OPTIMIZED ROUTING OF H.264 USING ANT COLONY OPTIMIZATION

PREPARED BY

MANISH KUMAR NUNIA
11MCEC22

GUIDED BY

Prof. Priyanka Sharma
Dept. Of Computer Science and Engg.

Mr. Joginder Singh
Microsemi India Pvt. Ltd.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY
AHMEDABAD-382481

May 2013

OPTIMIZED ROUTING OF H.264 USING ANT COLONY OPTIMIZATION

Major Project Part - II

Submitted in total fulfilment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

MANISH KUMAR NUNIA
11MCEC22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY
AHMEDABAD-382481

May 2013

Declaration

This is to certify that

1. The thesis compromises of my original work towards the degree of Master of Technology in Computer Science & Engineering at Institute of Technology, Nirma University and has not been submitted elsewhere for degree.

2. Due acknowledgement has been made in the text to all other material used.

MANISH KUMAR NUNIA
11MCEC22

Certificate

This is to certify that the Major Project Report entitled **” Optimized Routing of H.264 Using Ant Colony Optimization ”** submitted by **Mr. Manish Kumar Nunia (Roll No: 11MCEC22)** towards the partial fulfilment of the requirements of Master of Technology in the field of Computer Science & Engineering of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date:

Prof.Priyanka Sharma
Guide
Dept. Of Computer Sci. & Engg.
Institute Of Technology
Nirma University
Ahmedabad.

Mr. Joginder Singh
Guide
Software Engineering Department,
Microsemi India Pvt. Ltd.
Hyderabad.

Prof. Vijay Ukani
PG-CSE Section Head,
Dept. Of Computer Sci. & Engg.,
Institute Of Technology
Nirma University
Ahmedabad.

Dr. Sanjay Garg
Head of the Department,
Dept. Of Computer Sci. & Engg.
Institute Of Technology
Nirma University
Ahmedabad.

Dr.Ketan Kotecha
Director,
Institute of Technology
Nirma University
Ahmedabad.

Abstract

Streaming is the method of transmitting media as a continuous stream of data that can be processed by the receiving computer before the entire file has been completely sent. Increasing the quality of the streaming leads to pre-requirement of high bandwidth and a tight control over the QoS parameters. The strict QoS parameter control requires better compression techniques along with efficient routing.For compression H.264 compression is explored and for routing purpose Ant based algorithms are explored. An ant is a simple creature, collectively a colony of ants performs useful tasks such as finding the shortest path to a food source and sharing this information with other ants by depositing pheromone. In the field of ant colony optimization (ACO), models of collective intelligence of ants are transformed into useful optimization techniques that find applications in computer networking. In this research work we presented a multicast routing algorithm MRUA which uses the principle of ant colony to find the optimized route for the communication of multimedia streams.In multimedia communication scenario there are the chances of loss of packets due some reasons. If the critical packets like packets containing I-frames are lost then the whole sequence is corrupted resulting in loss of information. So we also proposed a scheme for prioritization of packets.

Keywords:H.264, Ad-hoc network, Ant Colony Optimization (ACO),AVC/SVC, Video Compression, SP/SI frames, HRD, Prioritize packet streaming, Cross-layer , MPEG-4 , PSNR , Video traffic

Acknowledgements

I take this opportunity to express my immense gratitude to **Prof. Priyanka Sharma** who always motivated me to do something extraordinary. She provided me with a very friendly environment in the institute. Her time particularity and tactics of what and how to learn have helped me a lot to step into the practical world as well as to become a better person. I would like to thank her for providing me all necessary resources, which greatly helped me in completing my project work.

I am grateful to her for his prolonged interest in my work and excellent guidance. She has been a constant source of motivation to me. Her unflinching demand for quality and his insistence, I was able to give my best. She has shown me a way to pursue excellence.

This thesis is the result of support and direction by people at Microsemi India Pvt. Ltd. A special thanks **Mr. Joginder** (Manager, Microsemi India Pvt. Ltd.) and **Miss Lavanya** (Supervisor, Microsemi India Pvt. Ltd.) who were always there when I really needed. Thank you doesnt seem sufficient but it is said with appreciation and respect to both of them for their support, encouragement, care, understanding and precious friendship. I am much indebted to Miss Lavanya for his valuable advice in my work, spending his precious times to read this thesis and gave his valuable suggestions.

It is also a pleasure to mention my good friends for their love, care, support and creating a pleasant atmosphere for me here. I doubt that I will ever be able to convey my appreciation fully, but I owe them my eternal gratitude. A journey is easier when you travel together. Interdependence is certainly more valuable than independence.

I am also thankful to the **Dr. Ketan Kotecha** (Director, Institute of Technology), **Dr. Sanjay Garg** (HOD, CSE Department), **Prof.Vijay Ukani** (Section-Head,PG-CSE) and entire NIRMA family for their co-operation and support. I thank them for providing me such a warm atmosphere to make my stay delightful and memorable.

Last but not least, I would like to pay high regards to my parents for their sincere encouragement and inspiration throughout my research work and lifting me uphill this phase of life. I owe everything to them. Besides this, several people have knowingly and unknowingly helped me in the successful completion of this project.

MANISH KUMAR NUNIA
11MCEC22

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

H.264 is an open, licensed standard that supports the most efficient video compression techniques available today. Without compromising image quality, an H.264 encoder can reduce the size of a digital video file by more than 80% compared with the Motion JPEG format and as much as 50% more than with the MPEG-4 Part 2 standard. This means that much less network bandwidth and storage space are required for a video file. Or seen another way, much higher video quality can be achieved for a given bit rate. Jointly defined by standardization organizations in the telecommunications and IT industries, H.264 is expected to be more widely adopted than previous standards [1].

H.264 has already been introduced in new electronic gadgets such as mobile phones and digital video players, and has gained fast acceptance by end users. Service providers such as on-line video storage and telecommunications companies are also beginning to adopt H.264 [1]. In the video surveillance industry, H.264 will most likely find the quickest traction in applications where there are demands for high frame rates and high resolution, such as in the surveillance of highways, airports and casinos, where the use of 30/25 (NTSC/PAL) frames per second is the norm. This is where the economies of reduced bandwidth and storage needs will deliver the biggest savings.

H.264 is also expected to accelerate the adoption of mega pixel cameras since the highly efficient compression technology can reduce the large file sizes and bit rates generated without compromising image quality. There are trade-off, however. While H.264 provides savings in network bandwidth and storage costs, it will require higher performance network cameras and monitoring stations.

1.2 Thesis Organization

The rest of the report is organized as follow: Chapter-1 provides development history of H.264 standard; a brief talk about the basic working of video compression and the different Video-Services; Chapter-2 contains the literature survey of H.264 standard and different routing protocols for Ad-Hoc networks. Ant Colony Optimization concept and the existing implementation of ACO in field of network Routing is also discussed in this chapter. Once having idea of what is useful and how, we need the tools to implement our idea. Chapter-3 provides an overview of NS2 and EvalVid the tools we used for testing our proposed algorithm. Chapter-4 describe the proposed algorithm, Multicast Routing Using Ant colony (MRUA) and importance of prioritization of streams. The evaluation of the purposed algorithm MRUA and prioritize scheme is an important part of this chapter which provides a comparative study with existing solutions, proving the worthiness of the new proposals. Chapter-6 concludes the work.

1.3 Development of H.264

[18]H.264 is the result of a joint project between the ITU-T's Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (MPEG). ITU-T is the sector that coordinates telecommunications standards on behalf of the International Telecommunication Union. ISO stands for International Organization for Standardization and IEC stands for International Electro technical Commission, which oversees standards for all electrical, electronic and related technologies. H.264 is the name used by ITU-T, while ISO/IEC has named it MPEG-4 Part 10/AVC since it is presented as a new part in its MPEG-4 suite [2] [4]. The MPEG-4 suite includes, for example, MPEG-4 Part 2, which is a standard that has been used by IP-based video encoders and network cameras.

Designed to address several weaknesses in previous video compression standards, H.264 delivers on its goals of supporting [18]:

- Implementations that deliver an average bit rate reduction of 50%, given a fixed video quality compared with any other video standard
- Error robustness so that transmission errors over various networks are tolerated
- Low latency capabilities and better quality for higher latency
- Straightforward syntax specification that simplifies implementations
- Exact match decoding, which defines exactly how numerical calculations are to be made by an encoder and a decoder to avoid errors from accumulating

H.264 also has the flexibility to support a wide variety of applications with very different bit rate requirements. For example, in entertainment video applications which include broadcast, satellite, cable and DVD. H.264 will be able to deliver a performance of between 1 to 10 Mbit/s with high latency, while for telecom services, H.264 can deliver bit rates of below 1 Mbit/s with low latency [3].

1.4 How Video Compression Works

Video compression is about reducing and removing redundant video data so that a digital video file can be effectively sent and stored . The process involves applying an algorithm to the source video to create a compressed file that is ready for transmission or storage. To play the compressed file, an inverse algorithm is applied to produce a video that shows virtually the same content as the original source video. The time it takes to compress, send, decompress and display a file is called latency [1]. The more advanced the compression algorithm, the higher the latency, given the same processing power.

A pair of algorithms that works together is called a video codec (encoder/decoder). Video codecs that implement different standards are normally not compatible with each other; that is, video content that is compressed using one standard cannot be decompressed with a different standard. For instance, an MPEG-4 Part 2 decoder will not work with an H.264 encoder [3]. This is simply because one algorithm cannot correctly decode the output from another algorithm but it is possible to implement many different algorithms in the same software or hardware, which would then enable multiple formats to be compressed.

Different video compression standards utilize different methods of reducing data, and hence, results differ in bit rate, quality and latency. Results from encoders that use the same compression standard may also vary because the designer of an encoder can choose to implement different sets of tools defined by a standard. As long as the output of an encoder conforms to a standards format and decoder, it is possible to make different implementations. This is advantageous because different implementations have different goals and budget. Professional non-real-time software encoders for mastering optical media should have the option of being able to deliver better encoded video than a real-time hardware encoder for video conferencing that is integrated in a hand-held device. A given standard, therefore, cannot guarantee a given bit rate or quality. Furthermore, the performance of a standard cannot be properly compared with other standards, or even other implementations of the same standard, without first defining how it is implemented [2] [6].

A decoder, unlike an encoder, must implement all the required parts of a standard in order to decode a compliant bit stream. This is because a standard specifies exactly how a decompression algorithm should restore every bit of a compressed video. The graph below provides a bit rate comparison, given the same level of image quality, among the following video standards: Motion JPEG, MPEG-4 Part 2 (no motion compensation), MPEG-4 Part 2 (with motion compensation) and H.264 (baseline profile) [7].

1.5 Video Services and Transportation

Video transmission is nowadays widely used in different areas in our daily life, science and industry. The digital transmission of television signal via satellites is very common, and we can also chat with our colleagues face-to-face through video phone in our office or through mobile phone. Video compression is applied to reduce the quantity of data used to represent digital video images. Normally, it exploits spatial and temporal redundancy contained in a sequence of images. Video compression is absolutely required by the video services because of the limited disk space, bandwidth, CPU power, etc. It is being used wherever digital video communications, storage, processing, acquisition and reproduction occur.

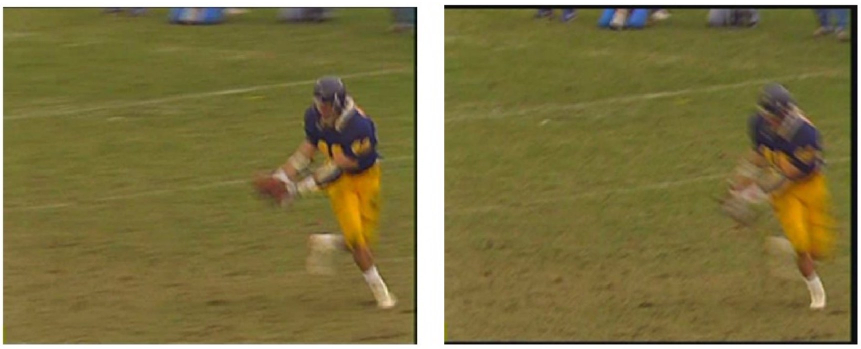
Video compression has two important benefits:

1. It makes it possible to use digital video in transmission and storage environments that would not support uncompressed (raw) video. For example, current Internet throughput rates are insufficient to handle uncompressed video in real time (even at low frame rates and/or small frame size). A Digital Versatile Disk (DVD) can only store a few seconds of raw video at television-quality resolution and frame rate and so DVD-Video storage would not be practical without video and audio compression.

2. Video compression enables more efficient use of transmission and storage resources. If a high bit rate transmission channel is available, then it is a more attractive proposition to send high-resolution compressed video or multiple compressed video channels than to send a single, low-resolution, uncompressed stream.

Even with constant advances in storage and transmission capacity, compression is likely to be an essential component of multimedia services for many years to come [7] The video compression considered here involves the bit rate reduction of a digital video signal carrying visual information. Traditional video-based compression focuses on eliminating the redundant elements of the signal like other information compression techniques.

The redundancy removed can be the one located in temporal, spatial or frequency domains.Figure 1.1 illustrates the spatial and temporal redundancy in two consecutive video frames. In the left side of the image, there is less variation in the grass field. So that area exists significant spatial redundancy. The frame rate of this video sequence is 30 frames/second (NTSC), so the interval between these two frames is 1/30 second. Most



(a) frame #92 (b) frame #93

Figure 1.1: Two consecutive frames from football sequence with large homogeneous region

of these frame remains the same, which means there is also large temporal redundancy [5]. The compression system normally includes an encoder and a decoder. The encoder converts the source data into a compressed form (with a reduced number of bits) prior to transmission or storage and the decoder converts the compressed form back into a representation of the original video data.

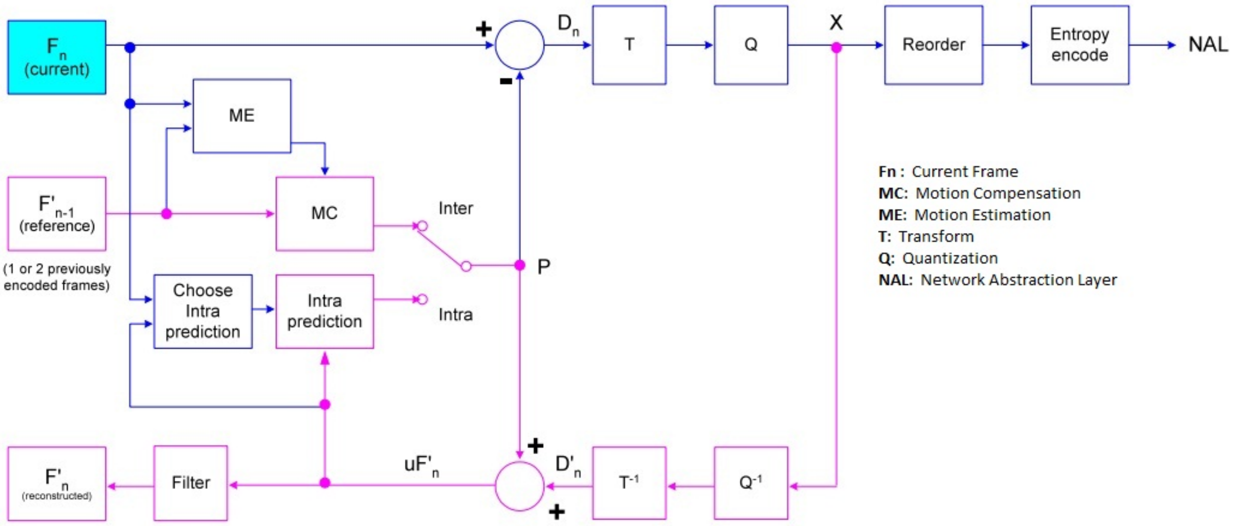


Figure 1.2: H.264 Encoder [7]

The compression system normally includes an encoder and a decoder . The encoder converts the source data into a compressed form (with a reduced number of bits) prior to transmission or storage and the decoder converts the compressed form back into a representation of the original video data. The encoder/decoder pair is often described as a CODEC (enCOder/ DECoder,Figure 1.2 / Figure 1.3)

1.6 Video Coding

Basic Concepts

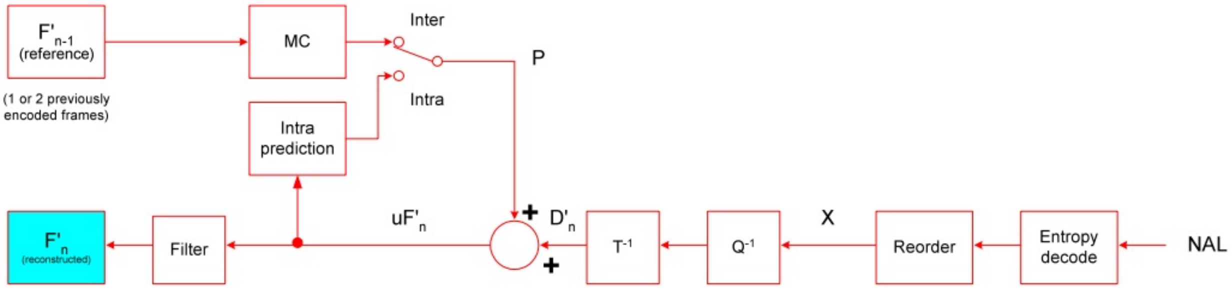


Figure 1.3: H.264 Decoder [7]

1.6.1 Capture

A video sequence is spatially and temporally continuous. To represent a visual scene in digital form, it is necessary to sample the real scene (on a rectangular grid of the image) and temporally (as a series of still frame at regular intervals in time). Each spatio-temporal sample (picture element or pixel)(figure 1.4) is represented as a number or set of numbers that describes the brightness and color of the sample [7].

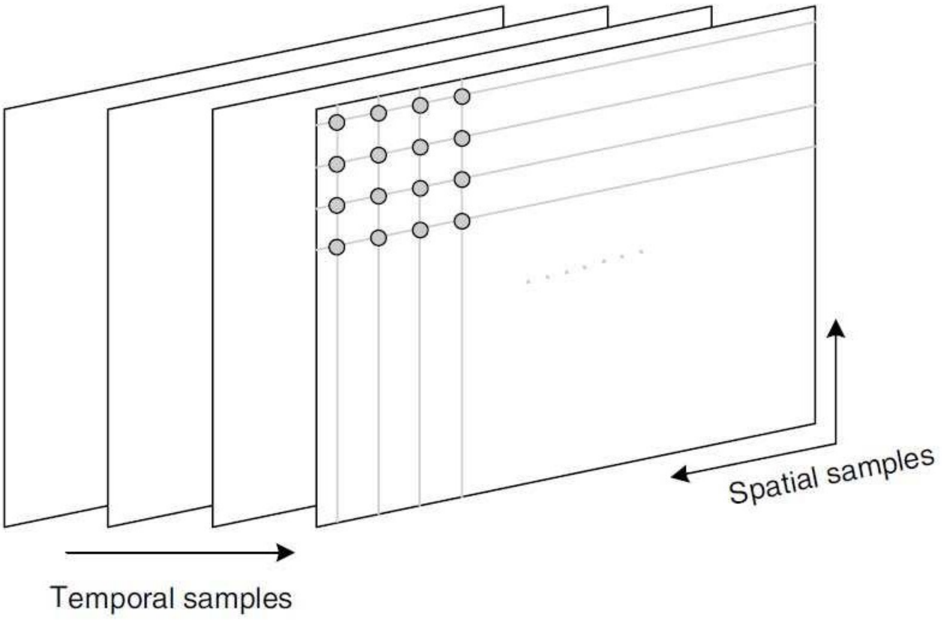
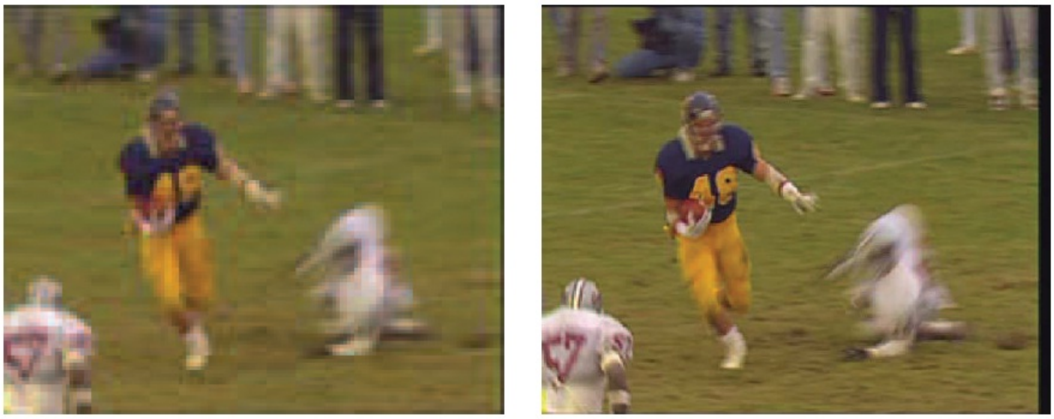


Figure 1.4: Sampling of a video sequence

The output of a CCD array is an analogue video signal in which a varying electrical signal that represents a video image [7]. Sampling the signal at a specific time produces a sampled image or frame that has defined values at a set of sampling points. The common format of a sampled image is a rectangular grid. For spatial sampling, it occurs at each of the intersection points on the grid and the sampled image may be reconstructed by representing each sample as a square picture element. The quality of the image is decided by the number of sampling points. Higher resolution can be achieved by increasing the sampling points (Figure 1.5).

Temporal sampling is related to a moving video. It is like taking a rectangular screen-shot of the signal at periodic time intervals. Then these series of frame are played back to produce the appearance of motion. It is obvious that a higher temporal resolution (sampling rate) can provide smoother motion in the video scene with



(a) Image with low number of sampling points

(b) Image with high number of sampling points

Figure 1.5: Image with different number of sampling points(a.176x144, b.352x288) [7]

the cost of more samples to be captured and stored in the single time unit. The standard for television is 25 or 30 complete frames per second; very low bit-rate video communication maybe below 10 frames per second;and the smooth motion can be up to 60 frames per second.

1.6.2 Color Spaces

The digital video applications rely on the display of color video and so need a mechanism to capture and represent color information. A monochrome image requires just one number to indicate the brightness or luminance of each spatial sample. Color images, on the other hand, require at least three numbers per pixel position to represent color accurately. A color space is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components [7]. The RGB (red-white-blue) is the most well-known color space. A color image sample is represented with three numbers that indicate the relative proportions of Red, Green and Blue (the three additive primary colors of light) in the RGB color space. Any color can be created by combining red, green and blue in varying proportions. The RGB color space is very suitable to capture and display of color images.

