within a frame. Although selection of each concealment error ($\lambda_{i,j}$) is decoder specific, when packet loss is less than 10%, difference between actual and approximated PSNR values do not exceed 1 dB even if a constant concealment error is chosen.

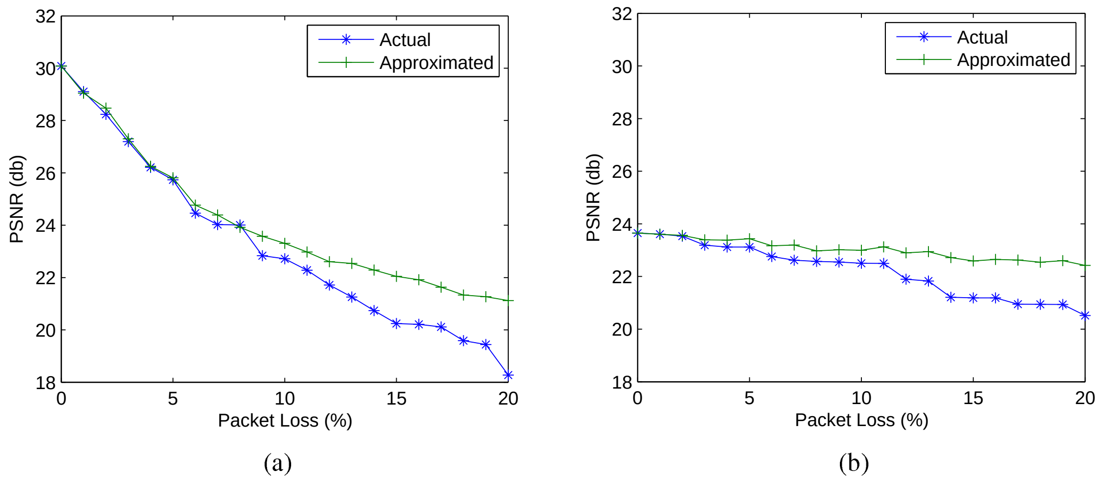


Figure 3.2: PSNR approximation with linearly increased $\lambda_{i,j}$ for 30 dB (a) 23 dB (b) sample video.

In VSCP, estimated received frame PSNR (p^r) is used by sink in calculation of aggregate multimedia reliability (R) and reliability adaptive rate control which are explained in sections 3.5.1 and 3.5.2.

3.4 Multimedia Congestion Detection

As explained in Chapter 2, the existing protocols use different methods to detect network congestion. However, it is observed that especially active congestion detection methods are not suitable for multimedia communication in WSN since they waste limited network bandwidth and does not provide good results. VSCP uses a passive congestion detection method specifically tailored for multimedia communication in WSN as outlined following.

Table 3.2: Video Packet

STREAM ID
QUEUE DELAY
CONGESTION FLAG
CONGESTION ID
END-TO-END DELAY
HOP COUNT
Payload

Algorithm 1: VSCP: Intermediate Nodes

```
qDelay = Average queuing delay
qLevel = Ratio of queue length to queue capacity
ω = Queue level threshold that implies congestion
if a packet received then
    record STREAM_ID
end
if a packet will be sent then
    if qLevel > ω then
        set CONGESTION_FLAG
        set CONGESTION_ID from recorded STREAM_ID
    end
    if qDelay > QUEUE_DELAY then
        QUEUE_DELAY = qDelay
        set CONGESTION_ID from recorded STREAM_ID
    end
    END-TO-END_DELAY = END-TO-END_DELAY + packet's service time
    HOP_COUNT = HOP_COUNT + 1
end
```

In Table 3.2 extra packet fields used by VSCP is shown. It is assumed that each packet has a video stream identifier that it belongs to. If an intermediate

Algorithm 2: VSCP Congestion Detection : Sink

```
congestedFrameCounter[STREAM_ID]: A data structure that holds number of
  frames that are marked as congested of video stream identified by
  STREAM_ID
frameCounter[STREAM_ID]: A data structure that holds number of frames
  that are received from video stream identified by STREAM_ID
congestedStreams[STREAM_ID]: A data structure that holds set of video
  stream identifiers that are involved in congestion of video stream identified by
  STREAM_ID
while not end of decision interval I_i do
    foreach received packet do
        if (CONGESTION_FLAG is set) OR (END-TO-END_DELAY > d_max)
        then
            // Counters are incremented at most by 1 per
               frame id
            increase congestedFrameCounter[STREAM_ID]
            increase frameCounter[STREAM_ID]
            insert CONGESTION_ID to congestedStreams[STREAM_ID]
        end
    end
end
foreach STREAM_ID do
    if congestedFrameCounter[STREAM_ID]/frameCounter[STREAM_ID]
    > τ then
        Video stream identified by STREAM_ID is congested
        congestedStreams[STREAM_ID] are the set of video streams involved
        in congestion
    end
end
```

node's average queuing delay (q_d) is greater than received packet's QUEUE_DELAY, QUEUE_DELAY field is set to q_d and CONGESTION_ID field is set to previously received (i.e., whether or not the packet is destined to itself) packet's STREAM_ID. If queue level of a sensor node exceeds a threshold ω, CONGESTION_FLAG is set and CONGESTION_ID field is set as explained. When sink receives a packet with CONGESTION_FLAG set, it marks currently receiving frame of video stream that the packet belongs to, as congested and records CONGESTION_ID which indicates the video stream involved in congestion (i.e., if any). If during a time interval (decision interval, I_i), percentage of congested frames exceeds τ, sink decides that video stream is congested.

The second aspect of congestion is due to average end-to-end frame delay of a video stream (END−TO−END_DELAY) greater than d_{max} which is maximum tolerable delay for the application. Although there is no queue overflow along the path between a source and sink, it is possible that END−TO−END_DELAY $> d_{max}$. In that case, sink is informed with maximum queuing delay along a path and video stream identifiers that cause this maximum delay. There is no time synchronization needed between sources and sink to obtain END−TO−END_DELAY of a packet since every intermediate node increases that field by its service time of that packet.

Algorithm 1 runs at intermediate sensor nodes. The only function that an intermediate sensor node performs other than congestion detection is increasing the HOP_COUNT field which is used in source rate adaption described in Section 3.5.2. Algorithm 2 runs at sink for congestion detection. Performing above congestion detection method, sink has the knowledge of congestion events and video streams which cause congestion.

There is a packet overhead of VSCP protocol. CONGESTION_FLAG field is one bit length. To obtain queuing delay and end-to-end delay with 10 msec precision within [0-2550] msec range, 8 bit is sufficient. If there are at most 1023 sources, each stream identifier can be coded with 10 bit. For a maximum of 255 hop-count 8 bit can be used. Under these assumptions a total of 45 bits, say 6 Bytes extra per packet overhead is induced. In [18] an efficient header compression technique is proposed for real-time services that achieves up-to 40:1 compression. Although existing techniques are mainly specialized for IP networks, similar results can be obtained by adapting these methods to sensor networks.

3.5 Reliability and Energy Usage Efficiency

3.5.1 Aggregate Multimedia Reliability

Reliable event detection is a crucial issue for a sensor network. However, not only reliability depends on various parameters such as delay, different quality levels of received data from multiple sensors, energy consumption and packet loss rates, it is also application-specific and different type of applications need different set of guarantees. This brings a challenge on a generic definition of reliability. In VSCP a reliability definition is introduced that covers aggregate video quality and end-to-end frame delay. If desired reliability definition is different than the one proposed, applying simple changes on the protocol stack can provide the desired result. VSCP assumes that during a time interval (decision interval) if m distinct video streams are received;

- Each received video stream i is expected to have an average frame PSNR, i.e., $\bar{p_i^r}$ greater than minimum usable frame PSNR (p_{min}) while less than maximum required frame PSNR (p_{max}).
- Average frame PSNR of all streams (i.e., $(\sum_{i=1}^{m} \bar{p_i^r})/m$) is expected to be at a certain level (p_e)
- Average end-to-end frame delay of a stream denoted by d_i should not exceed d_{max}.

In accordance with above assumptions, VSCP operates based on the aggregate reliability which is calculated as;

$$R = (\frac{1}{m * p_e} \sum_{i=1}^{m} \bar{p_i^r} D_i)^m \tag{3.10}$$

where m is the number of source node (video streams) and $D[i]$ is a delay factor defined as:

$$D_i = \begin{cases} 1 & \text{if } d_i < d_{max} \\ d_{max}/d_i & \text{otherwise} \end{cases} \tag{3.11}$$

According to equation (3.10) if each received video stream provides expected video quality p_e without exceeding the given delay bound (d_{max}), R equals to 1, which is the required reliability to safely detect the event. If any received video stream has a video quality less than p_e or average end-to-end frame delay greater than d_{max}, it will make the part of equation within parenthesis to be less than 1. However, PSNR is itself a logarithmic measure and even when there is a big decrease in the received video quality, PSNR decreases more slowly which may result a value (i.e., value within parenthesis) very close to 1. In order to better differentiate this case from safe event detection, m^{th} power of this value is performed to give R. In order to have a reliability of 1 in such a case, some other source nodes should provide more information (i.e., better quality video) to the sink, increasing R. On the other hand, when received video quality of some streams are greater than p_e without exceeding the delay bound, this time the expression within parenthesis will result a value greater than but close to 1. Again m^{th} power of this value differentiates it from 1. Distinguishing between required, high and low reliability is important since high reliability is an indication of waste of energy due to extra packet transmission while low reliability does not provide safe event detection. In that manner, R given in (3.10) simulates an aggregate multimedia reliability obtained from m sources in a WSN where

$$1 - \epsilon < R^* < 1 + \epsilon \text{ where } 0 < \epsilon << 1 \tag{3.12}$$

is an indication of reliable event detection.

In a WSN, required reliability should be provided with minimum energy expenditure and this requirement becomes more important for multimedia applications where there is high bandwidth demand that requires more energy usage. In the following section energy efficient rate adaptation (*EERA*) feature of VSCP is introduced which aims to obtain the desired aggregate multimedia reliability (R^*) in an energy efficient way.

3.5.2 Energy Efficient Rate Adaptation

VSCP exploits a main property of sensor networks in order to maximize energy usage efficiency, which is the collective effort of sensor nodes to gather the information needed by application. Since there are more than one sources in the event area, it is not needed each source to provide same amount of information to sink. Instead of setting each source to send at same data rate, it is desired to give more chance, and hence greater reporting frequency to sensors that can provide better quality information to sink in terms of required QoS parameters such as end-to-end frame delay, jitter and received video quality. When densely deployed nature of sensor networks is taken into account, sources that are closer to sink are best candidates to provide better communication quality at the same time using less energy due to decreased number of total transmissions along the path between source and sink. In a densely deployed sensor field, as the number of source nodes involved in event detection increases, the packet traffic generated by source nodes generally increases as packets are forwarded from distant nodes (i.e., far from sink) towards closer ones (i.e., close to sink) as a result of multi-sender one-receiver nature of communication. Again decreasing data traffic generated at distant sources generally decrease the congestion bottleneck that will arise around closer sources. Decreased congestion results less number of dropped packets, hence energy usage efficiency in terms of lost packets (i.e., energy is wasted for lost packets) increases.

Aggregate multimedia reliability (R) defined in Section 3.5.1 is obtained from aggregate information quality (i.e., video quality and delay) received from source nodes. Hence, target multimedia reliability (R^*) can be obtained with increased energy usage efficiency by setting a higher target frame PSNR (p^s) to the sources closer to sink than the distant ones.

In VSCP, to achieve this goal,

- When an event is detected by m sources, sink is informed by end-to-end hop-count between each source and sink.
- An energy-optimized source rate regulation is performed that sets p^s of each source node to achieve R^* (i.e., $R = 1$).

If it is assumed that each wireless link is identical, has a constant bit error probability and there is no congestion in the network, transmission energy usage per received video stream can be approximated from number of transmitted bits along the path from each source to sink. Let,

ζ Successful bit transmission of wireless channel.

h_i Number of hops between i^{th} source node and sink.

b_i Data rate of i^{th} source node in bits/second.

c Energy usage per bit transmission.

Total energy usage of i^{th} video stream can be approximated as,

$$E_i = b_i c \sum_{j=0}^{h_i - 1} (\zeta)^j \tag{3.13}$$

and if there are m sources, overall energy consumption resulting from transmitted packets is;

$$E_T = \sum_{i=1}^{m} E_i \tag{3.14}$$

Consequently, the aim of VSCP is to provide target aggregate multimedia reliability, i.e., R^* with minimum energy consumption, so to maximize

$$R^*/E_t \text{ where } 1 - \epsilon < R^* < 1 + \epsilon \tag{3.15}$$

In order to minimize E_T in the above equation while having $R^* = 1$, VSCP performs the following rate adaptation to determine initial target frame PSNR of source nodes when information from the event area is just started to be received by sink.

3.5.2.1 Determination of Initial Target Frame PSNR

In the first decision interval after an event is detected, VSCP performs an energy efficient source rate regulation. Since each video stream has an identifier and each packet has a hop-count field, sink has the information of number of sources and corresponding hop-counts between each source and sink.

We use the terms *set* and *array* interchangeable where for any set A, $A[i]$ is the i^{th} element of A and $M(A)$ is the mean (average) of elements of A.

In Section 3.5.1 we stated that each received video stream is required to have an average frame PSNR greater than p_{min} and less than p_{max}. Let assume that the sink selects target frame PSNR (p^s) of each source from a finite set (V) of p^s values where $p_{min} < V[i] < p_{max}$. Hence, if there are m source nodes, any m element permutation of V denoted by V_j^m is a candidate set of p^s values that can be assigned to source nodes where $V_j^m[i]$ is the p^s of i^{th} source node.

$$V_j^m \text{ is the } j^{th} \text{ m element permutation of V} \tag{3.16}$$

$$V_j^m[i] \text{ is the } p^s \text{ of } i^{th} \text{ source node} \tag{3.17}$$