However, [7] in the RGB color space the three colors are equally important and so all those three components need to be saved at the same resolution. But it is possible to represent a color image more efficiently by separating the luminance from the color information and representing luma with a higher resolution than color. That is why YCbCr color space is proposed. It is more efficient and used as a part of the color image in video and digital photography systems. Y is the luminance (luma) component and can be calculated as a weighted average of R, G and B:

$$Y = k_r R + k_g G + k_b B$$

where k are weighting factors.

The color information can be represented as color difference (chrominance or chroma) components, where each chrominance component is the difference between R, G or B and the luminance

$$C_r = R - Y$$

$$C_b = B - Y$$

Format	Luminance Resolution
Sub-QCIF	12896
Quarter CIF	176144
CIF	352288
4CIF	704576
HDTV	1280720 or more

Table 1.1: Video Foramts

$$C_g = G - Y$$

The complete description of a color image is given by Y (the luminance component) and two color differences Cb and Cr that represent the difference between the color intensity and the mean luminance of each image sample.

1.6.3 Video Formats

To facilitate the compression and transmission, it is common to capture or convert to a set of defined size. The Common Intermediate Format (CIF) is the basis for a popular set of formats listed in Table 1.1

The choice of frame resolution depends on the application and available storage or transmission capacity. For example, 4CIF is appropriate for standard-definition television and DVD-video; CIF and QCIF are popular for video conferencing applications; QCIF or SQCIF are appropriate for mobile multimedia applications where the display resolution and the bitrate are limited.

1.7 Video Compression Standard

Two organizations dominate the video compression standardization. One is ITU-T VCEG (Video Coding Experts Group) and another is called ISO/IEC MPEG (Moving Picture Experts Group). In the last decades, there have been a lot of video compression standards been proposed. H.120 (ITU, 1994a) is the first digital video encoding standard. It was published by the CCITT (International Telegraph and Telephone Consultative Committee) in 1984, with a revision in 1988 that included contributions proposed by other organizations. As the first digital video encoding standard, H.120 video was not of good enough quality for practical use it had very good spatial resolution, but very poor temporal quality. However, after this standard, the researchers began to be aware that it was necessary to encode using an average of less than 1 bit for each pixel in order to improve the video quality without introducing large bit stream [6] [9]. To achieve this, it is required to group a set of pixels coded together. This led to the following block-based codecs.

H.261 (ITU, 1994b) is regarded as the first practical video encoding standard. It is ratified in November 1988 and originally designed for transmission over ISDN (Integrated Services Digital Network) lines on which data rates are multiples of 64 kbit/s. Two different frame sizes are supported: CIF (352288 luminance with 176144 chrominance) and QCIF (176144 with 8872 chrominance) using a 4:2:0 sampling scheme. The coding algorithm was designed to be able to operate at different video bit rates (between 40 kbit/s and 2 Mbit/s). It also has a backward compatible trick for sending still picture graphics with 704576 luma resolution and 352288 chroma resolution (which was added in a later revision in 1993). A lot of products support this standard. In fact, it has great influence that most of the subsequent international video coding standards are based closely on the H.261 design [3] [9]. Some typical structures which are still dominating today includes:

- 1616 macroblock motion compensation,

- 88 DCT,
- scalar quantization,
- zig-zag scan,
- run-length,
- variable-length coding.

[7]The Moving Picture Experts Group (MPEG) was established in 1988 to address the need for standard video and audio formats, and build on H.261 to get better quality by using more advanced coding methods. Nowadays, MPEG-1 has become the most widely used lossy audio/video format in the world. Its technology is used in huge number of products. Part 2 of the MPEG-1 standard is about the video and is defined in ISO/IEC-11172-2 (ISO, 1993). The design was heavily influenced by H.261. However, the MPEG-1 still has some weakness. For example, no standard support for interlaced video with poor compression when used for interlaced video, and only one standardized profile" (Constrained Parameters Bit stream) which was not suitable for higher resolution video. In addition, it only supports one color space, 4:2:0 sub-sampling pattern.

[7] [8]MPEG-2, also known as H.262 (ITU, 2000), is the successor of MPEG-1. Part 1 and part 2 of MPEG-2 were developed in a joint collaborative team with ITU-T. It is now widely used as the format of digital television signals [7]. It is also popular as the movie format and other programs that are distributed on DVD and similar discs. Part 2 of MPEG-2, shares the main feature with the previous MPEG-1 standard. But it also provides support for interlaced video which is used by analog broadcast TV systems. MPEG-2 video is not optimized for low bit-rates, especially less than 1 Mbit/s at standard definition resolutions. All standards-compliant MPEG-2 Video decoders are fully capable of playing back MPEG-1 Video streams conforming to the Constrained Parameters Bitstream syntax. It is also used in some HDTV transmission systems.

H.263 (ITU, 2005) was developed as an improvement based on H.261, MPEG-1 and MPEG-2. It is originally designed as a low-bitrate compressed format for video conferencing. Its first version was completed in 1995 and provided a suitable replacement for H.261 at all bit rates. The version 2 of H.263 (also known as H.263+) has the entire technical content of the original version of the standard, but enhanced H.263 capabilities by adding several annexes which can substantially improve encoding efficiency and provide other capabilities [8] [10].

MPEG 4 - part 2 (or MPEG - Visual)(ISO, 2004c) belongs to the MPEG-4 ISO/IEC standards. It is a discrete cosine transform compression standard, similar to previous standards such as MPEG-1 and MPEG-2 [8] [10]. It has approximately 21 profiles, including profiles called Simple, Advanced Simple,Main, Core, Advanced Coding Efficiency, etc. The most commonly deployed profiles are Advanced Simple and Simple, which is a subset of Advanced Simple.

Chapter 2

Literature Survey

2.1 H.264/AVC and H.264/SVC

[6]H.264/AVC (ITU, 2003) as the latest video coding standard of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC MPEG, its main goal is to enhance compression performance and provide a "network-friendly" video representation addressing "conversational" (video telephony)and "non-conversational" (storage, broadcast or streaming) applications.

The new standard is designed for technical solutions in different application areas:

- Serial storage on optical and magnetic devices, Blueray, DVD, etc.
- Conversational services over Internet and Intranet.
- Video-On-Demand (VOD) and multimedia streaming services over Internet and Intranet.
- Broadcast over cable, satellite, cable modem, DSL, terrestrial, etc.
- Multimedia Messaging Services (MMS) over ISDN, DSL, Ethernet, LAN, wireless and mobile networks, etc.
- High-Definition Television (HDTV).

[6]The new applications may be also deployed over future networks. To make the standard be able to handle different applications and networks, the H.264/AVC design covers a Video Coding Layer (VCL), which is designed to efficiently represent the video content, and a Network Abstraction Layer (NAL), which formats the VCL representation of the video and provides header information in a manner appropriate for conveyance by a variety of transport layers or storage media (Figure 2.1)

2.1.1 H.264/AVC

Network Abstraction Layer

The NAL is designed in order to provide "network friendliness" to enable simple and effective customization of the use of the VCL for a broad variety of systems. The NAL facilitates the ability to map H.264/AVC VCL data to transport layers such as [8] [17].

- RTP/IP for any kind of real-time wire-line and wireless Internet services.

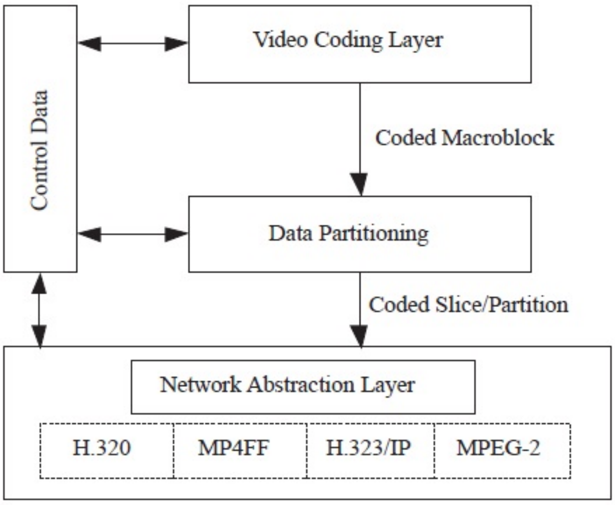


Figure 2.1: Structure of H.264/AVC video encoder [6] [8]

- File formats, such as ISO MP4.
- H.32X for wired and wireless conversational services.
- MPEG-2 systems for broadcasting services.

[50]One of the most important concepts for NAL is NAL unit. So the coded video data is organized into NAL units, each of which is effectively a packet that contains an integer number of bytes. The first byte of each NAL unit is a header byte that contains an indication of the type of data in the NAL unit, and the remaining bytes contain payload data of the type indicated by the header. The payload data in the NAL unit is interleaved as necessary with emulation prevention bytes,which are bytes inserted with a specific value to prevent a particular pattern of data called a start code prefix from being accidentally generated inside the payload. The NAL unit structure definition specifies a generic format for use in both packet-oriented and bitstream-oriented transport systems, and a series of NAL units generated by an encoder is referred to as a NAL unit stream.

Video Coding Layer

[10] [12] Video Coding Layer As in all prior ITU-T and ISO/IEC JTC1 video standards since H.261, the VCL design follows the so-called block-based hybrid video coding approach. Each coded picture is represented in block shaped units of associated luma and chroma samples called macro blocks. The basic source-coding algorithm is a hybrid of inter-picture prediction to exploit temporal statistical dependencies and transform coding of the prediction residual to exploit spatial statistical dependencies. There is no single coding element in the VCL that provides the majority of the significant improvement in compression efficiency in relation to prior video coding standards. It is rather a plurality of smaller improvements that add up to the significant gain.

2.1.2 H.264/SVC

In [8] H.264/SVC is an extension of the H.264/AVC which is standardized by the Joint Video Team of the ITU-T VCEG and the ISO/IEC MPEG. It is a highly attractive solution to the problems set by the characteristics of modern video transmission systems. The term scalability" refers to the removal of parts of the video bit stream in order to adapt it to the various needs or preferences of end users as well as to varying terminal capabilities or network conditions. SVC enables the transmission and decoding of partial bit streams to provide video services with lower temporal or spatial resolutions or reduced fidelity while retaining a reconstruction quality that is high relative to the rate of the partial bit streams. Hence, SVC provides functionalities such as graceful degradation in lossy transmission environments as well as bit rate, format, and power adaptation. These functionalities provide enhancements to transmission and storage applications.

[8] The usual modes of scalability are temporal, spatial, and quality scalability. Spatial scalability and temporal scalability describe cases in which subsets of the bit stream represent the source content with a reduced picture size (spatial resolution) or frame rate (temporal resolution), respectively. With quality scalability, the sub stream provides the same spatio-temporal resolution as the complete bit stream, but with a lower fidelity where fidelity is often informally referred to as signal-to-noise ratio (SNR). Quality scalability is also commonly referred to as fidelity or SNR scalability.

As an extension of H.264/AVC, the SVC tries to provide following essential requirements.

- Similar coding efficiency compared to single-layer coding for each subset of the scalable bit stream.
- Little increase in decoding complexity compared to single layer decoding that scales with the decoded spatio-temporal resolution and bit rate.
- Support of temporal, spatial, and quality scalability.
- Support of a backward compatible base layer (H.264/AVC in this case).
- Support of simple bit stream adaptations after encoding.

[8] As H.264/AVC, SVC also includes VCL and NAL layer. But there are also very important differences, especially in the VCL:

- The possibility to employ hierarchical prediction structures for providing temporal scalability with several layers while improving the coding efficiency and increasing the effectiveness of quality and spatial scalable coding.
- New methods for inter-layer prediction of motion and residual improving the coding efficiency of spatial scalable and quality scalable coding.
- The concept of key pictures for efficiently controlling the drift for packet-based quality scalable coding with hierarchical prediction structures.
- Single motion compensation loop decoding for spatial and quality scalable coding providing a decoder complexity close to that of single-layer coding.
- The support of a modified decoding process that allows a lossless and low-complexity rewriting of a quality scalable bit stream into a bit stream that conforms to a non-scalable H.264/AVC profile.

2.1.3 H.264:Summary

Video is an important service provided by networks. Beginning from 1980s, there have been a lot of coding standard proposed. The latest standard is H.264/AVC, which is a block-oriented motion compensation-based codec standard developed by the ITU-T VCEG together with the ISO/IEC MPEG. SVC is an extension of the H.264/AVC video compression standard, which aims to enable the encoding of a high-quality video bitstream that contains one or more subset bitstreams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using the existing H.264/MPEG-4 AVC design with the same quantity of data as in the subset bitstream [8] [3] [15].

Other than the video codec, another important issue is the transmission interface for networks. So the video transmission over networks is also introduced briefly in this chapter. We are specially interested in the transport interface for H.264/SVC video [8]. The file format of H.264/SVC is compatible with H.264/AVC. The RTP payload format for H.264/SVC allows for packetization of one or more Network Abstraction Layer Units (NALUs) produced by an H.264 video encoder [3] [15].

2.1.4 Comparative Study of Different Standards

A Comparison of different compression techniques [8] [10] [11]:

Feature/Standard	MPEG-1	MPEG-2	MPEG-4 part 2 (visual)	H.264/MPEG-4 part 10
Macroblock size	16x16	16x16 (frame mode) 16x8 (field mode)	16x16	16x16
Block Size	8x8	8x8	16x16, 16x8, 8x8	16x16, 16x8, 8x16, 8x8, 4x8, 8x4, 4x4
Transform	8x8 DCT	8x8 DCT	8x8 DCT/Wavelet	4x4, 8x8 Int DCT, 4x4,2x2 hadamard
Entropy coding	VLC	VLC	VLC	VLC, CAVLC, CABAC
Motion Estimation & Compensation	YES	YES	YES	Yes, more flexible Up to 16 MVs per MB
Playback & Random Access	YES	YES	YES	YES
Quantization	Scalar Quantization with step size of constant increment	Scalar Quantization with step size of constant increment	Vector Quantization	Scalar Quantization with step size of increase at the rate of 12.5
Profiles	NO	5	8	3
Reference Picture	one	one	one	multiple
Bidirectional Prediction Mode	forward/backward	forward/backward	forward/backward	forward/forward, forward/backward, backward/backward
Picture Types	I, B, P, D	I, B, P	I, B, P	I, B, P, SP, SI
Error Robustness	Synchronization & concealment	Data partitioning, FEC for important packet transmission	Synchronization, Data partitioning, Header extension, Reversible VLCs	Data partitioning, Parameter setting, Flexible macroblock ordering, Redundant slice, Switched slice
Transmission Rate	Up to 1.5 Mbps	2-15 Mbps	64 Kbps- 2 Mbps	64kbps -150Mbps
Compatibility with previous standards	N/A	YES	YES	NO
Encoder complexity	LOW	MEDIUM	MEDIUM	HIGH

Table 2.1: Comparison of H.264 and other formats

2.2 Routing Protocol for ad hoc networks

[11] Routing Protocols are used to discover and maintain routes between the source and destination nodes. For MANET, there are two main kinds of routing protocol: on-demand protocols (also called reactive protocols) and table-based protocols (also called proactive protocols). For reactive protocols, nodes only compute routes when they are needed. Usually, caches are used to reduce the effort of route discovery. For proactive protocols, each node maintains a routing table containing routes to all nodes in the network. Nodes must periodically exchange messages with routing information to keep routing tables up-to-date.

[11] Furthermore, some hybrid protocols are proposed. This is because both proactive and reactive routing have specific advantages and disadvantages that make them suitable for certain kinds of scenarios. The hybrid methods try to take the advantages of those two and achieve better performance.

In the rest part of this chapter, some typical reactive and proactive protocols are firstly introduced. Then the protocols that try to fulfill the demands of QoS or security requirements are presented.

2.2.1 Factors of consideration routing protocol design

In [11] the main factors which plays a critical role while designing a routing protocol are discussed:

1. *Effective Routing*: This is the foremost requirement of the protocol to successful discover and deliver the packet from the source to the destination. Some of the measures for effective routing include Packet Delivery Ratio, percentage of Optimal Routes taken and average end-to-end delay.

2. *Congestion Avoidance*: Strongly coupled with the previous parameter, this is to ensure that the routing protocol does not congest a particular route/node thereby leading to packet drops or even failure of the nodes.

3. *Energy Consumption*: Most of the mobile nodes are laptops, pdas and other portable devices which have a strong requirement for energy consumption. Hence the routing algorithm must minimize the energy consumption of the individual nodes.

4. *Load Balancing*:Some of the nodes may be strategically located resulting in being present in most of the optimal routes of communication. Such nodes get unfairly overloaded leading to network congestion. Hence there is a need for balancing the loads on the individual nodes, though compromising on the optimal routes, thereby resulting on fair load distribution.

5. *Reachability*: Reachability is the ability of the routing algorithm to find at least one path between source/destination pair.

2.2.2 Taxonomy of routing protocols

In [11]Tarique et.al. and in [12] Shauatul Islam et.al.

1. *Single path and multipath routing* : Routing protocols may maintain single or multiple routes to a given destination. Single path protocols can discover one or multiple routes and then always select the best path for data transport, discarding the other paths. On the other hand, multipath routing refers to the protocols that discover, maintain, and use multiple paths to transport the sensed data. Multipath routing protocols can help in extending the network lifetime because they favour battery depletion of different nodes at a comparable rate. In the case of so called alternate path protocols, the information about

multiple paths is maintained in the routing table but is used only as a backup in case the primary path fails.

2. *Reactive, proactive, and hybrid routing*: In reactive/on-demand protocols, paths are searched and set up only when required. In proactive protocols, routing information for all known destinations is maintained up-to-date all the time, irrespective of whether a destination is being selected or not for data transmission. Some protocols use a combination of both techniques and hence are called hybrid protocols. Usually, a proactive approach is quite energy expensive, such that it should be adopted only if the application justify its use.

3. *Source and next hop routing*: In next hop routing, a data packet only contain the information about its final destination. At each node, the routing protocol decides the next hop using the information stored in the local routing table. In the case of source routing, the source node encapsulates all the path information in the packet datagram, such that intermediate nodes only need to read the next hop information from the datagram and forward it accordingly. Some hybrid protocols make use of both techniques. For instance, the ant packets normally used in ACO approaches, retrace a discovered path making use of source routing, while data packets are always routed according to a next hop scheme. From the one hand, the use of source routing can be very effective to reduce per packet processing requirements and to avoid loops. From the other hand, it can limit the scalability of a protocol and can incur into problems in highly dynamic networks, especially when used to route data packets rather than control packets.

4. *Flat and hierarchical routing*: Flat routing protocols view the entire network as a set of nodes located on the same hierarchical level. Their job is to find a route between any arbitrary pair of nodes. Hierarchical protocols, on the other hand, divide the network into regions called zones/clusters. The nodes within a cluster only need to deliver their data to the cluster head (CH). In turn, the cluster head can be part of a further level, according to some hierarchical arrangement of the nodes rooted at the final sink nodes.

5. *Data-centric and address-centric routing*: Data-centric protocols do not require globally unique node IDs, while address centric protocols do. Data-centric routing is commonly used when assigning a unique ID to each node is either not feasible or appropriate given the purpose and/or the size of the network. Data-centric routing, which is also indicated as content-based routing, is the common way of operating in sensor networks, global grid infrastructures, and publish/subscribe and event-notification schemes for peer-to-peer/ overlay networks. In data-centric routing, data packets are named using high level descriptors and queries are generated for the named data. The nodes that have the requested data only respond to these queries.

6. *Distributed and centralized routing*: In centralized routing models, discovery and maintenance of routing information is controlled by a single node known as sink/base station. In the distributed approach, each node gathers/builds routing information on its own. The distributed routing model is more robust to network variations and is therefore more appropriate for dynamic and ad hoc networks. The centralized model presents a single point of failure and might be unable to follow timely the changes due to network dynamics. On the other hand, the powerful processing capabilities of the controller node can be additionally exploited to effectively execute a variety of useful processing tasks.

7. *Best-effort and QoS-aware routing*: Protocols that do not provide any guarantees in terms of quality of the service delivered to the application are categorized as best-effort. Protocols that can provide to the application routing services with quality guarantees (e.g., in terms of end-to-end delay, delay jitter, available bandwidth, packet losses, etc.) are indicated as QoS-aware.

8. *Event-driven and query-based routing*: This classification is based on the nature of the applications the routing protocol is serving for. In event-driven protocols, data routing starts after the detection of an event from a sensor node. For instance, an event might be triggered when the value of a monitored variable (e.g., the temperature) exceeds a certain threshold value, or after the expiration of a timer used for the periodical report from the sensors. In the case of query-based protocols, data are sent from the sensor nodes to the monitor node in response to a specific query. Some protocols may support both types of applications.

9. *Energy-aware routing*: Routing protocols that prioritize routes on the basis of an energy metric (e.g., the residual energy of the nodes on the route) are classified as energy aware. Since nodes in sensor networks have limited non-rechargeable batteries, it is customary to make an efficient utilization of the available energy if the network has to stay operational for a long time (this might not be the case for uses of the WSN that involve the acquisition of very precise information over short time periods).

10. *Loop free*: If the paths used by data packets are guaranteed to have no cycles, the protocol is termed as loop-free. In order to provide this guarantee, the protocol must include explicit mechanisms to check and avoid the possible occurrence of loops. In addition to data packets, also control packets can incur in loops (e.g., during path discovery). Looping of data packets can have a strong negative impact on network performance: it reduces data throughput and/or increases packet delay, wasting at the same time bandwidth and energy resources. Looping of control packets might be less critical, but it should still be avoided for the same reasons. Loop generation is an issue for all next hop routing protocols.

11. *Fault-tolerance*: Wireless sensor networks are dynamic in nature. From the one hand, nodes can fail due to hostile environment or battery outage. On the other hand, control packets can get lost due to interference or memory/processing problems. A routing protocol which is robust to topological changes and to packet losses is termed as fault-tolerant. Multipath routing is often used as a way to provide fault-tolerance.

12. *Load balancing*: Load balancing refers to the mechanism in which data packets are spread in a balanced way across multiple paths from sources to destinations. A balanced traffic distribution can help to optimize network throughput and can allow all nodes to deplete their batteries at a similar rate. The use of multipath routing is a natural way to implement load balancing. However, data spreading across multiple paths must be done by minimizing path interference, since a high rate of radio collisions would nullify the positive effects derived from the use of multiple paths.

2.2.3 Reactive Routing Protocols

Because reactive routing only tries to find a route when necessary, it is believed that it is more scalable to dynamic, large networks. When a node needs a route to another node, it initiates a route discovery process to find a route. Generally, it consists of the following two main phases:

Route discovery

It is the process of finding a route between two nodes, whether directly reachable within wireless transmission range or reachable through one or more intermediate network hops through other hosts.

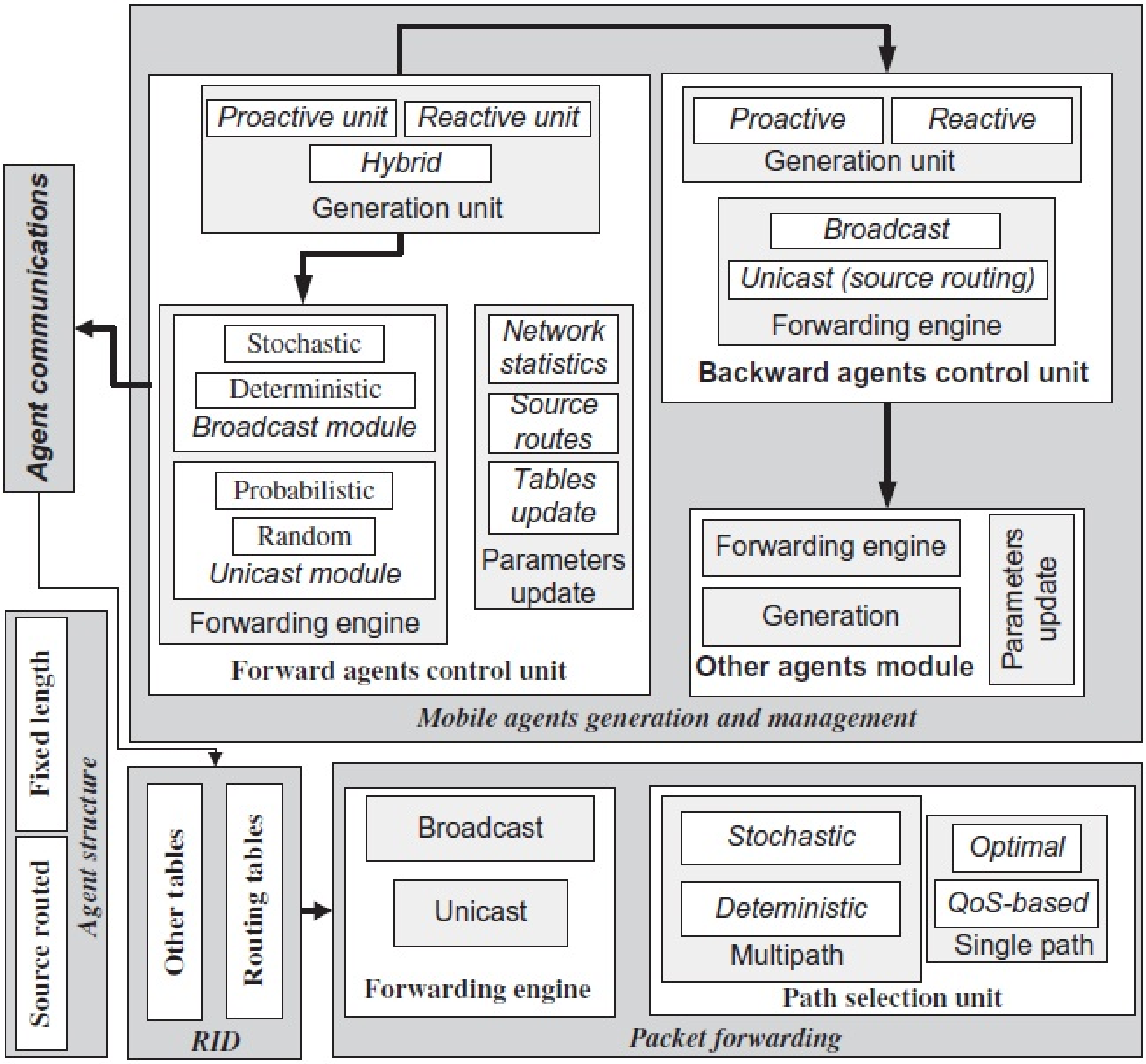


Figure 2.2: Diagram of the common routing framework for SI routing protocols. [10]

Route maintenance

It is the process of repairing a broken route or finding a new route in the presence of a route failure. Route maintenance procedure monitors the operation of the route and informs the sender of any routing errors.

2.2.4 Proactive Routing Protocols

In Proactive Routing, also called table-driven Routing, routes are calculated before one is needed. The protocol tries to keep routing information to all nodes every time up-to-date. The update of the tables can be periodically or driven by events.

2.2.5 Reactive Routing vs. Proactive Routing

At the moment, there are two protocols mainly discussed on the IETF standard track:

- [51]Dynamic MANET On-demand (DYMO) routing is successor to AODV. So it shares the reactive feature of its ancestor. Compared to AODV, DYMO has TLVs to allow for extensibility and allows a route to be improved by changing in response to superior routing information.

- OLSRv2, as already introduced, is the successor to OLSRv1.

[15] They represents the reactive and proactive protocols respectively in the standardization of MANET routing protocols. The performance evaluation in shows that the traffic load, the mobile node mobility and the network density all have impact on the performance of the routing protocol. The proactive protocol offers better performances for CBR (Constant Bit Rate) sources given that it guaranties lowest delay and jitter. But it consumes more bandwidth. And when the mobility is low, the reactive protocol performances low delay and overhead.

2.2.6 Multipath Routing Protocols

[10] Based on uni-path (reactive and proactive)protocols, a lot of multi-path routing protocols are proposed. These protocols consist of finding multiple routes between a source and destination node by exploiting the density of the network. These multiple paths between source and destination node pairs can be used to compensate for the dynamic and unpredictable nature of ad hoc networks.

The multi-path routing could offer several benefits: load balancing, fault-tolerance, higher aggregate bandwidth, lower end-to-end delay, etc.

2.3 Ant Colony Optimization

[16] [3] Swarm Intelligence (SI) is an artificial intelligence technique based around on the study of collective behavior in decentralized, self-organized systems. Ant Colony Optimization is popular among other Swarm Intelligent Techniques. Ants-based routing algorithms have attracted the attention of researchers because they are more robust, reliable, and scalable than other conventional routing algorithms. Since they do not involve extra message exchanges to maintain paths when network topology changes, they are suitable for mobile ad-hoc networks where nodes move dynamically and topology changes frequently.

[19] Ant colony optimization (ACO) is a meta-heuristic for solving hard combinatorial optimization problems inspired by the indirect communication of real ants. In ACO algorithms, (artificial) ants construct candidate solutions to the problem being tackled, making decisions that are stochastically biased by numerical information based on (artificial) pheromone trails and available heuristic information. The pheromone trails are updated during algorithm execution to bias the ants search toward promising decisions previously found. [19] [3] Despite being one of the youngest meta-heuristics, the number of applications of ACO algorithms is very large. In principle, ACO can be applied to any combinatorial optimization problem for which some iterative solution construction mechanism can be conceived. Most applications of ACO deal with *NP*-hard combinatorial optimization problems, that is, with problems for which no polynomial time algorithms are known. ACO algorithms have also been extended to handle problems with multiple objectives, stochastic data, and dynamically changing problem information. There are extensions of the ACO meta-heuristic for dealing with problems with continuous decision variables, as well.

ACO was primarily intended for solving combinatorial optimization problems, among which *NP*-hard problems are the most challenging ones. In fact, no polynomial-time algorithms are known for such problems, and therefore heuristic techniques such as ACO are often used for generating high-quality solutions in reasonable computation times, as shown in Figure 2.3

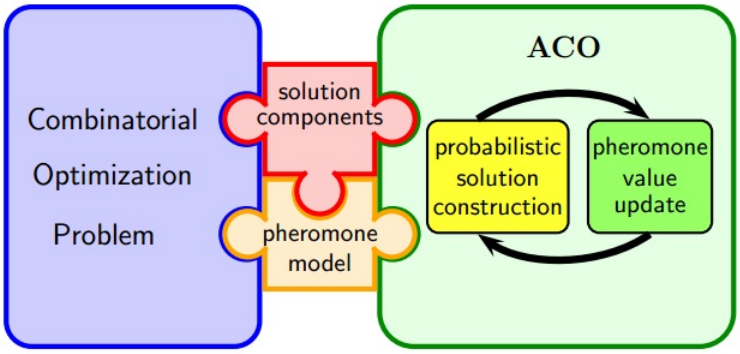


Figure 2.3: The ACO Metahurestic [3]

2.3.1 Advantages of Nature Inspired Routing Algorithms

[16] Nature Inspired Routing Algorithms have advantages as follows:

1. *Adaptability*: Adaptability is the ability of the routing algorithm to quickly and accurately adapt to a variety of network circumstances. Nature-inspired routing protocols are adaptive to changes in the network environment because of the stochastic decision making used by the multiple agents working in parallel.

2. *Robustness*: Robustness is defined as the ability to perform correctly in the face of unusual or unforeseen situations. Nature inspired routing algorithms are robust with respect to individual agent errors, agent losses, communication errors, or other system failures. This is due to the redundancy among the multiple agents in the system.

3. *Scalability*: Scalability is the ability of a routing protocol to perform efficiently as one or more inherent parameters of the network grow to be large in value. Typical parameters that are studied for ad-hoc networks are the number of nodes and the average rate of mobility in m/s under various mobility models.

2.4 Routing and ACO

[14] Many researchers have developed proactive and reactive algorithms for MANETs. DSR and AODV are reactive algorithms establishing paths only when they are needed. When a node has some data to send to another node, it searches for a path by flooding the network with control messages. Their dissemination introduces some delay before data packets can be sent and reactive routing algorithms are inefficient when there is much continuous but intermittent traffic in the network. DSDV on the other hand, is a typical proactive routing algorithm. They prepare paths to all destination nodes beforehand and maintain them by exchanging control messages periodically. They require a network to carry a lot of control traffic into a network.

In this chapter we presented an comparative analysis of different ant colony based routing algorithms for MANET's.

2.4.1 Routing in MANET

[13] Routing in MANET is a Dynamic Optimization Problem as the search space changes over time. The routing policy is defined as the rule that specifies what node to take next at each decision node to reach the destination node. Due to the time varying nature of the topology of the networks, traditional routing techniques

such as distance-vector and link-state algorithms that are used in fixed networks, cannot be directly applied to mobile ad hoc networks. The constraints of MANETs demand the need of specialized routing algorithms that can work in a decentralized and self-organizing way. The routing protocol of a MANET must dynamically adapt to the variations in the network topology.

[13] The routing scheme in a MANET can be classified into two major categories Proactive and Reactive(discussed in Section 2.2). The proactive or table driven routing protocols maintain routes between all node pairs all the time. It uses periodic broadcast advertisements to keep routing table up-to-date. This approach suffers from problems like increased overhead, reduced scalability and lack of flexibility to respond to dynamic changes. The reactive or on-demand approach is event driven and the routing information is exchanged only when the demand arises. The route discovery is initiated by the source. Hybrid approaches combines the features of both the approaches. Destination Sequence Distance Vector (DSDV) is a flat proactive routing protocol whereas Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector (AODV) routing are examples of flat reactive or on demand protocols. The increased latency may render the reactive protocols unsuitable for time- critical traffic.

Important Elements of Ant Colony Based Routing Algorithms [16]

- *Stigmergy* : Stigmergy is a form of indirect communication used by social insects like ants, bees etc., through environment. This is the most important element of the ACO meta-heuristic.
- *Evaporation* :In ACO algorithms the value of the Pheromone in all links is decreased by a factor p, such that:

  $$t_{ij} = t_{ij}(1 - p)$$

  This helps in reducing the influence of past experience during decision making.
- *Aging*: The amount of pheromone that an ant lays on path decreases with its age, an older ant lays less pheromone than a younger one. Since ant mostly assumes symmetric links, in which cost of links in both directions is same. The solution for asymmetric links is that ants measure the cost during the forward trip and deposit pheromone on backward trip. Suppose a data network, with N nodes, being S a generic source node if it generates an agent (or ant) toward a destination D. Two types of ants are defined:
  - Forward Ant, which will travel from a source node S to a destination D.
  - Backward Ant will be generated by a forward ant at the destination D. It will return to S through path used by forward ant, to update routing tables of the visited nodes, according to the information before collected by forward ant.

2.5 Ant Colony Based Routing Algorithms

2.5.1 Ant Based Control (ABC) Routing

In this Algorithm, every node in the network has a pheromone table entry for every possible destination in the network, and each table has an entry for every neighbour. Initially all are assumed to have 0.5 probabilities.

[19] Ants are launched from any node in the network. Each node has random destination. Ants move from node to node, selecting the next node to move to according to the probabilities in the pheromone tables for their destination node. Arriving at a node, they update the probabilities of that node's pheromone table entries corresponding to their *source* node. They alter the table to increase the probability pointing to their previous

node. When ants have reached their destination, they die. The increase in these probabilities is a decreasing function of the age of the ant, and of the original probability. The ants get delayed on parts of the system that are heavily used. Some noise can be added to avoid freezing of pheromone trails.

The method used to update the probabilities is quite simple: when an ant arrives at a node, the entry in the pheromone table corresponding to the node from which the ant has just come is increased according to the formula:

$$P = (P_{old} + \Delta p) \, / \, (1 + \Delta p)$$

Here p is the new probability and delta p is the probability increase.The probabilities are updated according to the following formula, where age stands for the number of time steps that passed since the launch of the ant:

$$\Delta P = ((0.08/age) + 0.005)$$

The delay in time steps that is given to the ant is a function of the spare capacity s of the node:

$$delay = \lfloor 8 * e^{-0.075s} \rfloor$$

2.5.2 Ant Colony Based Routing (ARA)Algorithm

[19] [14] Ant Colony Based Routing Algorithm (ARA) works in an on demand way, with ants setting up multiple paths between source and destination at the start of a data session FANTs are broadcast by the sender to all its neighbours. Each FANT has a unique sequence number to avoid duplicates. A node receiving a FANT for the first time creates a record [**destination address, next hop, pheromone value**] in its routing table.

The node interprets the source address of the FANT as destination address, the address of the previous node as next hop, and computes the pheromone value depending on the number of hops the FANT needed to reach the node. Then the node relays the FANT to its neighbours When the FANT reaches destination, it is processed in a special way. The destination node extracts the information and then destroys the FANT. A BANT is created and sent towards the source node. In that way, the path is established and data packets can be sent.Data packets are used to maintain the path, so no overhead is introduced. Pheromone values are changing.

2.5.3 Probabilistic Emergent Routing (PERA) Algorithm

[14] This algorithm works in an on-demand way, with ants being broadcast towards the destination at the start of a data session. Multiple paths are set up, but only the one with the highest pheromone value is used by data and the other paths are available for backup. The route discovery and maintenance is done by flooding the network with ants. Both forward and backward ants are used to fill the routing tables with probabilities. These probabilities reflect the likelihood that a neighbour will forward a packet to the given destination. Multiple paths between source and destination are created. First of all, neighbours are discovered using HELLO messages, but entries are only inserted in the routing table after receiving a backward ant from the destination node. Each neighbour receives an equi-probable value for destination. This value is increased as a backward ant comes from that node, establishing a path towards destination. As ants are flooded, the algorithm uses sequence numbers to avoid duplicate packets. Only the greater sequence number from the same previous hop is taken into account. Forward ants with a lower sequence number are dropped. This approach is similar to AODV Route Request packets, but discovers a set of routes instead of one. Data packets can be routed according to the highest probability in the routing table for the next hop.

2.5.4 Ant Agents for Hybrid Multipath Routing (AntHocNet)

[14] AntHocNet is a multipath routing algorithm for mobile ad-hoc networks that combines both proactive and reactive components. It maintains routes only for the open data sessions. This is done in a Reactive Route Setup phase, where reactive forward ants are sent by the source node to find multiple paths towards the destination node. Backward ants are used to actually setup the route. While the data session is open, paths are monitored, maintained and improved proactively using different agents, called proactive forward ants.

2.5.5 Antnet: Ant Algorithm

[14] In Antnet algorithm, at regular intervals t from every network node s, a forward ant fsd is launched toward a destination d to discover a feasible, low-cost path to that node and to investigate the load status of the network along the path. If fsd is a measure (in bits or in number of packets) of the data flow s to d, then the probability of creating at node s a forward ant with node d as destination is:

$$P_{sd} = f_{sd} / \sum_{i=1}^{N} f_i$$

While traveling toward their destination nodes, the forward ants keep memory of their paths and of the traffic conditions found. The identifier of every visited node i and the time elapsed since the launching time to arrive at this i-th node are stored in a memory stack.

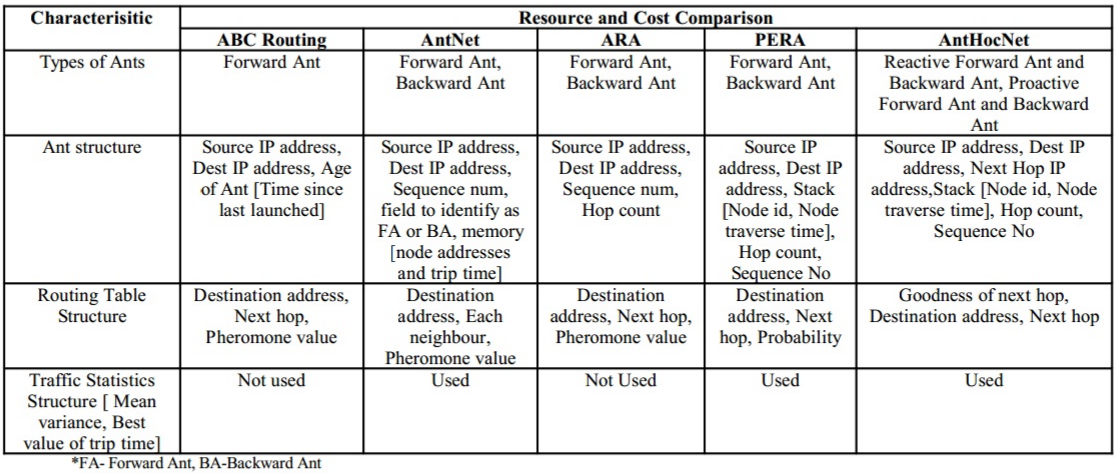
2.5.6 Comparison Of Different Ant Based Routing

[12] [13]ABC Routing was developed for wired telecommunication networks and it assumes symmetric path costs between nodes. Like in AntNet, each node s periodically sends out ants to randomly chosen destinations. Each ant has an associated age, which is increased proportionally to the load of each visited node. While traveling from its source s to its destination d, the ant updates the pheromone for the path backward to s, based on its age. This is an important difference with AntNet: ants update pheromone about the path to their source while going forward, and no backward ants are used. This is possible because of the assumption of symmetric path costs. Another difference is that no path statistics are used to evaluate path quality measurements reported by the ants, and that no local heuristic is used to help guide the ants (in AntNet, the local queue lengths are used). Finally, in ABC, it is not data packets that are routed according to the pheromone, but call setup messages. Moreover, these messages do not follow pheromone probabilistically, but greedily choose the directions with the highest pheromone level. Once a call has been set up successfully, data packets follow its circuit.

[14] AntNet was developed for packet switched wired networks. HELLO messages are used initially to discover the neighbours In PERA, HELLO messages are broadcast each time any node moves to a different position so that node can discover its new neighbours In ARA entry in the routing table for each node is created when a forward ant arrives at that node. Pheromone value is the number of hops required by the forward ant to reach the current node rom the destination.ARA is quite similar to PERA. One difference is that both forward and backward ants leave pheromone behind: forward ants update pheromone about the path to the source, while backward ants update pheromone about the path to the destination. Another difference is that also data packets update pheromone, so that paths which are in use are also reinforced while the data session is going on. This comes down to repeated path sampling, so that ARA keeps more of the original ACO characteristics than PERA. PERA uses routing table which has the following structure: **[Destination, Next hop, Probability]**. Initially probability value each node from source to destination is initialized with uniform

probabilities (1/N) where N is the number of neighbours for each node. It is used for wireless networks. It works similar to Antnet. Each node periodically sends forward ant to randomly chosen destination, whereas source node is chosen according to some probability of data flow.

[14] Anthocnet requires more number of resources as compared to other ant based algorithms. This is because there are two forward ants [Proactive and Reactive] and two backward ants [Proactive and Reactive]. Structure of ant is similar but number of ants generated varies with other ant based algorithms. Amount of control traffic generated due to ants is more than other ant based algorithms. Anthocnet is most efficient in maintaining paths. It has greater chance of exploring new paths due to proactive nature with a hint of probability. This is due to the fact that proactive ants are normally unicast to sample the existing path found by reactive forward ants but also have a small probability at each node of being broadcast. Therefore even though Anthocnet maintains paths between nodes and explores new routes it is costly and requires more resources. A brief summary of comparison of different algorithms is shown in Figure 2.4



Characterisitic	Resource and Cost Comparison				
	ABC Routing	AntNet	ARA	PERA	AntHocNet
Types of Ants	Forward Ant	Forward Ant, Backward Ant	Forward Ant, Backward Ant	Forward Ant, Backward Ant	Reactive Forward Ant and Backward Ant, Proactive Forward Ant and Backward Ant
Ant structure	Source IP address, Dest IP address, Age of Ant [Time since last launched]	Source IP address, Dest IP address, Sequence num, field to identify as FA or BA, memory [node addresses and trip time]	Source IP address, Dest IP address, Sequence num, Hop count	Source IP address, Dest IP address, Stack [Node id, Node traverse time], Hop count, Sequence No	Source IP address, Dest IP address, Next Hop IP address,Stack [Node id, Node traverse time], Hop count, Sequence No
Routing Table Structure	Destination address, Next hop, Pheromone value	Destination address, Each neighbour, Pheromone value	Destination address, Next hop, Pheromone value	Destination address, Next hop, Probability	Goodness of next hop, Destination address, Next hop
Traffic Statistics Structure [Mean variance, Best value of trip time]	Not used	Used	Not Used	Used	Used

*FA- Forward Ant, BA-Backward Ant

Figure 2.4: Comparison table [12] [13]

2.6 Summary

Among wireless networks Anthocnet is more efficient among all the considered ant based algorithms because it has greater chance of exploring new paths based on probability but it is more costly and requires more resources for implementing it. This is due to the fact that there is lot of ant traffic generated. PERA is better in terms of less cost and also efficient in maintaining and exploring new paths. ARA is similar to PERA but in ARA both forward and backward ants update pheromone value. For wired networks Antnet works best in maintaining the established paths as compared to ABC routing because ABC uses greedy approach and Antnet chooses best paths based on probability.