When the event is first detected by m source nodes, each source node starts to encode the frames at p_e dB which is the expected average received PSNR of all video streams. Hence, at start-up we can assume that there is a default set (i.e., m element permutation of V) of p^s values assigned to source nodes where each element equals to p_e. If we denote that set by D,

$$(\forall i)\ D[i] = p_e \tag{3.18}$$

In Section 3.2 we defined $U(p^s)$ as the maximum allowed bit-rate that will result while coding a video stream at p^s dB PSNR. For a given set of p^s values denoted by H, we define $B(H)$ as the maximum source bit-rate when target frame PSNR of each source node i is assigned according to $p_i^s = H[i]$,

$$B(H) = \sum_{i=1}^{m} U(H[i]) \tag{3.19}$$

In Section 3.5.2, equation (3.13) we proposed a method to estimate total end-to-end energy usage for a given source bit-rate and end-to-end hop-count. For a given set of p^s values denoted by H we define $W(H)$ as the maximum total energy usage when target frame PSNR of each source node i is assigned according to $p_i^s = H[i]$,

$$W(H) = \sum_{i=1}^{m} E_i \tag{3.20}$$

where $b_i = U(p_i^s)$ in E_i formula given in equation (3.13)

According to the R definition given in equation (3.10), if p^s of each source node is set to p_e using the default set D, expected value of R will be 1. However this setting is not efficient in terms of energy usage efficiency as explained in Section 3.5.2 and it is required that sources that are closer to sink should send at a higher data rate than others. In order to perform an energy efficient source rate adaptation (*EERA*) in the first decision interval we need the following requirements to be satisfied by the desired set of p^s values denoted by H^*:

$$M(H) = p_e \quad (3.21)$$

$$B(H) < B(D) + \sigma \quad (3.22)$$

$$(\forall \text{set } H \text{ that satisfies equations (3.21) and (3.22)})\ W(H^*) <= W(H) \quad (3.23)$$

First requirement (equation (3.21)) sets average target frame PSNR to p_e, thus expected value of R is set to 1 which is the required aggregate multimedia reliability to safely detect the event. Second one (equation (3.22)) guarantees that network traffic is not more than (i.e., $\sigma << B(D)$) the one that results when each source encodes the frames at p_e dB. Finally, when third requirement (equation (3.23)) is satisfied, the target set of p^s values (set H^*) is found where total energy usage is minimized, desired aggregate multimedia reliability, i.e., R^* is set to 1 and total source data rate (i.e., network traffic) is bounded with the one that will result when p^s of each source node is set to expected average frame PSNR (p_e).

VSCP searches throughout each m element permutation of V and finds the desired set of p^s values (H^*) that satisfies equations (3.21), (3.22) and (3.23) and sets initial target frame PSNR of each source node accordingly (i.e., $p_i^s = H[i]$).

3.5.2.2 Source Rate Regulation During Congestion

As explained in previous section, initially (i.e., in the first decision interval, I_1) the sink assigns target frame PSNR (p^s) of each source node to achieve target reliability ($R = 1$) in an efficient way. In each decision interval except first one ($I_i, i > 1$), the sink first checks if there is congestion in the network. When there is congestion along a path between some source nodes towards the sink, sink is informed by the identifiers of source nodes (video streams) that are involved in congestion as explained in Section 3.4. In such a case, sink decreases p^s of these source nodes setting the new p^s value between average of received frame PSNR ($\bar{p^r}$) and p^s.

Algorithm 3: VSCP Congestion Resolution

```
// p̄_i^r:  Average received frame PSNR of i^th video
   stream
// p_i^s:  Target frame PSNR of i^th video stream
```

if *there is congestion* **then**
    foreach *source node id* (i) *involved in congestion* **do**
        $p_i^s = \bar{p_i^r} + \left(\frac{p_i^s - \bar{p_i^r}}{2}\right)$
    end
end

Since $\bar{p_i^r} < p_i^s$ when there is congestion, setting new target frame PSNR as given in Algorithm 3 provides a better quality video in the next decision interval due to decreased packet loss caused by congestion. If congestion persists in the following decision intervals, VSCP just re-applies Algorithm 3. When congestion is resolved but aggregate multimedia reliability is not sufficient ($R < 1 - \epsilon$), it starts to increase p^s of each source node in an energy efficient way which is explained in the next section.

3.5.2.3 Determination of Incremental Target Frame PSNR

In Section 3.5.2.1, we defined H as a set of target frame PSNR (TFP) values that can be assigned to source nodes (i.e., $p_i^s = H[i]$) by the sink and H^* is the desired set of p^s values for efficient initial TFP setting of source nodes. We define H' as the set that includes average frame PSNR of each video stream that is received during a decision interval when p^s of each source node is assigned according to set H.

$$(\forall i)\ H[i] = p_i^s, H'[i] = \bar{p_i^r} \tag{3.24}$$

When there is no congestion in the network, R may be less than target reliability ($R < 1 - \epsilon$). It means that average PSNR of frames received from all source nodes ($M(H')$) is less than expected per stream average frame PSNR, p_e. In order to increase R in a controlled manner (i.e., not causing network congestion), average p^s of source nodes ($M(H)$) can be increased by β where $\beta << p_e$. In Section 3.5.2.1 we have already described how to assign initial p^s of each source node to have $M(H) = p_e$. We can use the same method to have $M(H) = M(H) + \beta$. However, this method directly searches a set of p^s values that sets p^s of sources closer to sink a greater value than others. If during congestion resolution the source nodes that were involved in congestion are these closer nodes, the method given in Section 3.5.2.1 tries to first increase p^s of these closer source nodes which increases probability of a congestion event in the next decision intervals. In order to prevent this case, it is required that p^s of *each* source node is increased. We define following requirements to be satisfied by the target set of p^s values denoted by L^* to increase $M(H)$ by β:

$$M(L) = M(H) + \beta \tag{3.25}$$

$$(\forall i) L[i] > H[i] \tag{3.26}$$

$$(\forall L \text{ satisfying equations (3.25) and (3.26)}) \; W(L^*) <= W(L) \tag{3.27}$$

When first requirement (equation (3.25)) is satisfied, average p^s is increased by β, thus expected value of R is increased. Second one (equation (3.26)) indicates that p^s of each source node is increased. Finally, when third requirement (equation (3.23)) is satisfied, the target set of p^s values (set L^*) is found.

As in the case in Section 3.5.2.1, VSCP searches throughout each m element permutation of V (i.e., the set of PSNR values) and finds the set (L^*) that satisfies equations (3.25), (3.26), (3.27) and sets new target frame PSNR of each source node accordingly (i.e., $p_i^s = L^*[i]$).

3.5.2.4 Overall Rate Adaptation Algorithm

In Section 3.5.2.1 determination of initial target frame PSNR (TFP) were presented. During a time duration of length decision interval, VSCP records stream identifiers (i.e., stream identifier can be used as source identifier) and number of hops between each source node and sink. At the end of any decision interval if number of hops of a source node that was recorded in the previous decision interval changes or number of source nodes within the event area changes, VSCP treats this decision interval as the first one and performs initial TFP assignment. After on, in each decision interval Algorithm 4 is run at sink.

As it can be understood from Algorithm 4, after first source rate regulation if there is congestion in the network VSCP just regulates source nodes that are involved in congestion until congestion is resolved. If there is no congestion VSCP checks if R is close to 1 and if not, it performs related source rate regulation accordingly. In the next section performance evaluation results of VSCP in real-time MPEG streaming scenarios are presented.