Chapter 3

Software Tools

3.1 EvalVid

It is a complete framework and tool-set for evaluation of the quality of video transmitted over a real or simulated communication network [20] [21]. It was integrated into NS2 as described in Integration section. We used it for end to end delay, end-to-end jitter and PSNR evaluations in NS2-based simulations. EvalVid evaluation framework:

The main components of the evaluation framework are described as follows [20]:

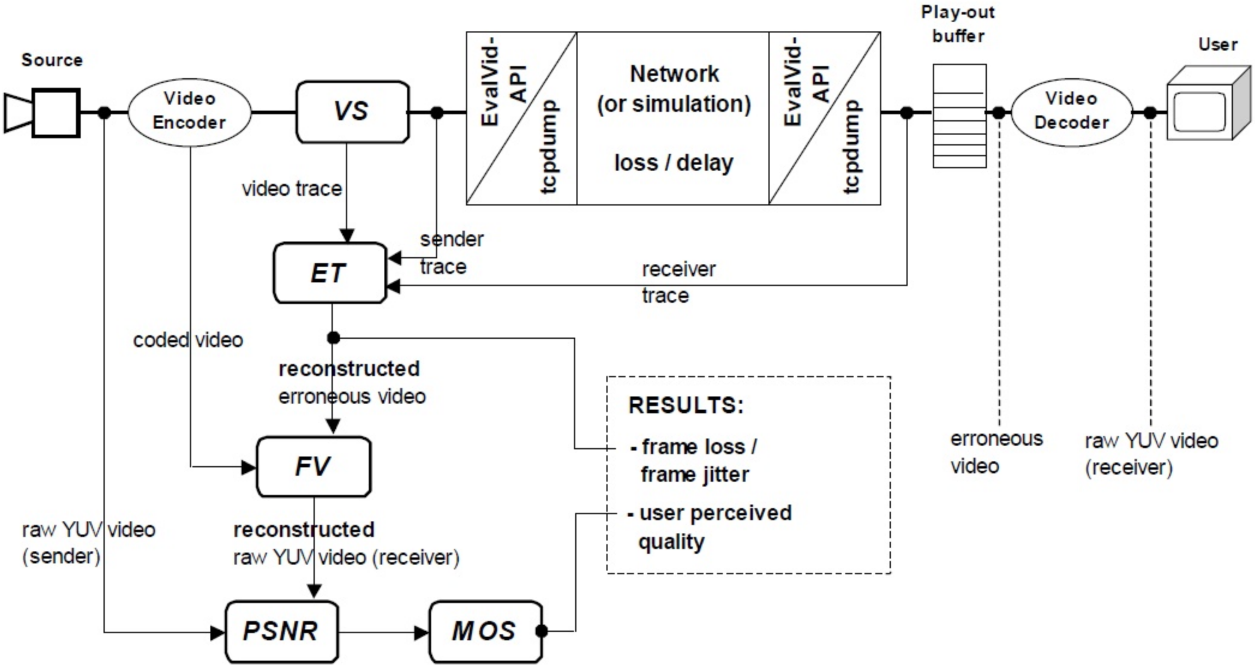


Figure 3.1: EvalVid evaluation framework

Source: The video source can be either in the YUV QCIF (176 x 144) or in the YUV CIF (352 x 288) formats.

Video Encoder and Video Decoder: Currently, EvalVid supports two MPEG4 codecs, namely the NCTU codec and ffmpeg.

VS (Video Sender): The VS component reads the compressed video file from the output of the video encoder, fragments each large video frame into smaller segments, and then transmits these segments via UDP packets over a real or simulated network. For each transmitted UDP packet, the framework records the timestamps, the packet id, and the packet payload size in the sender trace file with the aid of third-party tools, such as tcp-dump or win-dump, if the network is a real link. Nevertheless, if the network is simulated, the sender trace file is provided by the sender entity of the simulation. The VS component also generates a video trace file that contains information about every frame in the real video file. The video trace file and the sender trace file are later used for subsequent video quality evaluation.

ET (Evaluate Trace): Once the video transmission is over, the evaluation task begins. The evaluation takes place at the sender side. Therefore, the information about the timestamps, the packet id, and the packet payload size available at the receiver has to be transported back to the sender. Based on the original encoded video file, the video trace file, the sender trace file, and the receiver trace file, the ET component creates a frame/packet loss and frame/packet jitter report and generates a reconstructed video file, which corresponds to the possibly corrupted video found at the receiver side as it would be reproduced to an end user. In principle, the generation of the possibly corrupted video can be regarded as a process of copying the original video trace file frame by frame, omitting frames indicated as lost or corrupted at the receiver side. Nevertheless, the generation of the possibly corrupted video is trickier than this and the process is further explained in more details later. Furthermore, the current version of the ET component implements the cumulative inter-frame jitter algorithm for play-out buffer. If a frame arrives later than its defined playback time, the frame is counted as a lost frame. This is an optional function. The size of the play-out buffer must also be set, otherwise it is assumed to be of infinite size.

FV (Fix Video): Digital video quality assessment is performed frame by frame. Therefore, the total number of video frames at the receiver side, including the erroneous ones, must be the same as that of the original video at the sender side. If the codec cannot handle missing frames, the FV component is used to tackle this problem by inserting the last successfully decoded frame in the place of each lost frame as an error concealment technique.

PSNR (Peak Signal Noise Ratio): PSNR is one of the most widespread objective metrics to assess the application-level QoS of video transmissions. The following equation shows the definition of the PSNR between the luminance component Y of source image S and destination image D:

$$PSNR(n)_{db} = 20\log_{10}\left(\frac{V_{peak}}{\sqrt{\frac{1}{N_{col}N_{row}}\sum_{i=0}^{N_{col}}\sum_{j=0}^{N_{row}}\left[Y_S(n,i,j)-Y_D(n,i,j)\right]^2}}\right)$$

$V_{peak} = 2^k - 1$

$k = number\ of\ bits\ per\ pixel(luminance\ component)$ A set of information is required for calculating different QoS parameters:

Sender side information:

- raw uncompressed video
- encoded video

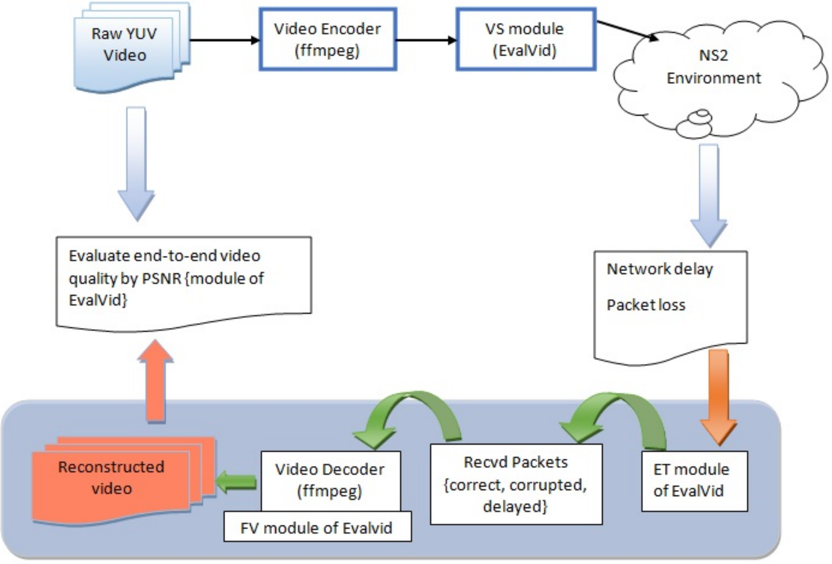


Figure 3.2: QoS Assessment Framework including EvalVid

- time-stamp and type of packet sent

Receiver side information:

- time stamp and type of packet received
- reassembled encoded video possibly erroneous
- raw uncompressed video to be displayed

For the tools within EvalVid only the trace files, the original video file and the decoder are needed. Therefore, we can say that for EvalVid the network is just a 'black box' which generates delay, loss and possible packet reordering. It can be a real link or a simulation or emulation of a network. Due to the use of trace files, the network box can be easily replaced, which makes EvalVid very flexible. The QoS assessment framework of the EvalVid is depicted by the figure 3.2 The other architectural details regarding the working of EvalVid can be found in literature.

3.2 NS2

NS is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of many protocols, routing, and multicast protocols over wired and wireless (local and satellite) networks. All of my simulations were conducted under this framework. For testing a demo network, the network shown in figure 3.4 is used:

There are three connecting simulation agents, namely MyTrafficTrace, MyUDP, and MyUDPSink, are implemented between NS2 and EvalVid. These interfaces are designed either to read the video trace file or to generate the data required to evaluate the video delivered quality.

*MyTrafficTrace:*The MyTrafficTrace agent is employed to extract the frame type and the frame size of the video trace file generated from the output of the VS component of EvalVid. Furthermore, this agent fragments

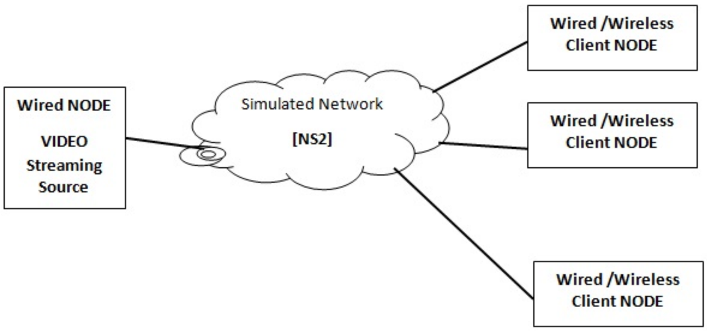

Wired /Wireless Client NODE
Wired NODE
VIDEO Streaming Source
Simulated Network
[NS2]
Wired /Wireless Client NODE
Wired /Wireless Client NODE

Figure 3.3: NS2 Environment depicts the video server and the clients

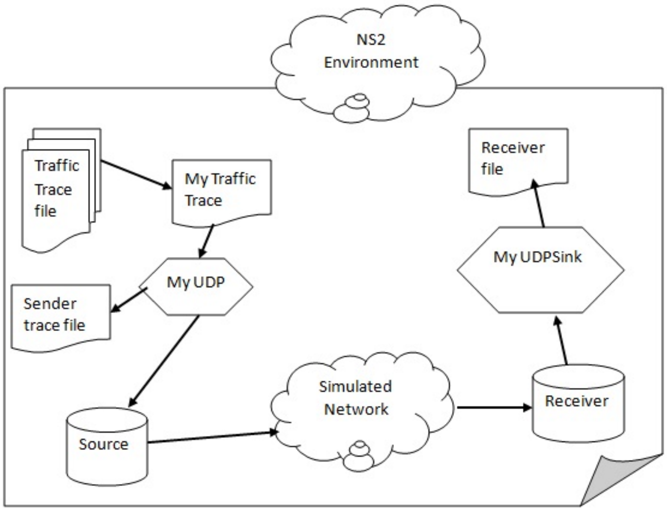


Figure 3.4: NS2 Environment

the video frames into smaller segments and sends these segments to the lower UDP layer at the appropriate time according to the user settings specified in the simulation script file.

*MyUDP:*MyUDP is an extension of the UDP agent. This new agent allows users to specify the output file name of the sender trace file and it records the time-stamp of each transmitted packet, the packet id, and the packet payload size. The task of the MyUDP agent corresponds to the task that tools such as tcp-dump or win-dump do in a real network environment.

*MyUDPSink:*MyUDPSink is the receiving agent for the fragmented video frame packets sent by MyUDP. This agent also records the time-stamp, packet ID, and payload size of each received packet in the user specified file.

Chapter 4

Proposed Solution

4.1 Algorithm:Introduction

To support QoS, the essential problem is to find a route with sufficient available resources to meet the QoS constraints and possibly to incorporate optimizations, such as finding the lowest cost or most stable of the routes that meet the QoS constraints. Given these goals, the following are the basic design considerations for a QoS-aware routing protocol [14]:

→ Resource estimation:
To offer a resource-guaranteed route, the key concept is to obtain information about the available resources from lower layers. This information helps in performing call admission and QoS adaptation.

→ Route discovery:
There are two main approaches to routing MANETs: reactive routing and proactive routing. Reactive routing reduces overhead at the expense of delay in finding a suitable route; whereas, the reverse is true for proactive routing. For QoS-aware routing, another issue is determining the combination of reduced latency and reduced overhead that is best for supporting QoS.

→ Resource reservation:
As previously stated, the bandwidth resources are shared by neighbouring hosts in MANETs. Therefore, another challenging issue is how to allocate these shared resources, the type of resource reservation scheme, and the kind of call admission that should be used for setting up and maintaining QoS-aware routes.

→ Route maintenance:
The mobility of nodes in MANETs causes frequent topology changes in the network, making it difficult to meet the QoS constraints. Incorporating a fast route maintenance scheme into QoS-aware routing is the fourth design consideration.

→ Route selection:
QoS-aware routing has more stringent requirements on route stability, because frequent route failures adversely affect the end-to-end QoS.

The following are the core features which characterizes ACO instances for routing problems [14]:

→ provide traffic-adaptive and multipath routing

→ rely on both passive and active information monitoring and gathering

→ make use of stochastic components

→ do not allow local estimates to have global impact

→ set up paths in a less selfish way than in pure shortest path schemes favouring load balancing

→ show limited sensitivity to parameter settings.

4.1.1 Draft:

After going through the currently available literature of ACO based Multimedia routing algorithms. I hereby by proposes a draft of algorithm which is based on Multicast Routing using Ant Colony Algorithm.

1. **Key Terms/Parameters**
   We considered network model as a directed graph **G(V,E)**; where V: Set of Nodes and E: Set of Edges

   → $R^+ \ : real\ positive\ number$

   → $(i,j)\ \in\ E\ \ : link\ from\ node\ i\ to\ node\ j;\ (i,j)\ \in\ V$

   → $s\ \in\ V\ \ : source\ node\ of\ multicast\ group$

   → $c_{ij}\ \in\ R^+\ \ : capacity\ of\ link$

   → $d_{ij}\ \in\ R^+\ \ : delay\ of\ link(i,j)$

   → $N_p\ \in\ V-s\ \ : set\ of\ destinations\ of\ multicast\ group$

   → $B\ \in\ R^+\ \ : traffic\ demand\ in\ bps$

   → $T(s,N_p)\ \ : multicast\ tree\ with\ source\ 's'\ and\ set\ of\ destination\ 'N_p'$

   → $P_T(s,n)\ \subseteq\ T(s,N_p)\ \ : path\ connecting\ 's'\ and\ a\ set\ of\ destination\ n\ \in\ N_p$

   → $d(P_T(s,n))\ \ : delay\ of\ path\ P_T(s,n) = \displaystyle\sum_{i,j\ \in\ P_T(s,n)} d_{ij}$

   → $t_{ij}\ \ : current\ traffic\ of\ link\ (i,j);\ t_{ij}\ \in\ R^+$

   → $z_{ij}\ \ : cost\ per\ bps\ of\ link\ (i,j);\ z_{ij}\ \in\ R^+$

2. **Objective Functions:**
   I considered four function:
   $f_1(T):\ Cost\ of\ tree$
   $f_2(T):\ Maximum\ end\ to\ end\ delay$
   $f_3(T):\ Average\ delay$
   $f_4(T):\ Maximum\ link\ utilization$

   Function definitions:

   → $f_1(T) =\ B * \sum_{(i,j)\ \in\ T} c_{ij}$

$\rightarrow\ f_2(T) =\ Max\{d(P_T(s,n))\};\ n \in\ N_p$

$\rightarrow\ f_3(T) =\ \frac{1}{|N_p|} * \sum_{n\ \in\ N_p} d(P_T(s,n))$

$\rightarrow\ f_4(T) =\ Max\ \left\{\frac{B+t_{ij}}{c_{ij}}\right\};\ (i,j)\ \in\ T$

With satisfying following **constraint**:
$(B + t_{ij})\ \leq\ c_{ij}\ ;\ \forall\ (i,j)\ \in\ T(s, N_p)$
So,
$X =\ T(s, N_p)\ Y =\ [f_1(T)\ \ f_2(T)\ \ f_3(T)\ \ f_4(T)]$

3. **Algorithm:For selecting a node randomly:**
/* Node must be unvisited node */

```
Select randomly a node 'q' /* $q, q_0 \in (0,1]$ */;
if $q > q_0$ then
    choose node 'j' with larger $P_{ij}$;
else
    randomly choose 'j' using probability $P_{ij}$;
end
```

Algorithm 1: Random node selection

4. **Algorithm: Probability function definition:**

$$P_{ij} = \left\{ \begin{array}{cc} \frac{\tau_{ij}^{\alpha}\left([\eta_d]^a\ [\eta_c]^c\ [\eta_t]^t\right)^{\beta}}{\sum\limits_{\forall g \in N_i} \tau_{ig}^{\alpha}\left([\eta_d]^a\ \ [\eta_c]^c\ \ [\eta_t]^t\right)^{\beta}}; & if\ j \in N_i \\ 0\ ; & otherwise \end{array} \right\} \text{ Where:}$$

a,b,c: Algorithm Parameters
$\alpha, \beta:\ Relative\ importance\ of\ the\ pheromone.$
$\eta_d:\ \frac{1}{d_{ij}}$
$\eta_c:\ \frac{1}{c_{ij}}$
$\eta_t:\ \frac{1}{tij}$
$initially\ \tau_{ij} =\ \tau_0;\ \tau_0 > 0$

5. **Algorithm:Pheromone updation:**

```
forall the $T \in Y_{known}$ do
    forall the $(i,j) \in T$ do
        $\tau_{ij} = (1-\rho) * \tau_{ij} + \rho * \Delta\tau$;
    end
end
```

Algorithm 2: Pheromone Update

Here;

$Y_{known} = \; partial\ solution$

$$\Delta\tau = \frac{1}{\sum\limits_{\forall T \in Y_{known}} (f_1(T) + f_2(T) + f_3(T) + f_4(T))}$$

$\rho \;\in\; (0, 1]$

A range for the values of τ is defined $[\tau_{min}, \tau_{max}]$

$\tau_{min} = \frac{\Delta\tau}{2*w*(1-\rho)}$; $w = \; number\ of\ ants\ in\ generation$

$\tau_{max} = \frac{\Delta\tau}{1-\rho}$

Following condition are checked for each new value:

```
if $\tau_{ij} < \tau_{min}$ then
    $\tau_{ij} = \tau_{min} \quad \forall i, j \in \; E$
end
if $\tau_{ij} > \tau_{max}$ then
    $\tau_{ij} = \tau_{max} \quad \forall i, j \in \; E$
end
```

Algorithm 3: Checking τ

6. **Algorithm:Tree Generation**

```
Initialize $\alpha, \beta, a, b, c, B, (s, N_p), t_{ij}$
$Set \quad T = \phi \quad /*Tree*/$
$Set \quad D_r = \phi \quad /*set\ of\ destination's\ reached*/$
$Set \quad R = s \quad /*list\ of\ starting\ nodes*/$
while $R \neq \phi \quad OR \quad D_r \neq \; N_p$ do
    Select a node i from R and generate set $N_i$
    if $N_i = \phi$ then
        R=R-i; /* erase non-feasible neighbour node */
    else
        $Set\ probability\ P_{ij}\ to\ each\ node\ of\ N_i$
        $Select\ node\ j\ of\ N_i\ using\ algorithm$1
        $T = T \cup (i, j)$
        $R = R \cup \; j$
    end
    if $j \in \; N_p$ then
        $D_r = D_r \cup \; j \quad /* \; node\ j\ is\ a\ destination*/$
    end
    $Update\ \tau_{ij}$ using algorithm 2
end
$Prune\ Tree\ T \quad /* \; remove\ unused\ edges \; */$
return T
```

Algorithm 4: Tree Generation

7. **Algorithm:Pseudo Code for MRUA**

/* Multicast Routing Using Ant colony */

```
$Initialize\ algorithm\ parameters$
$Set\ \tau_{ij} = \tau_{max} \quad / * high\ exploration * /$
while $Terminating\ condition\ not\ satisfied$ do
    $T^{'} =\ Build\ Tree : Call\ Algorithm$ 4
    $\tau_{L_{ij}} = \frac{Q}{L_m}$
    $L_m : length\ travelled\ by\ ant(function\ of\ cost) =\ c_j - c_i$
    $Local\ pheromone\ update\ using\ \tau_{L_{ij}}$
    if $T^{'}\ is\ better\ then\ T_{known} \in Y_{known}$ then
        $Y_{known} =\ Y_{known}\ \cup\ T^{'} - \left\{\ T_{known} \in\ Y_{known}\ \mid\ \forall\ T^{'}\ >\ T_{known}\right\}$
    end
    Global Pheromone Update using Algorithm 2
end
return $Y_{known}$
```

Algorithm 5: Pseudo code of MRUA

4.2 Algorithm Work Flow: Pseudo-code

```
BEGIN
Routing Table SET-UP: For each node $k$ the routing tables are initialized
with a uniform distribution:
$P_{ij} = \tau_0 \ \forall \ i \ \in \ N_k$
while true do
    STEP-1: In regular time interval, each node $s$ launches an $F_{s \rightarrow d}$
    ant to a randomly chosen destination $d$.
    /* when $F_{s \rightarrow d}$ reach a node $k$ $(k \neq d)$, it performs STEP-2 */
    while for each $F_{s \rightarrow d}$ till jumping node $\neq d$ do
        STEP-2: $F_{s \rightarrow d}$ pushes in its stack $S_{s \rightarrow d}$(k) the node $k$
        identifier and the time between its launching from s to its arriving to k.
        $F_{s \rightarrow d}$selects the next node to visit in two possible ways:
        (a) It draws between i nodes, $i \in N_k$, where each node $i$ has a $P_{di}$
        probability (in the k routing table) to be selected.
        if the node selected in (a) was already visited then
            (b) It draws again the jumping node,
            but now with the same probability for all neighbour $i$,$i \in N_k$
            if the selected node was already visited then
                STEP-3: A cycle is found. $F_{s \rightarrow d}$ pops from its stack
                all data of the cycle nodes.
                The optimal path must not have any cycle.
                $F_{s \rightarrow d}$ returns to 2 (a) if the time spent in the cycle
                is minor than its half trip time;
                else it dies, for to avoid infinite loops.
            end
        end
    end
    STEP-4:$F_{s \rightarrow d}$ generates another ant, called backward ant $B_{d \rightarrow s}$
    $F_{s \rightarrow d}$ transfers to $B_{d \rightarrow s}$
    its stack $S_{s \rightarrow d}$ and then dies.
    /*$B_{d \rightarrow s}$, will return to $s$,following the same path used by $B_{d \rightarrow s}$*/
    while for each $B_{d \rightarrow s}$ till $k \neq s$ do
        /*When $B_{d \rightarrow s}$ arrives from a node $f$, $f \in N_k$ to a node $k$,
        it performs the STEP-5*/
        STEP-5: $B_{d \rightarrow s}$ updates the k routing table and list of trips,
        for the entries regarding to nodes $k'$ between k and d inclusive,
        according to the data carried in $S_{s \rightarrow d}$ $(k')$,
        increasing probabilities associated to path used and
        decrementing other paths probabilities.
        if $k \neq s$ then
            $B_{d \rightarrow s}$ will leave $k$ and jump to a node given by $S_{s \rightarrow d}$ (k-1)
        end
    end
end
END
```

4.3 Performance Analysis

4.3.1 Multicast Routing Using ACO

MRUA is an hybrid algorithm because it makes use of both reactive and proactive techniques to establish routing paths.