Algorithm 4: VSCP Rate Adaptation Algorithm

// $\bar{p_i^r}$: Average of frame PSNR received from i^{th} video stream
// p_i^s: Target frame PSNR of i^{th} video stream
// H: The set of p^s values such that $p_i^s = H[i]$
// getInitialConfiguration(p_e): A function that returns the set of initial p^s setting as explained in Section 3.5.2.1
// getConfiguration(H,β): A function that returns the set of incremental p^s setting as explained in Section 3.5.2.3
// broadcastConfiguration(H): A function that broadcasts p^s of each source node according to $p_i^s = H[i]$
// I_i: i^{th} decision interval

if *decision timer expired* **then**
  if $I_i = I_1$ **then**
    H = getInitialConfiguration(p_e)
    broadcastConfiguration(H)
  end
  else
    if *there is congestion* **then**
      foreach *source node id (i) involved in congestion* **do**
        $p_i^s = \bar{p_i^r} + \left(\frac{p_i^s - \bar{p_i^r}}{2}\right)$
        $H[i] = p_i^s$
      end
      broadcastConfiguration(H)
    end
    else if $R < 1 - \epsilon$ **then**
      H = getConfiguration(H,β)
      broadcastConfiguration(H)
    end
    else if $R > 1 + \epsilon$ **then**
      H = getInitialConfiguration(p_e)
      broadcastConfiguration(H)
    end
  end
end

3.6 Performance Evaluation

3.6.1 Real-time Video (MPEG-1) Streaming

The aim of performance evaluation performed in this section is to introduce VSCP with all its operational aspects. Hence, sensor node capabilities are chosen carefully (i.e., increased bandwidth and queue size) to enter the protocol in each state. Otherwise, at start-up there would be congestion and we could only observe congestion detection and resolution features of VSCP. VSCP parameters used during real-time MPEG streaming simulations are shown in Table 3.3. Again application specific VSCP parameters such as minimum usable, maximum required and expected average frame PSNR values are chosen to preserve that objective. In Section 3.6.2 where VSCP is compared with existing transport protocols, sensor capabilities (i.e., bandwidth and queue size) are decreased and expected average frame PSNR parameter is increased to the values used in Section 2.2 where performance evaluation results of existing protocols are presented.

Table 3.3: VSCP Simulation Parameters

ω	0.75
τ	0.15
ϵ	0.3
d_{max}	600 msec
p_{min}	23.65 dB
p_{max}	30.08 dB
p_e	26 dB
β	1 dB
ζ	$1 - 10^{-5}$
σ	100 Kbps
I	12 sec

Simulation experiments are performed using ns-2 [19]. 120 nodes are randomly placed in 200m X 200m field. At start-up, 4 source nodes are randomly selected within an event area of radius 70 m. After two decision intervals, two additional sources are involved to the event detection. Sink is randomly located within the field. Each node has a queue size of 200, radio range of 40m, 10Mbps 802.11 MAC where link layer RTS/CTS exchange mechanisms are disabled. Directed Diffusion [5] is used as the routing protocol. Simulations are performed for 25 times using random deployment and the results are averaged. Each decision interval is 12 seconds. Simulations are carried for 10 decision intervals.

The sample video (Foreman) is a 300 frame, 176x144 size QCIF video which is MPEG-1 [15] encoded at frame rate of 25 fps , 15 frame GOP size, IP...P GOP pattern. Different PSNR levels are obtained changing the I and P frame quality factors as explained in Section 3.2. The sample video is repeated to obtain a longer video sequence. Due to limitations of our simulation environment, the decision interval (I) duration is set to 12 seconds which is the duration of the sample video. At the end of each decision interval the sample video is re-encoded for each source node according to the new target frame PSNR set determined by VSCP and during next decision interval those videos are streamed by the source nodes.

Berkeley MPEG encoder [22] is actually an MPEG-1 encoder and provides *per-slice encoding* such that each row of an MPEG frame is self-decodable. This feature provides better quality videos in real-time video streaming applications in WSN since when there is packet loss due to channel errors or congestion, successfully received slices of a frame can be decoded without corruption (i.e., independent of lost slices). Hence, instead of using MPEG-4 [16] encoding we preferred using MPEG-1 [15] encoding during simulations.

3.6.1.1 Aggregate Multimedia Reliability

In Figure 3.3 aggregate multimedia reliability (R) obtained in each decision interval is shown. In the first decision interval, reliability is below required value. This means some sources provide average end-to-end delay greater than d_{max} while some others provide poor video quality. Since this is the first decision interval, VSCP determines initial TFP of each source node. It is observed that, after first source rate regulation, R is increased and it is within the required bounds. This simply shows that, although network capacity is sufficient to provide the target reliability, when each source node sends at the same TFP, R drops below 0.3 because of increased end-to-end frame delays of distant sources and increased congestion. However, *VSCP's energy efficient rate adaptation (EERA) feature decreases initial TFP of distant source nodes while increasing near ones, thus balancing the end-to-end frame delays and decreasing congestion.*

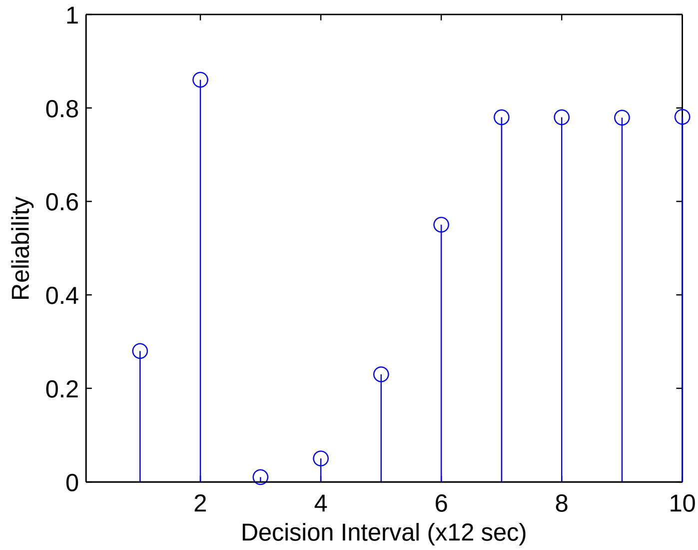


Figure 3.3: Aggregate Multimedia Reliability.

Between second and third decision intervals 2 more source nodes are involved to the event detection. It is observed that in the third decision interval R is nearly zero because of high network congestion level. In the third decision interval VSCP observes that 2 more source nodes are involved to the event detection, again it performs its initial source rate regulation. As a result of this first regulation it is observed that R is increased in the next decision interval, but there is congestion in the network. VSCP starts to decrease TFP of each source that is involved to the congestion. It is observed that when TFP is decreased during congestion, R increases in each decision interval. In the sixth decision interval congestion is resolved but R is below the minimum required reliability which is 0.7. VSCP increases each TFP in a controlled manner and in the next interval required reliability is provided to the application.

3.6.1.2 Energy Usage Efficiency

In Figure 3.4 total energy usage of sensor nodes are shown for each decision interval. While initial source sending rates result a more energy usage (i.e., intervals 1 and 3), VSCP provides more than 10 % energy saving as a result of its first source rate regulation (i.e., intervals 2 and 4). *Is is observed that at start-up, VSCP decreases overall energy usage while increasing the aggregate multimedia reliability.* As it can be seen from figures 3.3 and 3.4, during congestion resolution (i.e., 5. and 6. decision intervals) energy usage is decreased at the same time resulting a better reliability.