It is reactive in the sense that a node only starts gathering routing information for a specific destination when a local traffic session needs to communicate with the destination and no routing information is available. It is proactive because as long as the communication starts, and for the entire duration of the communication, the nodes proactively keep the routing information related to the ongoing flow up to date with network changes. In this way both the costs and the number of paths used by each running flow can reflect the actual status of the network, providing an optimized network response.

The reactive component of the algorithm deals with the phase of path setup and is totally based on the use of ACO ant agents to find a good initial path. Routing information is encoded in node pheromone tables. The proactive component implements path maintenance and improvement, proactively adapting during the course of a session the paths the session is using to network changes. Path maintenance and improvement is realized by a combination of ant path sampling and slow-rate pheromone diffusion: the routing information obtained via ant path sampling is spread between the nodes of the MANET and used to update the routing tables according to a bootstrapping scheme that in turn provide main guidance for the ant path exploration. Link failures are dealt with using a local path repair process or via explicit notification messages. Stochastic decisions are used both for ant exploration and to distribute data packets over multiple paths.

4.3.2 Structure

→ Network as a Graph G(N,L); N: nodes, L:duplex links

→ Each node has 1+n buffers:

- Incoming Message buffer (one)
- Outgoing Message buffer (two:low priority queue, high priority queue) per outgoing link (n)

→ Two type of network packets :Data, Overhead

- Overhead packets or Agents: Forward Ant(low priority) & Backward Ant (high priority)

→ Queues are FIFO in nature

4.3.3 Simulation Parameters

4.3.4 Experimental Setup

For the evaluation of the performance of MRUA, we use simulation experiments. This is the most commonly used approach in MANETs, since the complexity of MANET networks makes analytical evaluations difficult and limited in scope, while the high costs of purchasing and configuring hardware limit the use of real test-beds. All the simulation experiments reported in this work last 500 seconds, and each data point represents the average taken over 10 runs using different random seeds. The experiments are carried out in open space scenarios. Different number of node varying from 100-500 nodes move in an area of 2000 x 2000. The movements of the nodes or stations are defined according to the random way point mobility model. A Station selects a random destination point in the region, and move to that point with a randomly chosen speed. Upon reaching there,

Parameter	Value
Network Size	100,200,300,400,500
Area	2000m x 2000m
Transmission Range	250m
Link Bandwidth	2 Mbps
MAC	DCF
Packet Size	512 byte
Control Packet Size	64 byte
Queue Length	300
Mobility Model	Random Waypoint(Pt:30 msec)
Node Speed	0 - 20 msec
Simulation Time	500 secs

they stay for a fixed time defined by pause time, after which a new random location and speed are chosen. In our experiments, the node or station speed is chosen uniformly between 0 and 20 ms, unless stated otherwise. The pause time for the nodes is always 30 seconds. Radio signal propagation is modeled with the two-ray ground reflection model, which take care of both the direct and the ground reflection path. The transmission range of each station is 250 meters. At the physical and medium access control layers of the network protocol stack, we used the well known IEEE 802.11b protocol in DCF function with 2 Mbits/s bandwidth. At the application layer, data traffic is simulated by 20 constant bit rate (CBR) sources, sending packets of 512 bytes. The CBR sessions starts at a random time between 0 and 180 seconds after the start of the simulation, and go on until the end of the simulation. The data rate is 4 packets per second, unless specified. CBR uses the UDP protocol at the transport layer. All these configuration values reflect choices widely adopted in MANET research. Pertaining to the MRUA parameters, if not stated differently, the value of in β is set to 20 and the maximum number of entries in the pheromone diffusion messages is set to 10, and the sending interval for the proactive ants is 2 seconds.

To analyze the performance of the routing algorithms, we measure the ratio of correctly delivered versus sent packets (delivery ratio) and the average end-to-end delay for data packets. These are standard measurement parameters of effectiveness in MANETs. Other metrics which we judged are delay jitter and routing overhead. Delay jitter is defined as the average difference in inter-arrival time between packets. This is an important metric in QoS networks and in MANETs provide a measure of the stability of the algorithm's response to topological modifications. Routing overhead is defined as a measure of efficiency. We calculate it as the number of control packet transmissions per data packet delivered (counting every hop).

Pheromone definition metrics

A pheromone value encodes the adaptive expected goodness of a routing decision. Different path assessment metrics can be used to measure this goodness. We look into the use of path length in number of hops (referred to as "hops" in 4.1), path end-to-end delay ("delay"), number of hops combined with the quality of each hop in terms of signal-to-noise ratio ("snr"), and number of hops combined with end-to-end delay ("hops+delay"). We also plot results for a version of MRUA where proactive learning is switched off("no proactivity") and the "snr" metric is used. Figures 4.1 and 4.2 show respectively the average end-to-end delay and the delivery ratio of the different versions of the algorithm.

From the above results it is clear that to choose a good path assessment metric is to use in the pheromone model, based on knowledge of the unambiguous network environment. A metric mixture is the correct choice.

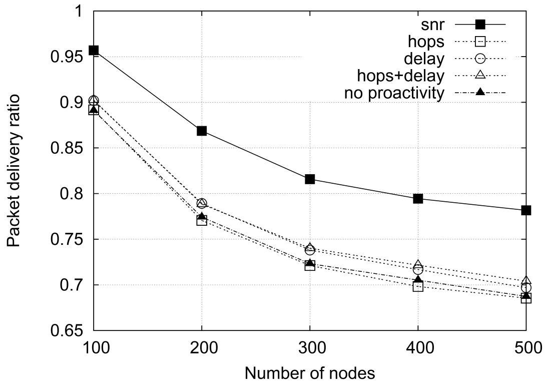


Figure 4.1: Delivery Ratio- pheromone definition metrics

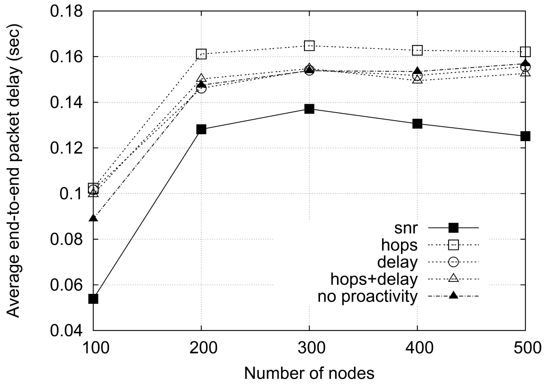


Figure 4.2: End-To-End to delay- pheromone definition metrics

Varying proactive ant send rate

The proactive ant send period is the time between improvement phase and successive proactive ants in the proactive path maintenance. It defines how frequently the algorithm looks for path improvements, and therefore how promptly it can adapt to changes. We made experiments with send intervals of 0.5, 1, 2, 5, 10, 20, and 50 seconds. The different curves in Figures 4.3 and 4.4 represent experiments using different maximum node speeds (so varying the network mobility).

From the above results it is clear that too low ant send interval causes bad performance, because the network gets flooded by ants. At 2 seconds, there seems to be a most favorable send interval. For frequencies lower than that, the performance degrades because the algorithm is not sending sufficient number of ants to keep up with the changes in the network. For low speeds, this fall in performance is slower since the network changes less fast. However, it is attention-grabbing that the best send interval value is free of the node speed. We also did some simulation keeping the speed constant on 10 m/s, and changing the data traffic load. Even though higher traffic load could be expected to leave less space for ants, also there the best ant send interval was always approximately 2 seconds.

Varying number of entries in pheromone diffusion messages

The upper limit of entries in the pheromone diffusion messages defines how much pheromone information is spread at each step of the data bootstrapping process. Concretely, a lesser number of entries spreads little

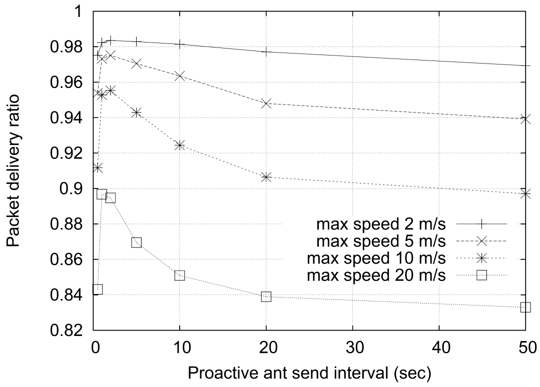


Figure 4.3: Delivery Ratio- proactive ant send rate

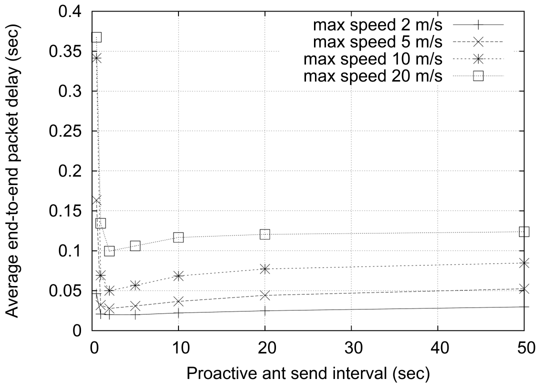


Figure 4.4: End-To-End Delay- proactive ant send rate

information and determines also a slow running of the bootstrapping process. 0 entries is the severe case where the supporting pheromone diffusion function is disabled and the proactive ants get no guidance. We perform simulation using 0, 2, 5, 10 and 20 entries. The destinations whose routing information is included in each message are elected randomly out of the known destinations stored in the pheromone table.

The graph plot show the significance of the supporting pheromone diffusion process: giving more efficiency to this process allows for better performance. In addition, for the tested sizes the gain of the increase in transmitted information is still greater than the negative impact due to the generation of larger messages.

Changing level of exploration β

The exponent β defines the depth of exploration allowed to the ants during their path search. For the reactive forward ants β is set to a fairly high value (approx 20) in order to reduce counterproductive ant proliferation and the establishment of sub-optimal paths at the start of a new data session.
On the other side, proactive ants are meant to walk around and check the path improvements suggested by bootstrapped pheromone. Therefore, we studied the effect of changing the degree of exploration allowed to the proactive forward ants by considering β values of 2, 5, 10, 20 and also the case of deterministic choice of the best path. We used scenarios with data rates of 1, 4 and 8 packets per second for the 20 CBR sessions.

The results shown in 4.7 and 4.8 depicts a large diversity in performance for the cases of 1 and 4 packets and that of 8 packets per second. Performance increases by dropping down exploration. These outcome show that in MANETs, exploration at the level of the ants do not really pay back due to constant changes and strong

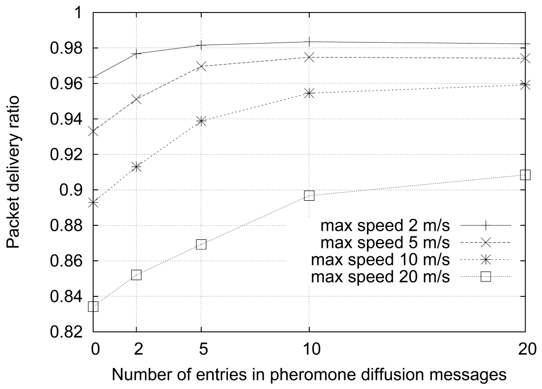


Figure 4.5: Delivery Ratio- pheromone diffusion messages

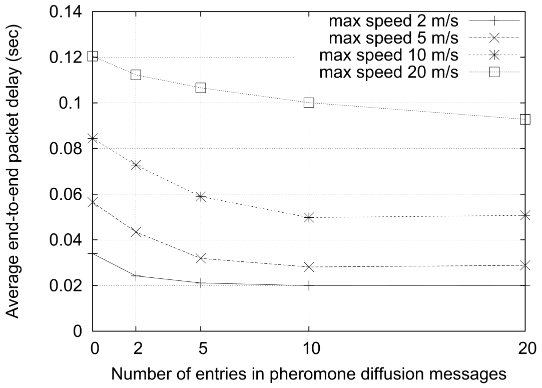


Figure 4.6: End-To-End Delay- pheromone diffusion messages

bandwidth limitations that limit the regularity of ant generation and the number of different paths that can be efficiently explored.

4.3.5 Comparison between MRUA and AntNet

Figure 4.9 shows the comparison of convergence time between the new Multicast tree based ant colony algorithm (MRUA) and the conventional AntNet on different scales of network topology. It is clear from the figure that with the growth of network topology, the convergence time of both algorithms increases as well. But it takes shorter time for MRUA to converge than the conventional AntNet. This is more promising with the increase of the topology scale. There are three reasons for this.
First, what MRUA has found is a tree while what AntNet has initially found is paths. Secondly, the time MRUA spends in finding a tree is not longer that the time AntNet searches paths and combines them into a tree. Third, due to the updating of pheromones of a entire tree when AntNet searches a path from source to destination node, the ant may be misled by the pheromone leading to another path, thus the convergence speed of the algorithm will be delayed. In contrast, the ant of MRUA cannot be misguided.

Figure 4.10 shows the comparison of performance of MRUA and AntNet in finding optimal multicast tree on different scales of topology. From the above figure it is clear that even though there is difference between the two algorithms in finding the best solution, approximately both are identical with different scales of topology. Hence, it indicates that the solution MRUA finds is not worse than AntNet.

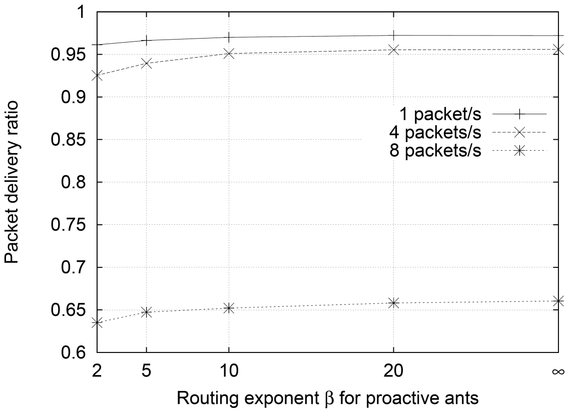


Figure 4.7: Delivery Ratio- level of exploration β

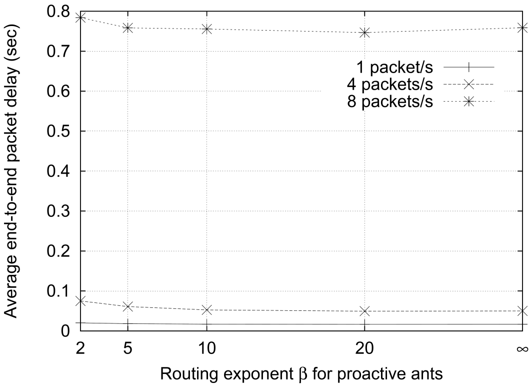


Figure 4.8: End-To-End Delay- level of exploration β

Based on the above experimental results, it can be found that the MRUA is much faster in convergence speed than the conventional AntNet. Their performance in finding the best solution is similar. So we can conclude that the MRUA is a more effective algorithm for multi-constraints multicast routing.

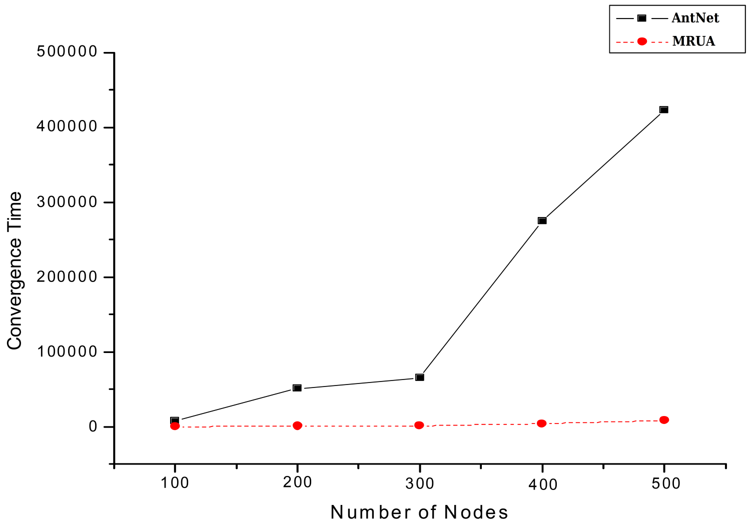


Figure 4.9: Convergence Time

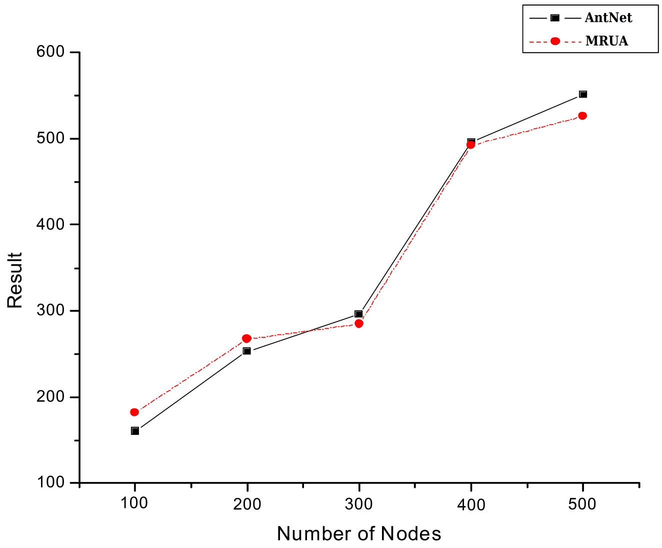


Figure 4.10: Quality Of Solution

4.4 Prioritized H.264 Video Transmission

Demand of multimedia applications like video streaming and video conferencing are increasing rapidly over wireless networks. Video data is delay-sensitive and requires more bandwidth to be sent. Wireless channel has limited bandwidth and is prone to errors due to time-varying signal strength, which limits the availability of bandwidth for these applications. Hence, video compression techniques are crucial for video applications.

The H.264/AVC video coding standard is widely used for streaming applications due to its high coding efficiency and the network friendly features. Loss of different packets in a video stream induces different amounts of distortion to the video quality; therefore, video bit stream should be prioritized. The basic IEEE 802.11 MAC protocol does not provide quality of service support to the users and cannot be used as the main access mechanism for transmitting prioritized video.

In this chapter, we discuss about a cross-layer and priority aware packet fragmentation scheme which transmits pre-encoded compressed H.264 video on IEEE 802.11e EDCA based MAC layer to enhance the video quality over error-prone wireless networks. Data fragmentation modifies the packet size by adapting to the varying channel conditions and improves successful video transmission. We also study the optimal packet size to maximize the weighted good-output for different number of users and channels bit error rates.

4.4.1 Challenges

The new H.264/AVC video coding standard [22] provides better compression efficiency and robustness over a range of bandwidth. The basic IEEE 802.11 [23] protocol is hardly sufficient to deliver such applications because of varying wireless channel characteristics and vulnerability of compressed video data to losses. The challenge is to improve the quality of compressed real time H.264 video transmitted in such an error prone channel.

The demand for real time multimedia streaming applications like video conferencing, video telephony, and network gaming over the wireless ad-hoc LAN is increasing rapidly. Unlike non real-time applications, real-time video applications are delay-sensitive; require high bandwidth, and have limited or no time for retransmission of lost data.

Video data delivery should be increased by designing the appropriate schemes which consider QoS requirements and adjust the medium access parameters according to the video characteristics. Also a trade-off exists between the desire to reduce the overhead by adopting larger packet size and the need to reduce packet error rates in the error prone environments by using smaller packet length.

Data packets formed at transport layer may use the maximum packet size requirements determined by the network. Doing packet fragmentation at MAC layer can adapt the packet size according to varying channel conditions and improve successful transmission probability. MAC layer fragmentation and retransmissions also avoid costly packet retransmissions at transport layer. In a video sequence, some data are more important than others. For example, the frame headers are much more important than data block because, once lost, the entire frame cannot be reconstructed well. These important data should be protected more to ensure that they can be received with lower uncorrected error rate. Loss of video packet induces different amounts of distortion of the quality, and hence we need to prioritize them and adapt fragment size according to its priority. Many solutions are proposed in the literature for this problem. An effective solution for H.264 video streaming application over a QoS based single-hop wireless ad-hoc LAN using packet fragmentation is proposed here.

The main objective of this work is to achieve good quality of prioritized H.264 video [24] streaming using the existing QoS based IEEE 802.11E protocol over wireless adhoc network by employing packet fragmentation at MAC layer. We consider timeout based buffer discard at MAC layer to discard video packets for a bounded end-to-end delay, as real time video is delay-sensitive and retransmission of lost fragments is not feasible.

This evaluation is carried out by showing the comparison of performance of three different schemes employed: the baseline scheme, the optimal packet size scheme, and the proposed scheme. Each of these schemes is simulated to extract optimal goodput, which is compared with the proposed scheme. These modules are developed in the network simulator (ns-2) [25] environment in order to facilitate the real-time prioritized H.264 video performance evaluation on IEEE 802.11E MAC layer with fragment burst implementation

4.4.2 Introduction To IEEE 802.11 Standards

In this protocol [23], two basic functions exist for medium access: the point coordination function (PCF) and the distributed coordination function (DCF). The fundamental access method of the IEEE 802.11 MAC is DCF, known as carrier sense multiple access with collision avoidance (CSMA/CA). The DCF can be implemented in all nodes for use within ad-hoc mode or infrastructure network configurations. While the DCF is responsible for asynchronous data services, the PCF was developed for time-bounded services. The PCF is used in the contention-free period (CFP), while the DCF pertains to the contention period (CP). IEEE 802.11 uses three different inter-packet gaps, denoted as inter-frame spaces to control the medium access, short inter-frame space (SIFS), PCF inter-frame space (PIFS), and DCF inter-frame space (DIFS) as shown in figure 4.11 [23].

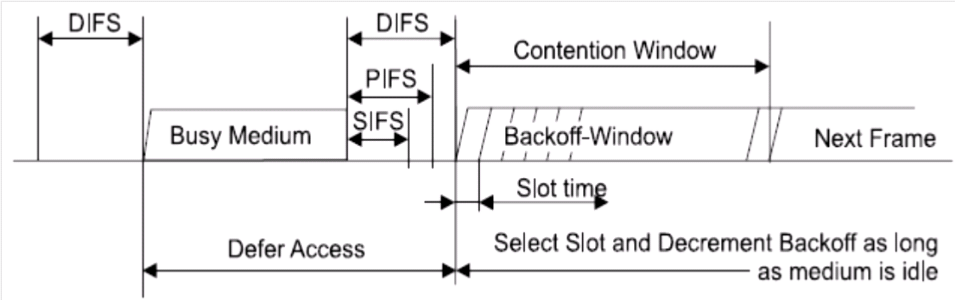


Figure 4.11: *Basic access mechanism in IEEE 802.11. Source: LAN MAN Standards Committee of the IEEE Computer Society, Sponsor, ANSI/IEEE Std 802.11, 1999 Edition: Part 11. Piscataway, NJ: IEEE, 1999.*

Fragment Burst Mode In IEEE 802.11

Due to high bit error rates in Wireless LAN, the probability of a large-sized packet being corrupted is more according to the formula, $PER = 1 - \left((1 - BER)^L\right)$), where L indicates the packet length. In such a condition, the smaller packets get transmitted successfully without any errors. Hence, the fragment burst mode is added to the IEEE 802.11 MAC mechanisms. Large frames are divided into smaller fragments according to the fragmentation threshold. Each fragment is transmitted and acknowledged separately. After a node gets an opportunity to transmit data, it is allowed to send all the fragments of a packet along with their ACKs separated by SIFS. All other nodes which hear these fragments set their NAV according to the duration of the transmission period as shown in the Figure 4.12 [23].

4.4.3 QoS Limitations of 802.11

DCF (Distributed Coordination Function)

→ Only support best-effort services.

→ No guarantee in bandwidth, packet delay and jitter.

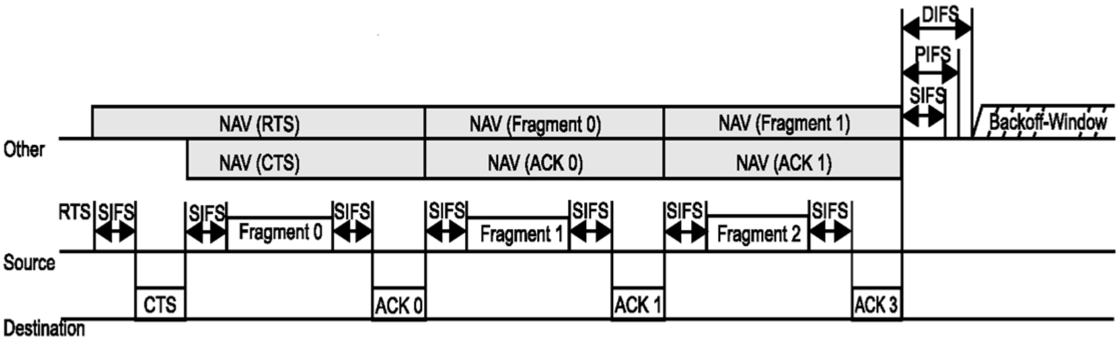


Figure 4.12: *Fragment burst scheme in IEEE 802.11. Source: LAN MAN Standards Committee of the IEEE Computer Society, Sponsor, ANSI/IEEE Std 802.11, 1999 Edition: Part 11. Piscataway, NJ: IEEE, 1999*

→ Throughput degradation in heavy load.

PCF (Point coordination Function)

→ Inefficient central pooling scheme.

→ Unpredictable beacon frame delay due to incompatible cooperation between CP and CFP modes.

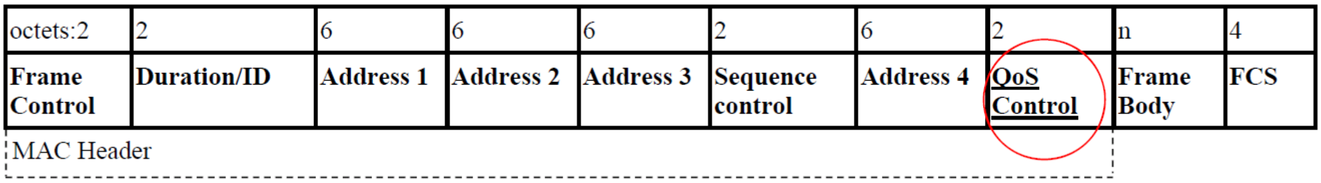
→ Transmission time of the polled stations is difficult to control.

So IEEE 802.11E came into existence to support QoS.

IEEE 802.11E MAC

IEEE 802.11 Task Group E [26] defines enhancements to the above-described IEEE 802.11 MAC, called IEEE 802.11e, which introduces ***EDCF*** *(Enhanced Distributed Coordination Function)* and ***HCF*** *(Hybrid Coordination Function)*. With IEEE 802.11e, there may still be the two phases of operation within the super frames, i.e., a CP and a CFP, which alternate over time continuously. The EDCF is used in the CP only, while the HCF is used in both phases, which makes this new coordination function hybrid.

IEEE 802.11e MAC frame Format: Shown in Figure 4.13



octets:2	2	6	6	6	2	6	2	n	4
Frame Control	**Duration/ID**	**Address 1**	**Address 2**	**Address 3**	**Sequence control**	**Address 4**	**QoS Control**	**Frame Body**	**FCS**

MAC Header

Figure 4.13: *IEEE 802.11 MAC header format. Source: 802.11-2012 IEEE Standard for Information Technology, Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks, Specific Requirements Pt 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*

4.4.4 Literature Survey of IEEE 802.11

There has been extensive research on the performance of IEEE 802.11 CSMA/CA. The approximation of saturated DCF as p-persistent CSMA is important for the analysis of DCF. Efforts have been made to analyze

Bits	**Usage**
0-7	UP for prioritized QoS (TC)
8-15	TSID for parameterized QoS (TS)

Table 4.1: Traffic Identifier Field in QoS Control of MAC Header

the performance of the IEEE 802.11 by simulation employing analytical models with back off rule [30]. In this paper, they employ a Markov model to represent the exponential back off in DCF, but it assumes the presence of ideal channel conditions. Wireless link is error-prone, and channel conditions vary essentially due to signal strength and it impacts on throughput performance of stations.

In [31], the performance of IEEE 802.11 DCF under saturated traffic condition is analyzed considering the effects of packet size, number of contending nodes, transmission collision, and the packet error probability. The authors derived an optimal packet size which achieves maximum network saturation throughput for a WLAN in IEEE 802.11 distributed coordination function mode. They considered the overhead due to retransmissions at MAC layer and evaluated the impact of error-prone channel on unsuccessful transmission probability and its impact on the performance. The channel characteristics were assumed to be constant over the network.

Packet length adaptation at MAC layer according to varying channel conditions is discussed in different papers. In [32], a rate-adaptive protocol with dynamic fragmentation is proposed to enhance the throughput over time-varying Ricean fading channels based on fragment transmission bursts and channel information in WLANs. The authors use multiple fragmentation thresholds for different rates according to channel condition. They evaluated IEEE 802.11b PHY performance using symbol error rate instead of BER and then used SER to find packet error rate based on number of symbols in data packet.

A cross layer adaptive design between MAC and PHY layer in IEEE 802.11 wireless LANs is studied in [33] for the frame length control. The authors derived an optimal frame length for direct sequence spread spectrum and frequency hopping spread spectrum, which depends on the SNR of the channel.

In [34], dynamic packet fragmentation algorithm, which matches channel characteristics adaptively, is proposed to alleviate the problem of MAC retransmissions of packets of different sizes which result in power-law delay and poor throughput for a highly concentrated Gaussian or exponential distribution of data. It also proposed mechanisms for aggregating smaller packets into larger ones, which in combination with fragmentation improves the performance.

In [35], an algorithm is proposed to enhance system goodput through dynamic optimal fragmentation, in which a sender estimates the SNR of a receiver adaptively and chooses a fragmentation threshold to shape arbitrary sized packets into optimal length packets.Due to wireless channel characteristics and lack of QoS support, the basic IEEE 802.11 DCF based channel access procedure is merely sufficient to deliver non-real time traffic. The delivery should be augmented by appropriate mechanisms to better consider different QoS requirements and ultimately adjust the medium access parameters to the video data content characteristics. There is a need to consider prioritized video packet transmission over an error-prone channel to provide more opportunity to transmit higher priority video compared to lower priority to achieve good quality. There are many papers in literature which have considered video prioritization for improving quality.

In [36] and [37] a robust cross layer architecture is proposed which uses the error resiliency features of H.264 and QoS-based IEEE 802.11e MAC protocol possibilities to show the performance of video. Data partitioning, an error resiliency tool [38], is present in H.264 extended profile which divides the video data into three partitions (DPA, DPB, and DPC) with different levels of priority. The authors use this tool in H.264 to map the different IEEE 802.11e MAC access categories to different partitions, which shows graceful degradation of video while minimizing packet loss and end-to-end delay.

In [39], the authors perform the classification of data partitioned H.264 video data into IEEE 802.11e MAC access categories and bandwidth reservation through Controlled Access Phase Scheduling in hybrid coordination function controlled channel access mode to show the performance.

The authors of [40] demonstrate that the performance of H.264 video applications can be improved over wireless LANs through a cross layer design, which optimizes the encoded H.264 video slices. They proposed the mechanisms for fragmentation and aggregation of H.264 NALUs (Network Abstraction Layer Units) in order to enhance the quality of decoded video. They showed the video quality can be increased by performing fragmentation in application layer than compared to doing that in MAC layer. They did not consider priorities of video slices and also combining fragmentation at both of these layers.

In [41], the authors show that there exists an optimal packet size of encoded video that minimizes packet loss rate and maximizes the video quality during transmission over 3G networks by performing fragmentation of NALUs in application layer.

In [42], authors investigated the improvement in quality of H.264 video as affected with the increase in number of concurrent video streams sent over a multi-rate IEEE 802.11e network. They compared several packet mapping schemes to study the performance of video and showed that mapping schemes that differentiate video based on the frame type are more successful at maintaining good quality when congestion occurs compared to other techniques.

There has been extensive research recently towards cross-layer protocols to transmit video applications over wireless networks. The [43] and [44] proposed a cross-layer priority aware packet fragmentation scheme at the MAC layer to enhance quality of pre-encoded H.264 video over error-prone wireless networks. The authors calculated optimal fragment size for different priority levels using branch-and-bound technique combined with multidimensional arithmetic interval methods. They extended packet fragmentation with slice fragmentation by modifying conventional H.264 decoder to handle partial slice decoding and showed that the scheme provided considerable PSNR and VQM gains over priority agnostic fragmentation.

The authors of [45] discuss several techniques that provide significant performance gains through cross-layer optimizations. They mainly talk about the improvements of adaptive link layer techniques such as adaptive modulation and packet-size optimization

The [46] discusses cross-layer techniques which can be used to send the information about current time-varying channel conditions to upper layers to improve the overall performance of the data transmitted over wireless networks.

In [47] the authors give an excellent review of the existing solutions for combining techniques deployed at the application layer, and techniques available at either the PHY or the MAC layer. They classified cross-layer architectures for video transport over wireless networks into five categories:

→ Top-down: The higher layer optimizes their parameters and the strategies at the next lower layer.

→ Bottom-up approach: In this architecture the lower layer isolates the higher layers from losses and bandwidth variations.

→ Application-centric approach: The application layer optimizes the lower-layer parameters one at a time in either bottom-up (starting from the PHY layer) or top down manner, based on its requirements.

→ MAC-centric approach: In this cross-layer technique the application layer passes its traffic information and requirements to the MAC, which decides which application layer packets/flows should be transmitted and at what QoS level.

→ Integrated approach: The strategies to design cross-layer architecture are determined jointly by all the open system interconnection (OSI) layers.

4.4.5 Existing Techniques:Baseline and Optimal Packet-Size

In this chapter, the baseline scheme and optimal packet size schemes are compared against the proposed scheme for analysis. These two schemes are explained in this chapter. Simulation is performed for each of the schemes and the results are compared.

4.4.6 Simulation Setup Outline: For All Schemes

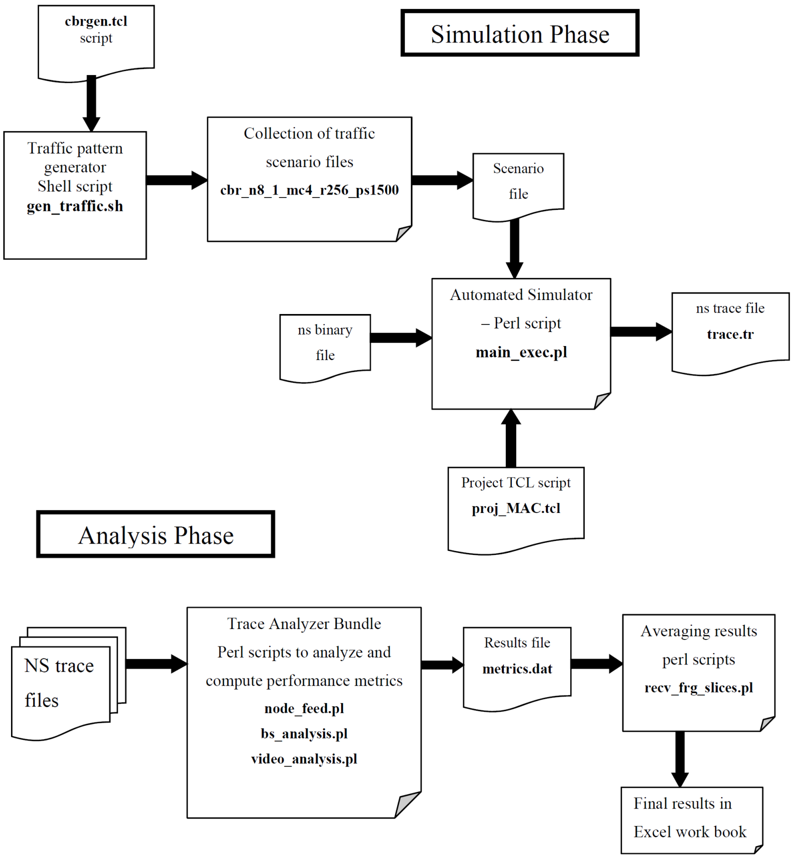

cbrgen.tcl
script
Simulation Phase
Traffic pattern generator Shell script gen_traffic.sh
Collection of traffic scenario files cbr_n8_1_mc4_r256_ps1500
Scenario file
Automated Simulator – Perl script main_exec.pl
ns binary file
ns trace file trace.tr
Project TCL script proj_MAC.tcl
Analysis Phase
NS trace files
Trace Analyzer Bundle Perl scripts to analyze and compute performance metrics node_feed.pl bs_analysis.pl video_analysis.pl
Results file metrics.dat
Averaging results perl scripts recv_frg_slices.pl
Final results in Excel work book

Figure 4.14: *Block diagram depicting the simulation flow using ns-2*

4.4.7 Baseline Scheme

In this scheme, the ad-hoc LAN is set up for transmitting H.264 video with no priority using the MTU packet size 1500 bytes. The network has unique source-destination node pair connections which communicate at different instants of time. The realistic channel has collisions and known bit error rates which affect the transmission of video packets. Hence, we consider retransmissions at MAC layer to ensure proper delivery of these packets.

Channel Bit Rate	2 Mbps
RTS	44 bytes
CTS	38 bytes
ACk	38 bytes
SIFS	10 μs
DIFS	50 μs
MAC header	34 bytes
PHY header	24 bytes

Table 4.2: Simulation Parameters for Baseline Scheme

Timeout Based Buffer Discard

The transmission of real-time video applications has certain bandwidth, delay, and loss requirements. In contrast to data transmission, which is usually not subject to delay constraints, real time video requires bounded end-to-end delay. That means every packet should arrive at the destination in time to be decoded and displayed for the video to be played continuously. A video packet which arrives beyond this time constraint is useless and will be considered lost even if it has no errors. Hence, to achieve a real-time effective transmission of video packets in the network we have employed a module in each node which performs timeout-based buffer discard before sending the packets over network. This module is implemented in MAC layer which gets packets from upper layers in the queue. Each packet has a timestamp associated with it, which is assigned in the UDP layer when the packet is being formed. This timestamp is compared with the current time to check if it has exceeded a threshold timeout value. The timeout value is selected according to the delay bound for video. For real-time video applications, we use a delay bound of 250 ms. If the timestamp of a packet exceeds this timeout value, the packet will be discarded before sending it over the channel. This early-detection logic ensures that packet delay is limited to the timeout value.

Simulation Setup

The baseline scheme is set up for simulation using the network simulator ns-2 [25]. For simulation of n contending nodes within transmission range of each other, there are n/2 pairs of connections, i.e., n/2 pairs of sender and receiver nodes. The H.264 video packets are formed in UDP layer, sent to the lower layers, and transmitted across the network. The channel bit rate is 2 Mbps. The known channel bit error rates (BER) vary from considerably good channels to worse channels with errors. The connections vary from 2 to 9 to 5 to 18 nodes respectively. The video traffic bit rate is 256 kbps or 512 kbps. The system parameters and network configurations used in this scheme are shown in Table 4.2

The video data formed into 1500 byte packets are transmitted over an error-prone channel with certain bit error rate. The error module in ns-2 is used to insert errors in the channel in terms of packet error rates (PER). The following formula is used to calculate PER: $PER = 1 - ((1 - BER)^L) \quad ; L : \ is \ packet \ length$ The fixed channel BERs and their corresponding PERs for a packet length of 1500 bytes used in these simulations are listed in Table 4.3

The goodput is computed as the ratio of the payload transmission time to the total time needed to transmit a data packet. These goodput values are calculated for each connection in the steady state condition.

Baseline Scheme Results

The following figures show the plots for goodput versus the bit error rates for different node variations in a steady state network with retransmissions enabled. Since the packet error rate depends on the BER and packet

Bit Error Rate (BER)	Packet Error Rate (1500 bytes)
1×10^{-6}	0.0124
1×10^{-5}	0.1172
5×10^{-6}	0.4638
1×10^{-4}	0.7125

Table 4.3: Packet Error Rates Calculated for different BERs for 1500 Byte Packet Length

length in an error-prone environment, when the channel BER is higher, transmitting a large packet like 1500 bytes has a greater probability of getting corrupted. The goodput obtained in such a condition will be very low. The number of nodes participating in the network also affects the chance of obtaining better goodput values.

In the Figure 4.15, variation in the goodput values can be seen when the channel BER increases. Each node in the network transmits data with 256 kbps rate to its peer node. With the given channel capacity 2 Mbps, the total available bandwidth decreases as the number of connections increases. Hence, we observe that the goodput drops significantly when the number of connections increases.

With retransmissions, the goodput will be boosted when the channel is in considerably good condition. Hence, the goodput values seem to be improving when channel has fewer errors.

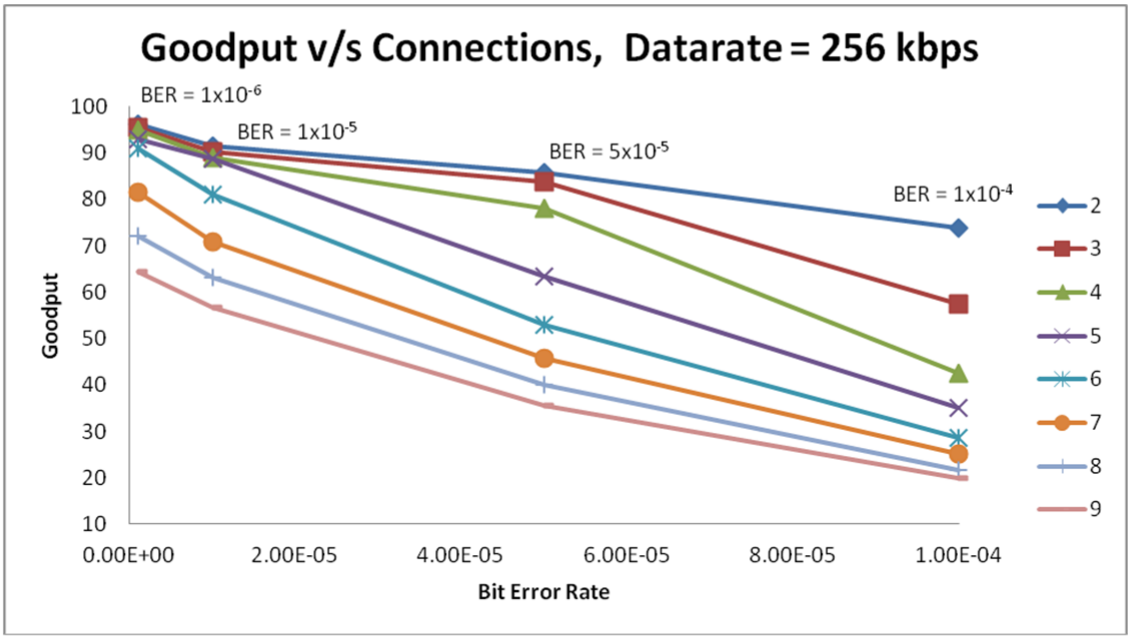


Figure 4.15: *Variation in goodput with channel BERs for different connections in baseline scheme*

Figure 4.16 shows variation in goodput against different BERs with varying number of connections. When BER is 1×10^{-4}, we observe that the goodput is 72% for 2 connections; it decreases exponentially and falls to 20% when there are 9 connections. This shows that the bandwidth available for data transmission decreases when number of connections increases, and the probability of getting packet corrupted increases with greater packet length. Hence for higher BER, the goodput deteriorates showing worse performance for increasing number of connections. There is a need to achieve better performance when the channel is worse, and this can be achieved using proposed scheme, which is explained later.