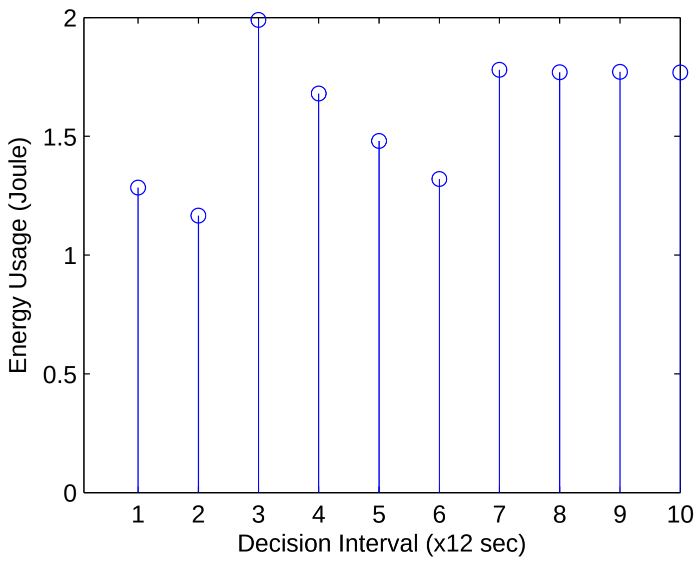


Figure 3.4: Energy usage (1 μJ per bit energy assumed).

In Figure 3.5 percentage energy saving of VSCP is shown with respect to initial source sending rates. It is observed that VSCP provides up-to 15% energy saving with 4 source nodes, while up-to 35 % during congestion resolution when there are 6 source nodes. After target reliability is provided, energy saving is about 10 % because of increased source rates.

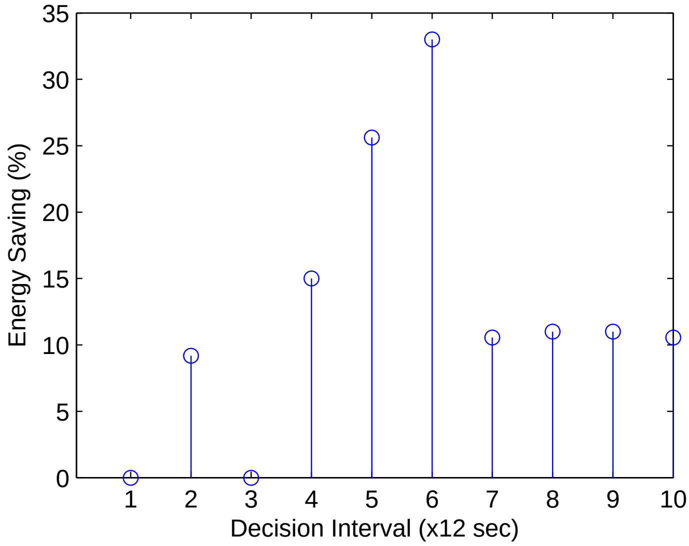


Figure 3.5: Energy saving.

3.6.1.3 Average Frame Delay and Cumulative Jitter

In Figure 3.6 average end-to-end frame delay is shown for each decision interval. In the first decision interval with 4 source nodes, average frame delay is greater than d_{max}. After initial TFP determination, delay is dropped below d_{max}. *This observation shows that, not only overall energy consumption but also average end-to-end frame delay is decreased by VSCP's initial TFP determination.* In the 3. decision interval where there are 6 source nodes within the event area causing a network congestion, average frame delay is about $2d_{max}$. This time after first rate regulation (i.e., 4. decision interval), average frame delay is higher than d_{max}. This shows that when there is high level of congestion, VSCP may not provide required delay guarantee at first attempt.

After congestion is resolved, VSCP guarantees that there is no video stream with average frame delay greater than d_{max} since this condition is treated as congestion as described in Section 3.4. Hence, in the 6. decision interval where congestion is resolved, average end-to-end frame delay is decreased below d_{max}. This observation is important in terms of *delay-bounded reliability* since for a real-time video

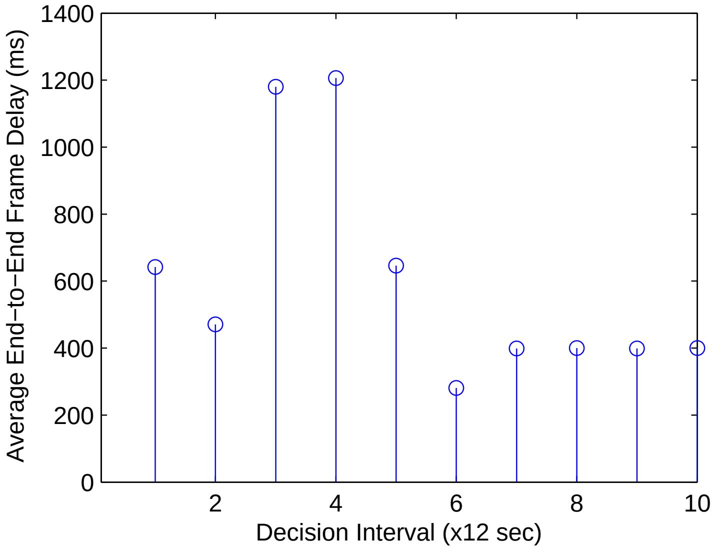


Figure 3.6: Average end-to-end frame delay.

streaming application there is a maximum delay bound on received frame delays which causes frames having larger delays to be useless for the application. VSCP aims to provide the target reliability in a time duration less than that delay-bound.

In Figure 3.7 cumulative frame jitter is shown for each decision interval . For a given time instant cumulative jitter shows the variance of frame jitter starting from first received frame until that time. It is observed that in 2. and 4. intervals where first source rate regulation is performed, cumulative jitter is decreased. Although that decrease is not much when there are 4 source nodes, there is a reasonable amount of decrease in cumulative jitter when there are 6 source nodes in the event area such that initial TFP of source nodes cause a network congestion. During congestion resolution (i.e., 5., 6. intervals) cumulative jitter is decreased down to acceptable levels. Finally, in 7. interval, where TFP of source nodes are increased to provide target reliability, cumulative jitter increases within acceptable bounds.

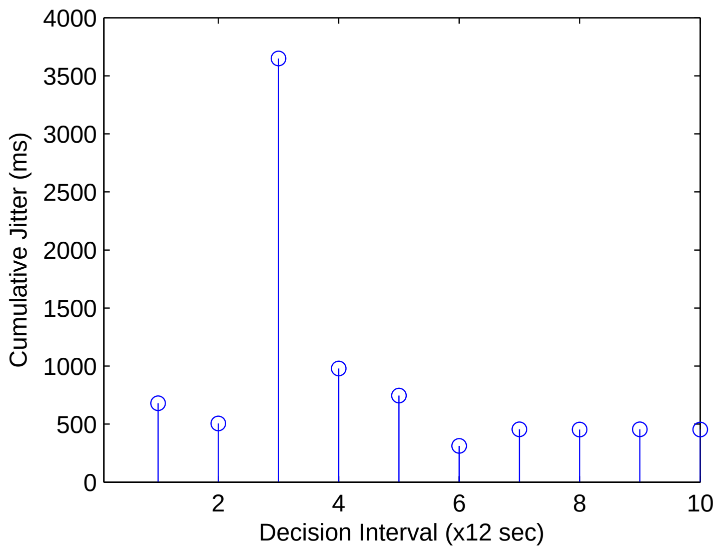


Figure 3.7: Cumulative end-to-end frame jitter.