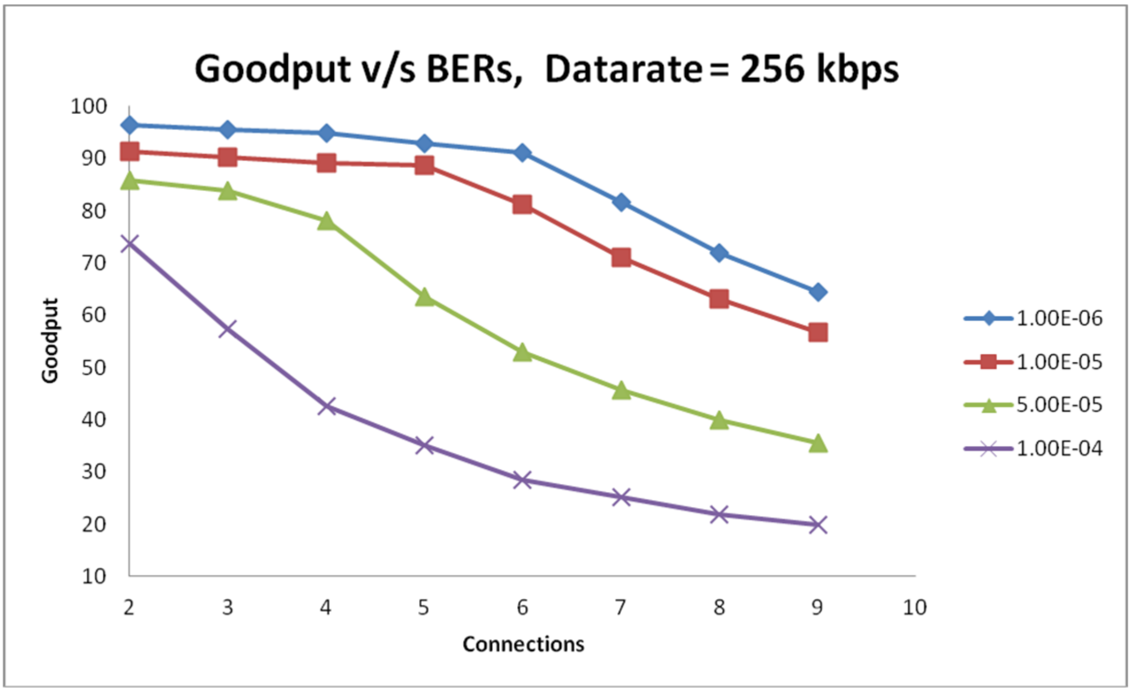


Figure 4.16: *Variation in goodput with number of connections for different BERs in baseline scheme*

4.4.8 Optimal Packet-Size Scheme

A wireless link is error-prone due to the variation in signal strength. In such conditions, larger packets sent over this channel are vulnerable to loss, and hence the cost of retransmissions is increased to achieve successful data transmission. There exists a trade-off between the desire to reduce the overhead by selecting larger packet length and to reduce the packet error rate by selecting a lower packet length in an error-prone channel. In this scheme, we consider IEEE 802.11b DCF protocol in error-prone WLANs and show that there is an optimal packet size, which, when sent across the medium, maximizes the network goodput for a given channel BER and number of connections under saturated system. This scheme considers the basic MAC protocol without QoS to find the required optimal packet length to be transmitted over the network which has maximum goodput.

To analyze the performance of IEEE 802.11b DCF protocol in an ad-hoc mode, we must consider the system under saturated traffic condition. This is also known as the steady state analysis wherein we take into account a number of factors like total contending nodes in the network, channel conditions, packet size, transmission probability of each node, collision probability, and PER. To analyze the system performance in steady state condition, we run each simulation for a long duration for the given number of connections and BER by varying seed values. The seed values randomize placement of nodes in the network. We average the goodput obtained for each seed over the total number of seeds used. The seed values are increased linearly to calculate goodput till we have a constant average goodput. Thus calculated average goodput remains the same even when the seed values are increased further. The average goodput obtained through this Monte-Carlo procedure is called the expected goodput.

Simulation Setup

In this section, the simulation results obtained by using ns-2 are discussed. The number of nodes varies from 5 to 18 for 2 and 9 connections, respectively. Each source node sends UDP/CBR traffic with data rate of 256 kbps or 512 kbps to its corresponding destination node at different instants of time. The packet sizes used by nodes vary from 200 to 1500 bytes. We enable timeout based buffer discard explained earlier at 250 ms to ensure that the end-to-end latency is controlled. The simulation is carried out over 150 second duration, and the expected goodput values are computed at steady state condition as explained in previous section. We will observe that for a particular BER there will be an optimal packet size which gives maximum goodput

Optimal Packet-Size Scheme Results

In this section, we discuss the simulation results for optimal packet size scheme. Each of the following plots shows the goodput variation for different BERs and for different number of connections. In Figure 4.17, we can observe the goodput values to be 90% and above for 2 connections when channel BER is higher, showing that goodput is better with the increasing packet sizes. For BER 1×10^{-4}, the maximum goodput occurs at 200 byte packet size and decreases as the packet sizes increase. This is due to the reason that, when the channel BER is higher, smaller packet length data has more probability to be received correctly, but, when the packet length is higher, it is more likely to be corrupted. Hence, we observe that at 1500 byte packet size the goodput drops to 75%.

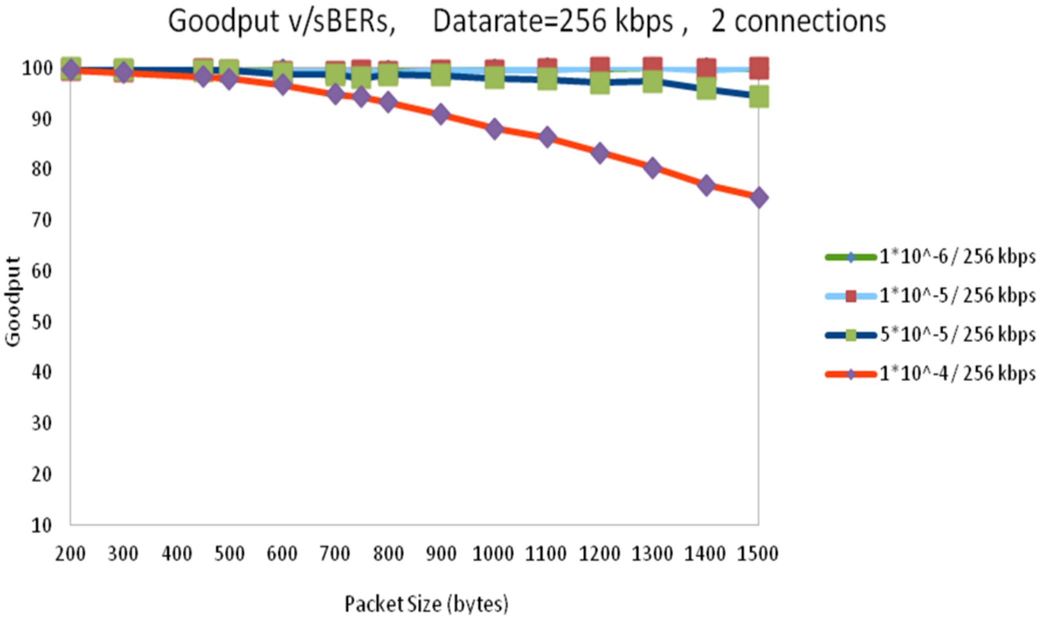


Figure 4.17: *Optimal goodput values with variation in packet sizes for 2 connections for different BERs in the optimal packet size scheme*

Figure 4.18 shows goodput plotted against BERs for 3 connections with varying packet size values. The maximum goodput occurs at 500 byte packet size. This can be explained as follows: the packet size at which maximum goodput is obtained is considered as optimal packet size Lo. When packet size $L < L_0$, excessive overhead in each packet limits the goodput, and when $L > L_0$, packet errors limit the goodput.

In Figure 4.19, the maximum goodput occurs at 500 bytes for 1×10^{-4} BER. For 5×10^{-5}, the optimal packet size is 800 bytes where the goodput is maximum. We observe that the goodput remains fairly same for BERs 1×10^{-6} and 1×10^{-5}, because of less number of connections. The maximum goodput occurs anywhere from 500 bytes through 1500 bytes packet size.

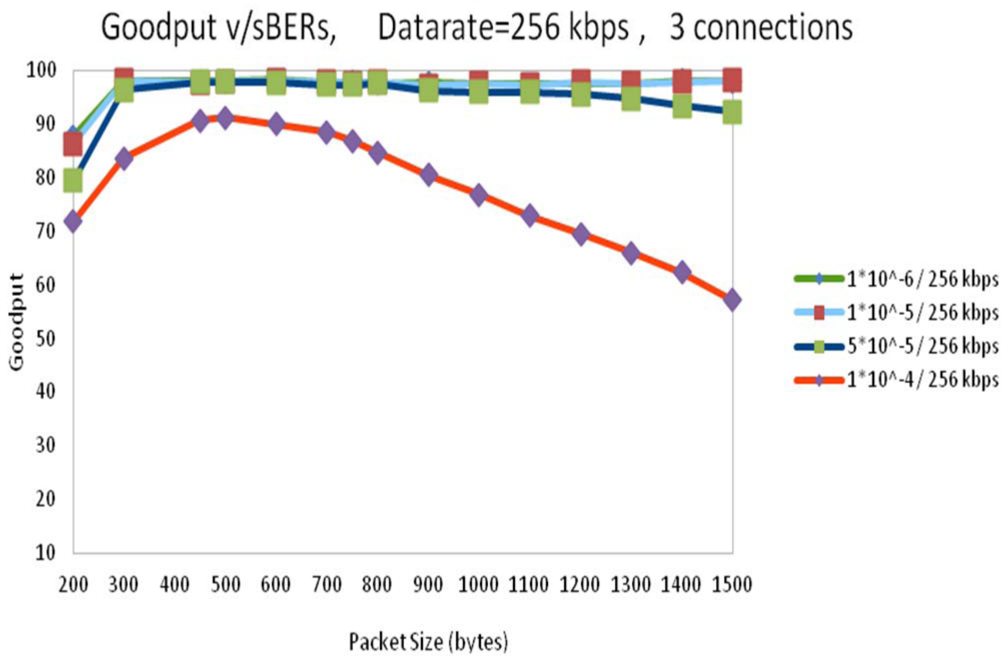


Figure 4.18: *Optimal goodput values with variation in packet sizes for 3 connections for different BERs in the optimal packet size scheme.*

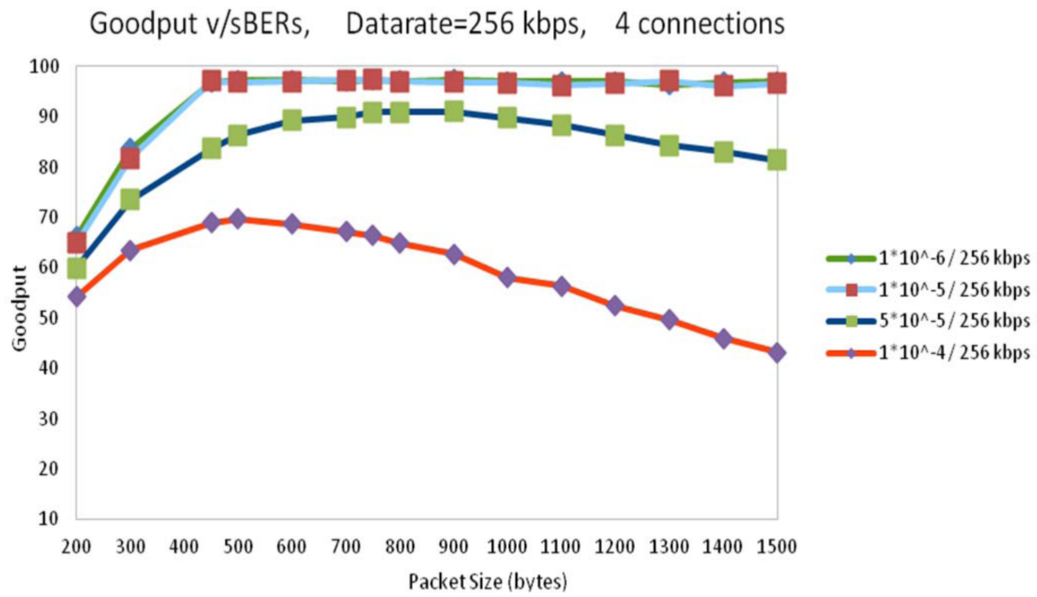


Figure 4.19: *Optimal goodput values with variation in packet sizes for 4 connections for different BERs in the optimal packet size scheme*

In Figure 4.20, we observe that for BER 5×10^{-5}, the maximum goodput occurs at 800 byte packet size, and it decreases on either side less than or more than 800 byte packet.

Figure 4.21 shows optimal packet size occurring at 500 bytes for BER 1×10^{-4}, 800 bytes for BER 5×10^{-5}. We observe that there is some difference in goodput variation between BERs 1×10^{-5} and 1×10^{-6} due

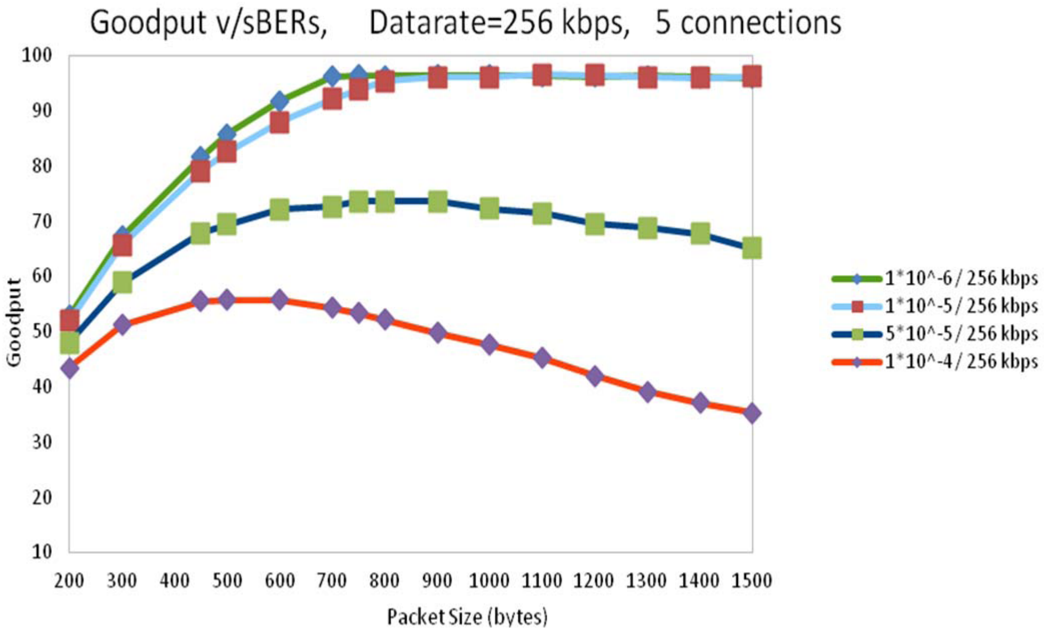


Figure 4.20: *Optimal goodput values with variation in packet sizes for 5 connections for different BERs in the optimal packet size scheme.*

to more number of connections but optimal packet size is 1500 bytes.

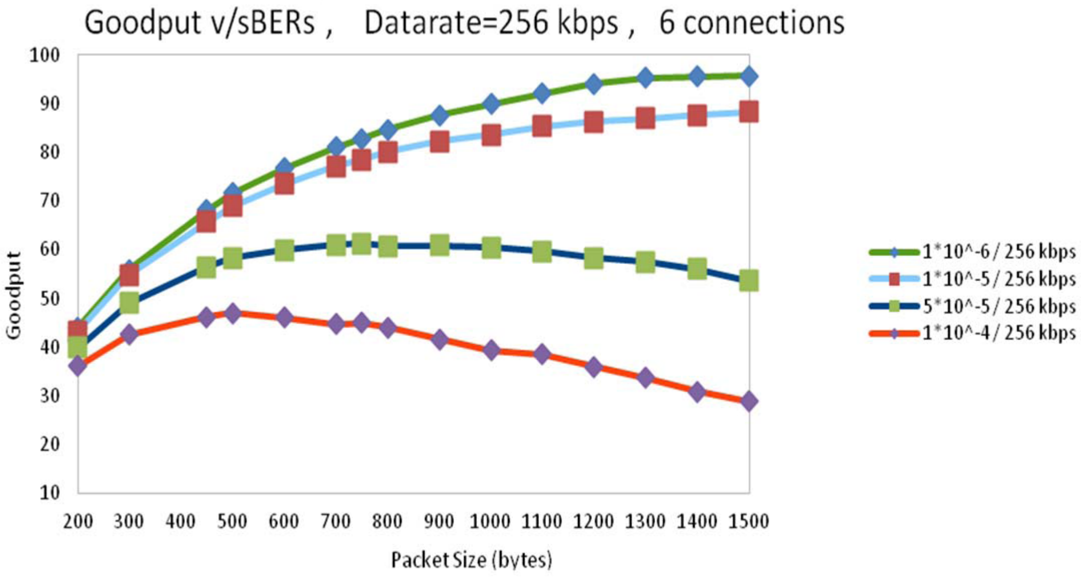


Figure 4.21: *Optimal goodput values with variation in packet sizes for 6 connections for different BERs in the optimal packet size scheme*

Figure 4.22 shows optimal packet sizes similar to that of seven connections shown in Figure 4.21, but the goodput obtained has decreased. Similarly, in Figure 4.23, the variation in goodput is as shown in Figure 4.22. The optimal goodput occurs at the same optimal packet sizes but there is a decrease in the goodput value. This is because of increase in number of connections.

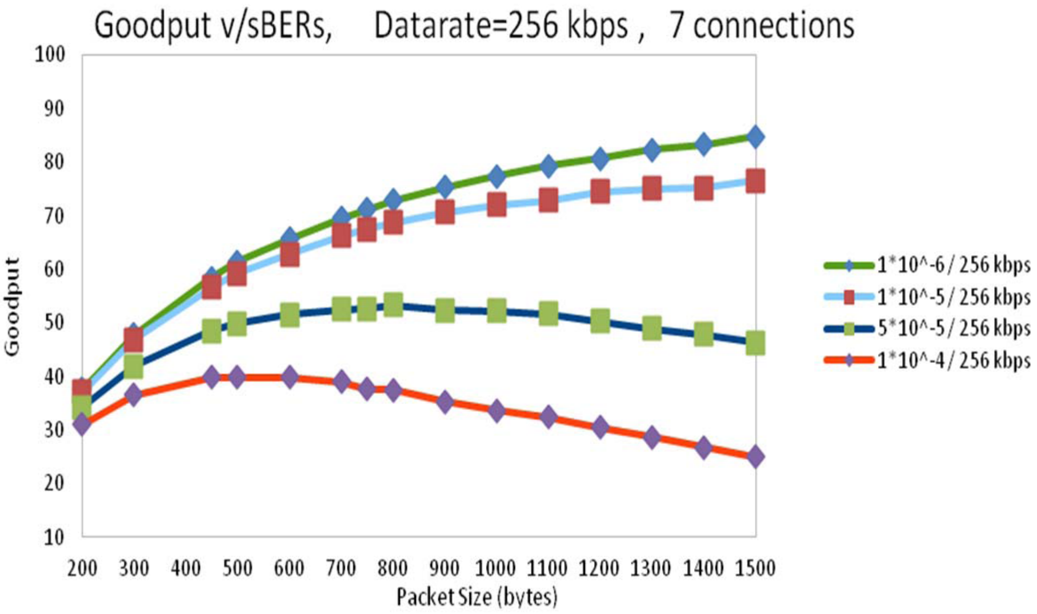


Figure 4.22: *Optimal goodput values with variation in packet sizes for 7 connections for different BERs in the optimal packet size scheme.*

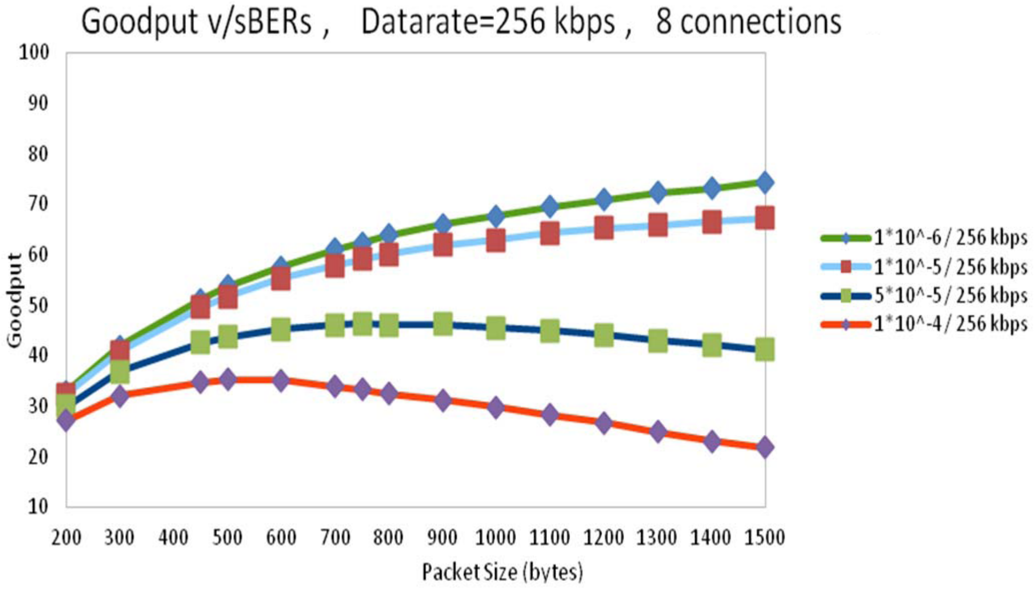


Figure 4.23: *Optimal goodput values with variation in packet sizes for 8 connections for different BERs in the optimal packet size scheme*

The Figure 4.24 shows optimal packet sizes occurring at 500 bytes for BER 1×10^{-4}, but the maximum goodput occurred is less than 30% for nine connections. For BERs 1×10^{-5} and 1×10^{-6}, the maximum goodput i.e. around 60% which occurs at 1500 bytes. This shows that the scheme performs very poorly when the channel is worse or when number of connections is increased.

When the number of connections increases, we observe that for BER 1×10^{-6}, the maximum goodput

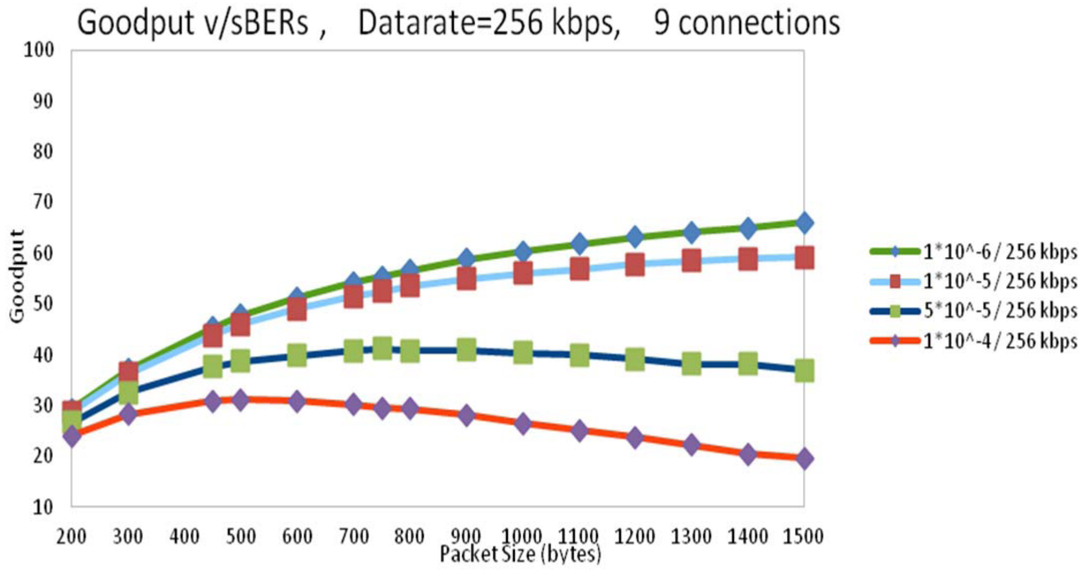


Figure 4.24: *Optimal goodput values with variation in packet sizes for 9 connections for different BERs in the optimal packet size scheme.*

BER	1.00E-06		1.00E-05		5.00E-05		1.00E-04	
Connection	packet size	goodput	packet size	goodput	packet size	goodput	packet size	goodput
2	200	100.0	200	100.0	200	97.9	200	95.6
3	600	98.3	300	98.1	450	95.8	450	91.2
4	900	97.4	750	96.4	750	91.0	500	69.7
5	750	96.4	1100	94.6	750	73.7	500	55.8
6	1500	95.7	1500	88.3	800	61.3	500	47.0
7	1500	84.8	1500	76.6	800	53.3	500	39.9
8	1500	74.4	1500	67.2	800	46.4	500	35.3
9	1500	66.1	1500	59.2	800	41.2	500	31.2

Table 4.4: Goodput Values Obtained at Optimal Packet Sizes for Different BERs and Connections for Data Rate 256 Kbps

occurs at 1500 byte packet size. This shows that the goodput is more when channel has less errors, and it keeps increasing with the increase in packet size. From these plots, we observe that the optimal packet size value keeps increasing with reduction in channel BER. The minimum optimal packet size is required to send the packets in a highly error-prone channel.