3.6.1.4 Successful Frame Delivery Probability

In Figure 3.8 successful frame delivery probability (SFD) is presented for each decision interval. Since network capacity is sufficient when there a 4 source nodes, until 3. interval SFD is more than 0.9. However, when number of source nodes increases, SFD drops to 0.5 immediately. SFD is increased up-to 0.8 due to decreased congestion after initial TFP assignment. Since VSCP detects congestion in the next interval, it decreases TFP of sources involved in congestion, thus increasing the SFD up-to 0.91. In the next interval TFP of each source node is increased to increase R which decreases SFD to 0.9.

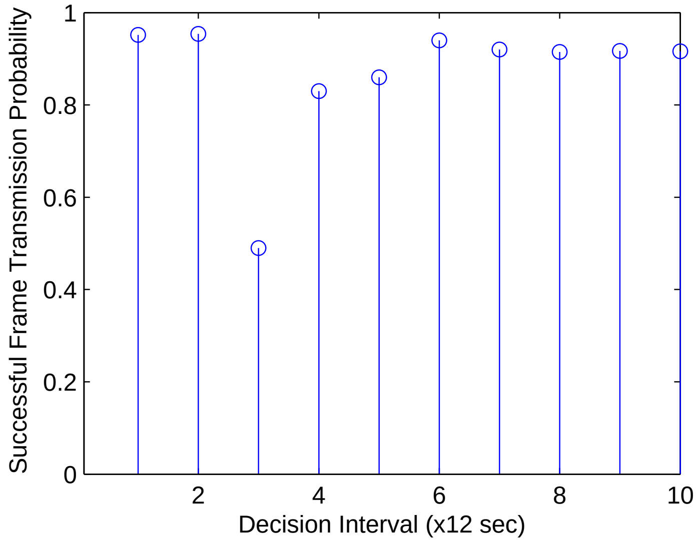


Figure 3.8: Probability of successful frame delivery.

3.6.2 Comparison of VSCP with Existing Protocols

The aim of performance evaluation performed in this section is to compare VSCP with existing protocols. The simulation scenario and sensor node capabilities are set same with the one used in Section 2.2 where performance evaluation results of existing protocols are presented. Same VSCP parameters given in Section 3.6.1 are used with an increased required average frame PSNR (i.e., $p_e = 38$ dB) since this is the initial video quality and for each protocol we expect to receive that video quality.

3.6.2.1 Average Received PSNR Comparison

In Figure 3.9 average received PSNR is shown for each protocol. In the first decision interval time of VSCP (i.e., t=10), each protocol provides a poor video quality changing between 15 and 20 dB. Indeed 10 second is sufficient for other protocols to enter a stable state, so after on, their average frame PSNR does not change much. However, VSCP just starts its operation by its first source rate regulation. As a result of this regulation, in the second decision interval it provides a 5 dB better average video quality than other protocols which is a reasonable amount of quality increase.

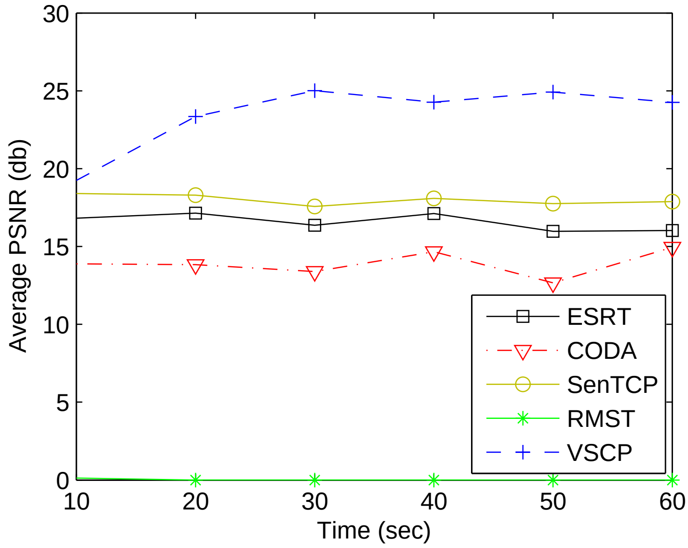


Figure 3.9: Average received PSNR comparison (Foreman video sequence).

Network is congested and cannot carry that much traffic, so VSCP decreases source sending rates in the second and third decision intervals. After congestion is resolved, VSCP increases each source rate in a controlled manner, however, in each attempt the network is again congested, hence it again decreases source sending rates. This process continues providing an average of 25 dB video quality.

In order to see the effect of different video characteristics on received PSNR, we performed same simulations one more with a different sample video which is Coastguard [23]. In Figure 3.10 average received PSNR is shown for Coastguard video sequence. When two sequences are encoded to obtain same average PSNR, Coastguard sequence results more bandwidth than Foreman sequence, hence VSCP can provide 22 dB video quality which is smaller than the one obtained with Foreman sequence. However, there is a significant amount of quality difference between VSCP and other protocols since increased bandwidth demand causes other protocols to provide very poor video quality.

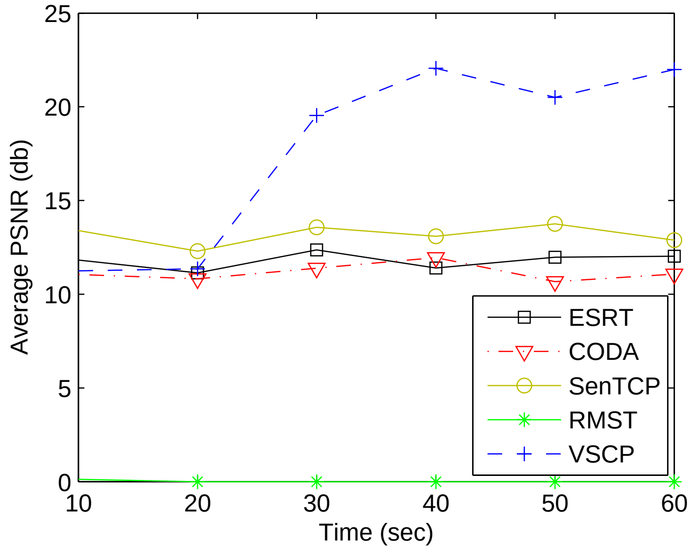


Figure 3.10: Average received PSNR comparison (Coastguard video sequence).

3.6.2.2 Energy Usage Efficiency Comparison

In Chapter 2 energy usage efficiency of protocols are presented by comparing ratio of received and total sent packets in the network. In Figure 3.11 energy usage efficiency of VSCP is compared with existing transport protocols. While other protocols cause most of the packets to be lost (i.e., loss due to congestion, late losses) wasting sensor nodes' energy, after second decision interval VSCP resolves the congestion in the network and provides a 90 % energy efficiency. This observation shows that VSCP's multimedia specific congestion detection and resolution method and energy efficient source rate regulation outperforms existing protocols in terms of energy usage efficiency.

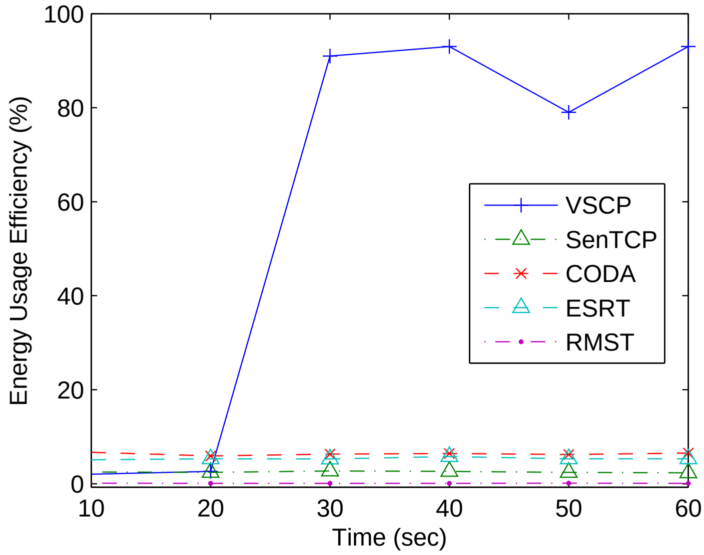


Figure 3.11: Energy usage efficiency comparison.

3.6.2.3 Average End-to-End Frame Delay & Cumulative Jitter Comparison

In figures 3.12 and 3.13 average end-to-end frame delay and cumulative frame jitter are shown. Since other protocols can not resolve congestion effectively, they provide high end-to-end frame delay and jitter. VSCP detects congestion and decreases source rates accordingly. After 3. decision interval, it satisfies required maximum delay requirement of application. Although congestion is resolved in the 3. decision interval, when VSCP tries to increase the aggregate multimedia reliability, it causes a tolerable increase in frame delays.

Guaranteed maximum delay (d_{max}) feature of VSCP is important in terms of *delay-bounded* PSNR, since frames having end-to-end delay greater than d_{max} are useless and dropped by the application. As shown in Section 2.2.3.4 when delay-bound is decreased below a certain threshold which is nearly 1 sec, none of the existing protocols provide a reasonable frame delivery. In Figure 3.12 it is observed that in VSCP nearly all frames are delivered to sink with an end-to-end average delay less than d_{max}, thus preserving the average PSNR observed in Figure 3.10.

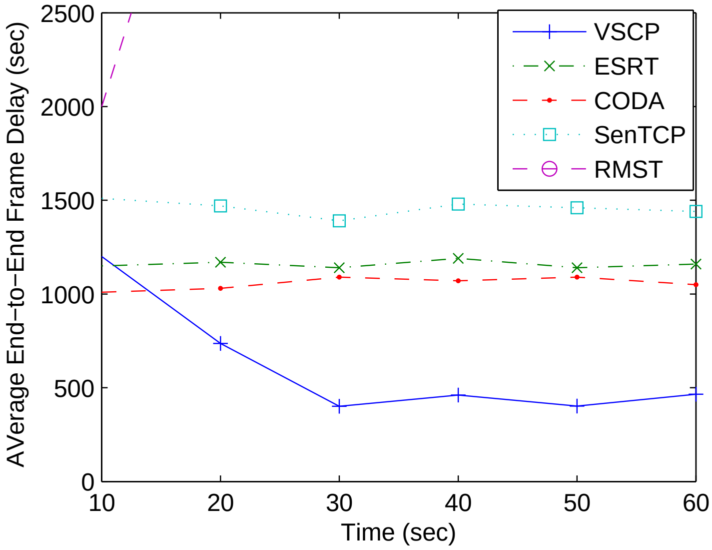


Figure 3.12: Average end-to-end frame delay comparison.

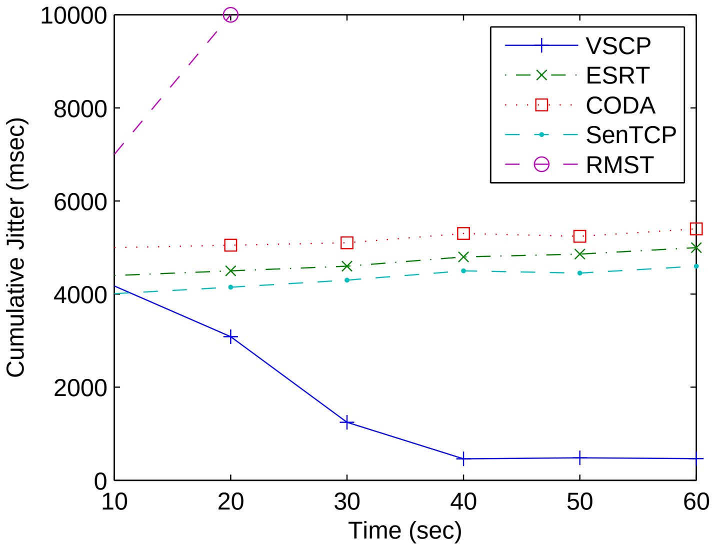


Figure 3.13: Cumulative jitter comparison.

3.6.2.4 Successful Frame Delivery Comparison

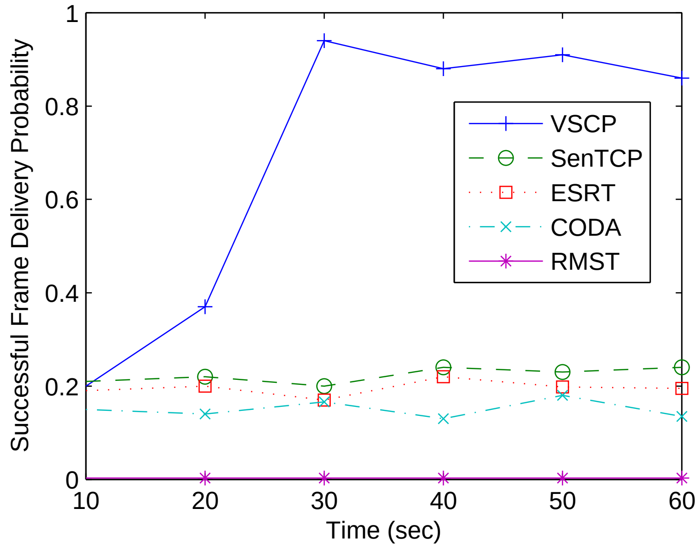


Figure 3.14: Probability of successful frame delivery probability.

In Figure 3.14 existing protocols are compared with VSCP in terms of successful frame delivery probability (SFD). After two decision intervals VSCP outperforms existing protocols and provides a SFD of 0.92. As discussed in previous sections VSCP's reliability adaptive source rate adaptation and multimedia-specific congestion detection methods decrease contention bottleneck in the event area increasing SFD. Secondly, VSCP performs multimedia-specific source rate adaptation by changing the quality of encoded video instead of decreasing frame rate which guarantees to stream the video at a given frame-per-second.

CHAPTER 4

CONCLUSIONS

Real-time multimedia streaming in wireless sensor networks exposes a wide range of new WSN applications such as target tracking, remote surveillance, environmental monitoring, discovering and following rare animals, source localization. However, in addition to limitations of sensor nodes especially in terms of processing power, battery and communication bandwidth, reliable multimedia communication brings extra challenges due to its strict QoS requirements such as high bandwidth demand and end-to-end delay bounds. Therefore, communication protocols should provide these strict multimedia requirements with minimum energy in order to increase network lifetime.

Sensor networks are generally event-based and information required by sink is obtained by collective effort of sensor nodes within the event area. Hence, in order to provide reliable communication to the undergoing application, communication protocols should be reliability-aware and provide aggregate reliability obtained from source nodes involved in communication. Since real-time multimedia data received from different source nodes differs in end-to-end delay, bandwidth, jitter, power consumption, signal-to-noise ratio, etc. which effects application reliability, it is crucial for the communication protocol to manage each source-to-sink flow to provide reliable communication.

In terms of communication protocol stack, event-to-sink reliability can best be provided by exchanging data between various communication layers, thus using a cross-layer approach. Indeed, reliable data transport is the main objective of transport layer. However, reliability depends on various parameters such as delay, different quality levels of received data from multiple sensors, energy consumption and packet loss rates and the only way to obtain an overall reliability measurement is combining information gathered from different layers of protocol stack such as MAC layer for queuing delays, application layer for quality of decoded multimedia data, network layer to obtain statistics of different data flows and so on.

Video Sensor Communication Protocol (VSCP) which is proposed in this thesis, is a sink-based, cross-layer protocol that aims to provide reliable real-time video communication with minimum energy in wireless sensor networks. It estimates each received video quality using lost segments of frames, computes aggregate multimedia reliability using received video qualities, end-to-end frame delays and overall energy usage of sensor nodes. It aims to provide target reliability with minimum energy performing an energy efficient source rate regulation. It uses queuing delays and queue levels to detect congestion. Whenever congestion is detected it regulates source rates in a controlled manner until congestion is resolved. After congestion is resolved, it increases source rates in an energy efficient way, thus increasing the observed reliability with less energy expenditure.

Although simulation results show that VSCP outperforms existing protocols especially in terms multimedia communication metrics, some operational blocks can be improved to obtain better results. For example, in the first decision interval the protocol performs an initial source rate regulation according to number of hops between each source node and sink. If there is a contention bottleneck around the source nodes that are closer to sink, initial source rate regulation may cause an increased contention. As it can be seen from the simulation results, in such a case VSCP resolves the congestion but only after a number of decision intervals have passed. Hence, initial target frame PSNR assignment can be improved by combining the hop-count information with identification of source nodes at contention bottlenecks, thus improving protocol robustness. Secondly, VSCP assumes that there is no background traffic in the network. However, this is not the case when there are different type of traffics. Therefore, congestion detection and resolution methods can be improved to handle background traffic. Thirdly, VSCP does not handle multiple events and assumes that all video streams are sent from same event area. And lastly, during source rate regulation VSCP only changes video qualities of source nodes. However, there may be some other application requirements such as frames per second and frame size which can also be changed to provide the required reliability. Again as stated in Section 3.3 estimation of received frame PSNR depends on the encoding method used and decoder specific error concealment process, hence needs further research effort. These improvements are left as future study.

REFERENCES

[1] O. Ayran, O. B. Akan, “Performance of Transport Protocols for Multimedia Communications in Wireless Sensor Networks,” *IEEE Communications Letters*, to appear in 2007.

[2] O. B. Akan, I. F. Akyildiz, “Event-to-Sink Reliable Transport in Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, Vol. 13, pp. 1003-1016, October 2005.

[3] E. Gurses, O. B. Akan, “Multimedia Communication in Wireless Sensor Networks,” *Annals of Telecommunications*, Vol. 60, no. 7-8, pp. 799-827, July-August 2005.

[4] I.F. Akyildiz, W. Su, “Wireless sensor networks: a survey,” *IEEE Communications Magazine*, Vol. 40, pp. 102-114, August 2002.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Transactions on Networking*, Vol. 11, no. 1, February 2003.

[6] C. Wan, S. B. Eisenman, A.T. Campbell, “CODA: Congestion Detection and Avoidance in Sensor Networks,” *Proc. ACM Sensys 2003*, Los Angeles, USA, November 2003

[7] C. Wang, K. Sohraby, B. Li, “SenTCP: A Hop-by-Hop Congestion Control Protocol for Wireless Sensor Networks,” *Proc. IEEE INFOCOM 2005*, Miami, Florida, USA, March 2005.

[8] F. Stan, J. Heidemann, “RMST: Reliable Data Transport in Sensor Networks,” *Proc. IEEE SNPA 2003*, pp. 102-112, May 2003.

[9] C. Y. Wan, A. T. Campbell, L. Krishnamurthy, “PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks,” *Proc. ACM WSNA 2002*, pp. 1-11, Atlanta, GA, USA, September 2002.

[10] S.-J. Park, R. Vedantham, R. Sivakumar, I. F. Akyildiz, “A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks,” *Proc. ACM MobiHoc 2004*, Roppongi, Japan, May 2004.

[11] I.F. Akyildiz, T. Melodia, K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw. (2006)*, doi:10.1016/j.comnet.2006.10.00 .

[12] T. Ozcelebi, F. De Vito, A.M. Tekalp, M.R. Civanlar, M.O. Sunay, J.C. De Martin, "An Analysis of Constant Bitrate and Constant PSNR Video Encoding for Wireless Networks," *Proc. of IEEE International Conference on Communication*, Vol. 11, pp. 5301-5306, June 2006.

[13] B. Zitova, J. B. Zitova, J. Flusser, "Image registration methods: a survey," *Image Vis. Comput.*, pp. 977-1000, 21/11/2003.

[14] S. Tilak, N. B. Abu-Ghazaleh, W. Heinzelman, "Infrastructure Tradeoffs for Sensor Networks," *WSNA'02*, Atlanta, GA, USA, September 2002.

[15] ISO-IEC JTC1, "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s - Video," *ISO/IEC 11172-2*, Geneva, November 1993.

[16] ISO-IEC, "Coding of audio-visual objects - Part 2: Visual," *ISO/IEC 14496-2*, April 1999.

[17] H. Sun, J. W. Zdepskib, W. Kwok, D. Raychaudhurid, "Error concealment algorithms for robust decoding of MPEG compressed video," *Signal Processing: Image Communication*, Vol. 10, pp. 249-268, 1997.

[18] K. Le, C. Clanton, Z. Liu, H. Zheng, "Efficient and Robust Header Compression For Real-Time Services," *Wireless Communications and Networking Conference, IEEE*, Vol. 2, pp. 924-928, 2000.

[19] University of Southern California, "Network Simulator - (ns-2)," *http://www.isi.edu/nsnam/ns*, last accessed July 2007.

[20] Technical University of Berlin, "EvalVid - A Video Quality Evaluation Tool-set," *http://www.tkn.tu-berlin.de/research/evalvid*, last accessed July 2007.

[21] Fabrice Bellard, "FFmpeg - Video and audio converter," *http://ffmpeg.mplayerhq.hu/*, last accessed July 2007.

[22] Berkeley Multimedia Research Center, "The Parallel Berkeley MPEG-1 Encoder," *http://bmrc.berkeley.edu/frame/research/mpeg/*, last accessed July 2007.

[23] The xiph open source community, "Sample Test Videos," *http://media.xiph.org/video/derf/*, last accessed July 2007.