Table 4.4 shows the variations in goodput values and optimum packet sizes for different BERs and connections for 256 kbps data rate. Each value is the maximum goodput obtained using the optimum packet size. For a given BER, the optimal packet size increases and the corresponding goodput decreases with the number of connections. This can be explained as follows: the maximum goodput occurs at particular packet size for a given number of connections. As the number of connections increases, the bandwidth available to transmit data decreases, and hence the goodput decreases. In such condition, there will be need to reduce overhead by selecting a large packet size, but since there are more channel errors, we will have to limit the packet size at the value where the goodput is maximum. This is the optimal packet size for achieving maximum utilization of the network with limited resources.

Figure 4.25 shows variation in goodput with connections for different BERs for 256 kbps data rate. Each point in the plot is obtained by using an optimal packet size shown in the figure where optimal goodput is maximum. This graph represents data shown in Table 4.4. We observe that the optimal packet sizes increase with the decrease in channel BER.

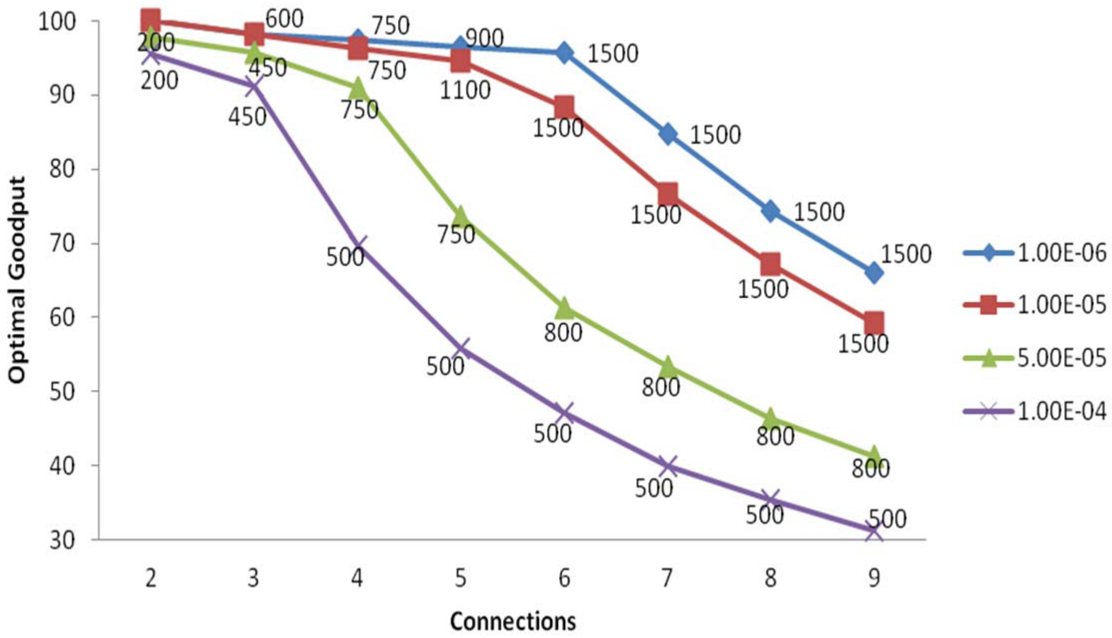


Figure 4.25: *Optimal goodput values with variation in number of connections for different BERs for data rate 256 kbps*

The H.264 video is sent using this particular optimal packet size value to evaluate the video quality, which is explained in later section. Table 4.5 shows maximum goodput values obtained at optimal packet sizes listed in next column for data rate 512 kbps.

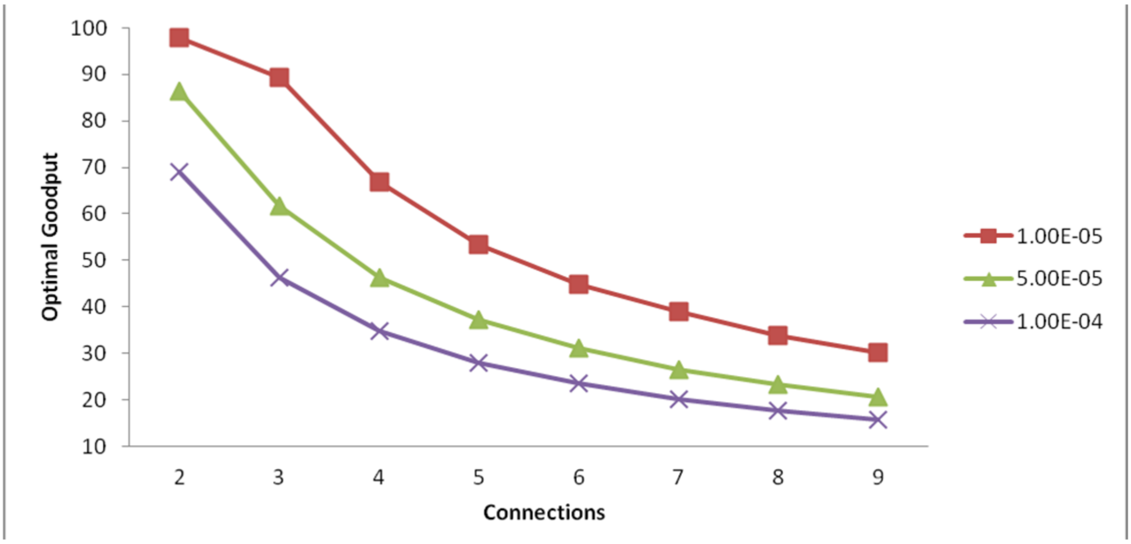


Figure 4.26: *Optimal goodput values with variation in number of connections for different BERs for data rate 512 kbps*

BER	1.00E-05		5.00E-05		1.00E-04	
Connection	packet size	goodput	packet size	goodput	packet size	goodput
2	1200	97.8	800	86.4	500	69.0
3	1500	89.4	750	61.8	500	46.3
4	1500	66.9	800	46.3	500	34.8
5	1500	53.3	800	37.1	600	28.0
6	1500	44.8	800	31.1	500	23.5
7	1500	39.0	800	26.4	500	20.0
8	1400	33.8	900	23.2	500	17.6
9	1500	30.1	800	20.7	600	15.7

Table 4.5: Goodput Values Obtained at Optimal Packet Sizes for Different BERs and Connections for Data Rate 512 Kbps

Figure 4.26 represents the data shown in Table 4.5. The optimal packet-size scheme shows improvement over baseline scheme when the channel BER is high. In baseline scheme, the data is transmitted using 1500 byte packet for all BERs and it had greater probability of being corrupted and hence goodput decreased. But in this scheme, we considered using a lower packet size when the channel BER is high and hence the overall performance is better compared to baseline scheme.

This scheme does not consider priority of data packets transmitted. When the channel BER is higher, the goodput achieved is significantly lower for nine connections transmitting data at 256 kbps. To improve this and achieve more goodput we have designed the proposed scheme explained in next chapter.

4.5 Proposed Prioritized Scheme

In this scheme, we transmit the prioritized H.264 video packets through access categories in IEEE 802.11e MAC layer and perform fragment burst mode to transmit the fragments over the network. The encoded H.264 video has prioritized self-contained slices which can be transmitted and decoded independent of others. These slices are formed using macro blocks in a video frame and have same timestamp value, which is the time when the frame is generated. The video trace file is attached to node using EvalVid module which is explained in detail in a later section. The video packets are sent to the network layer where the priority information of each packet is saved in the Traffic class field of its header. This information is later accessed in MAC layer which puts it in the QoS field of MAC header to segregate the packets in different access categories.

4.5.1 IEEE 802.11E For Prioritized Video Data Transmission

The UDP/CBR data is used to form data packets of size varying from 200 to 1500 bytes, which are transmitted over the access categories to find the optimal goodput for each priority. The optimal packet sizes calculated for each priority are used as fragmentation thresholds for transmitting H.264 video for a given channel BER and number of connections. Figure 4.27 shows Node Architecture in the proposed scheme. At the application layer, prioritized video slices are aggregated and sent to the transport layer to form MTU size video packets. At MAC layer, these packets are segregated into different access categories (ACs) according to their priorities. There is timeout based buffer discard implemented here to ensure early-detection of expired packets. Each AC contends for the channel access with others. When the AC gets channel access, it sends the packet to fragment burst module where it is divided by the fragment threshold selected for the given priority. The fragments thus obtained are transmitted over error-prone channel. The timeout logic explained in 4.4.5 is used here with four different timeout values for four priority access categories. A prioritized timeout value is defined for each priority

Priority	**Short Retry Limit**	**Long Retry Limit**	**TTL (ms)**
1	7	4	500
2	5	4	450
3	4	4	400
4	2	2	250

Table 4.6: Retry Limits and Timeout Values Used in Proposed Scheme

Access Category	Priority	CWmin	CWmax	AIFS
AC3-Highest	1	3	7	2
AC2-High	2	7	15	2
AC1-Low	3	15	31	3
AC0-Lowest	4	31	1023	7

Table 4.7: Contention Window and AIFS Values for Four MAC Access Categories of IEEE 802.11e Used in Proposed Scheme

in UDP layer. A higher priority packet will be assigned a higher timeout value and the lower priority a lower value. It is important to make sure that the higher priority packets get more time period to be sent without any loss at the cost of lower priority packets. The higher priority video slices contribute more to video quality compared to lower priority packets, and hence we can afford to lose the latter ones.

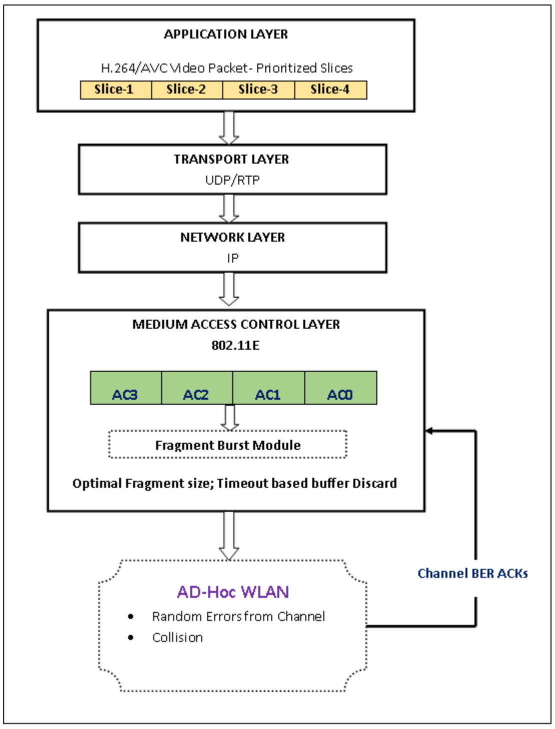


Figure 4.27: *Node architecture in proposed scheme*

The retry limits and timeout values chosen for this scheme are shown in Table 4.6

The IEEE 802.11e MAC layer reads priority information present in each packet header and puts it in the access categories accordingly. The contention window and ***AIFS (Arbitration Inter Frame Space)*** values are assigned to each access category as shown in Table 4.7

Fragment Burst Module

This module is a feature in IEEE 802.11, which performs fragmentation on the prioritized video packet using fragmentation threshold. Each node uses a pre-computed optimal packet size, which maximizes the expected goodput of the network for a given channel BER. This optimal packet size is used as fragmentation threshold which divides the prioritized video packet coming from network layer into fragments.

The fragment ID and size of each fragment is put in its header along with other video packet header information like source node, destination node, and priority present in that packet. Each packet coming out of access category undergoes fragmentation, where each of its fragments is transmitted and immediately acknowledged independent of others.

The fragmentation threshold is selected according to priority of packet. In an error prone channel the highest (lowest) priority packet will have the lowest (highest) fragmentation threshold. As the packet length decreases, the PER decreases and hence the probability of packet being transmitted successfully without error increases. The highest priority packet has to be sent with more accuracy and success probability than the lowest priority packet.

In this scheme, the optimal packet sizes for each priority were pre-computed for a combination of different nodes, number of connections, and BERs. These values are used as fragmentation thresholds for each priority, which are used to perform fragmentation of the packet. The fragment burst module explained above picks a correct threshold value according to the priority of packet before fragmenting it.

For selecting optimal packet size, the packets were transmitted and the goodput was calculated for each priority packet and the total optimal goodput was calculated from the network. We will have an optimum packet size for each priority which maximizes the goodput for a given channel bit error rate and node combination.

In this work, EvalVid framework is used along with ns-2 to evaluate the H.264 video quality as shown in Figure 3.4. In place of the network, this simulator which is connected with the framework using trace files is used. Three connecting simulation agents, namely MyEvalVid, MyUDP, and MyUDPSink, are implemented between ns-2 and EvalVid. These interfaces are designed to read the video trace file, to generate data required, and to evaluate the delivered video quality.

The video used here is pre-encoded using H.264 AVC with fixed slice size configuration; macro blocks are aggregated into a slice so that their accumulated size does not exceed a pre-defined size [43], [44]. The network limits the number of bytes that can be transmitted in a single packet based on the MTU size over the channel. The slices thus formed are prioritized based on their distortion contribution to the received video quality.

The total distortion of one slice loss is computed using CMSE (Cumulative Mean Square Error), which takes into consideration the error propagation within the entire GOP (Group of Pictures). All slices in a GOP are distributed into four priority levels based on their precomputed CMSE values. Priority 1 slices induce the highest distortion whereas Priority 4 slices induce the least distortion to the received video quality.

The pre-encoded H.264 video trace file has information of each slice number, size, time stamp, priority, frame number and frame type. This file is read by MyEvalvid module through TCL script, which synchronizes the timestamp of frames with that of ns-2 clock to generate the frames at a specified rate. The slices generated from a single frame will have same timestamp values. The algorithm aggregates the slice sizes to get length less than or equal to MTU size of 1500 bytes, which are sent to MyUDP agent that forms the UDP packets, generates sender trace files, and transmits them over the network at their respective timestamp values. The slice numbers and their corresponding size information present in each packet are saved in a file for future use. At the receiver side, the module MyUDPSink collects these packets and generates a receiver trace file which has the information about the received packet time. These two files are used for further evaluation as explained above.

Figure 4.28 shows the sender node which forms and transmits video fragments over the network. The pre-encoded prioritized H.264 video slices are attached to application layer, where the slices with same priorities are aggregated to form an MTU size video packet. The priority information of each video packet is specified in its header. These packets are segregated into different access categories (ACs) in IEEE 802.11e MAC layer according to their priorities. Each AC contends for channel access with others, and when it gets the transmission opportunity, it sends the packet to fragment burst module. This packet is divided by the fragment threshold value selected as per the priority of the packet. This threshold value is the optimal packet size, pre-computed using CBR data at which the goodput is maximum for a given channel BER and number of connections.

The fragmentation process can be explained as follows:

→ Given the video packet size is p bytes and the fragment threshold is x bytes

→ $if\ (p\ mod\ x)\ ==\ 0\ then,\ p/x\ =\ n\ fragments\ each\ of\ size\ x$
$else\ if\ (p\ mod\ x)\ >\ 0\ then,\quad p/x\ =\ n\ +\ 1\ fragments\ each\ of\ size\ x$
$except\ for\ the\ last\ one\ which\ will\ be\ (p\ mod\ x)$

At the receiver node shown in Figure 4.29, the fragments received are scanned for video slices present in them. The H.264 decoder is modified to perform partial slice decoding [49]. The video slices received completely are the ones successfully received. If there is a partially received video slice, it is taken into account only from the beginning of the header portion for the number of bytes received completely, even if the tail portion is lost. The video slice will be discarded completely even if one bit of the header portion is lost. When an intermediate fragment is in error, the fragments received before that are successfully decoded, and the remaining video is concealed at the decoder using concealment technique in [49]. Accordingly, the received video slices are listed in the receiver trace file, which is further sent to video decoder to evaluate the quality of video received, as discussed earlier.

Simulation Setup

In this scheme, the ad hoc LAN is setup to transmit H.264 video over a realistic error prone channel. The channel has known bit error rates, and the packet error rates are calculated for varying packet size values from 200 to 1500 bytes. These packet error rates are applied using error module in ns-2. The pre-encoded 256 kbps bit rate video slices are aggregated and formed into packets whose size vary from 200 to 1500 bytes, which are sent over the access categories in MAC layer according to their priorities. The video sequence has weights for each priority calculated using CMSE values. We calculate the goodput achieved for each priority individually, and then we calculate the weighted goodput according to weights given for a video sequence in the steady state condition of the network. The analysis is carried out for varying bit error rates and nodes to study the behavior of the network. We obtain an optimal packet size for each of the priority and nodes combination in the network for a given channel bit error rate which maximizes goodput. The parameters used in these simulations are shown in Table 4.8.

For prioritized retransmissions, we give more importance to the highest priority access category by giving more retry limit counter value and decrease the level of counter as we go to the lowest priority access category. This is to ensure that the highest priority gets more probability to transmit the packet with guarantee by attempting more number of times than the lower priority access categories. Also, we give prioritized timeout values for different access categories. The highest priority access category gets more time to allow more packets in the buffer to be transmitted without discarding them. The lower access categories get reduced timeout values, and hence we are giving more importance to the higher priority packets. The retry limits and timeout value given to

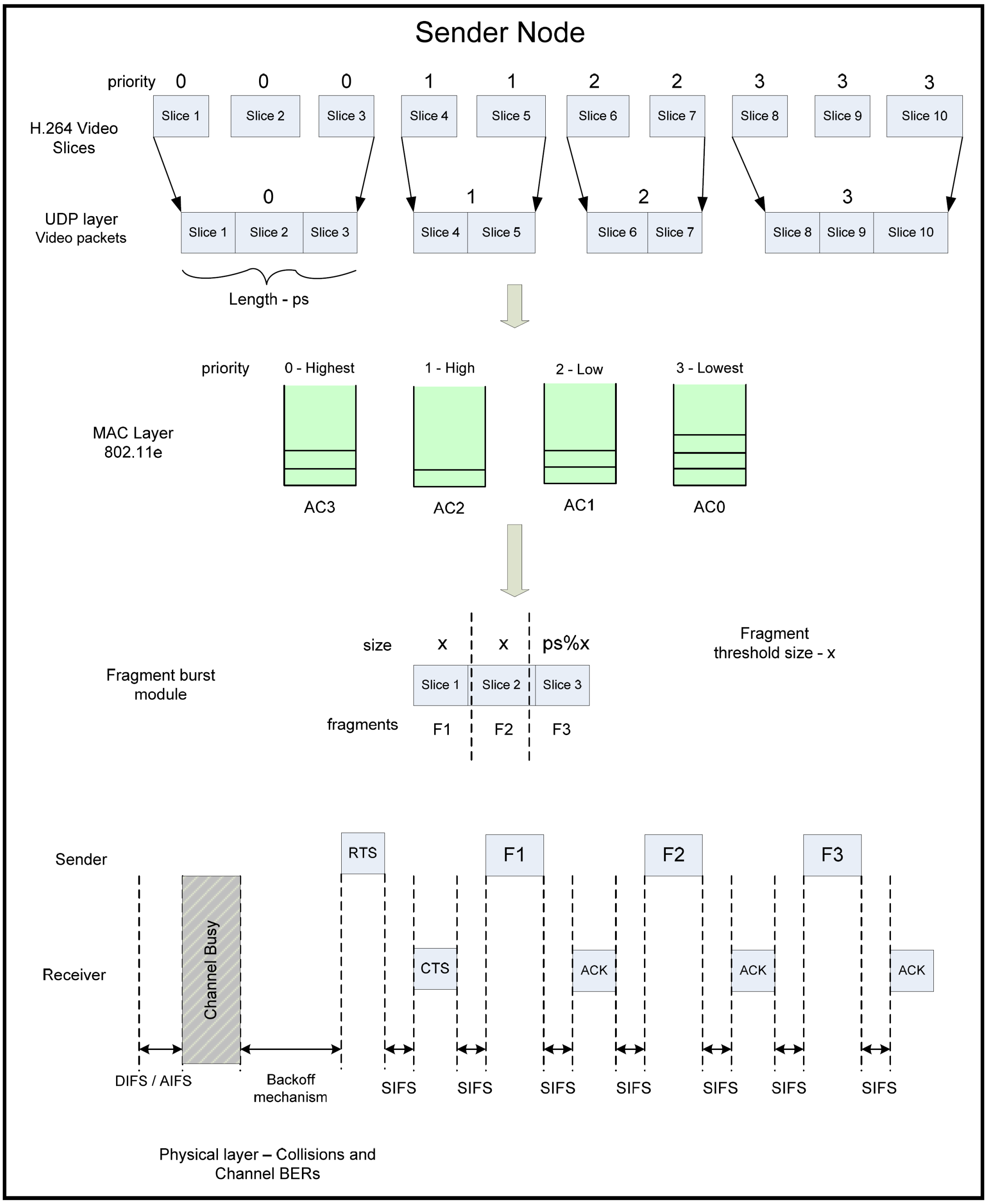


Figure 4.28: *H.264 video packet formation at application layer, mapping into access categories and fragmentation at MAC layer in a sender node in proposed scheme*

each access category is shown in Table 4.9. The output files from network are sent to H.264/AVC decoder which is modified to perform partial slice decoding, and we later calculate MSE and PSNR of the received video.

Purposed Scheme Results

The following plots show optimal goodput values obtained for ACs comparing between priorities of different nodes with variation in BER. The optimal goodput is obtained for each AC stream data at an optimal packet

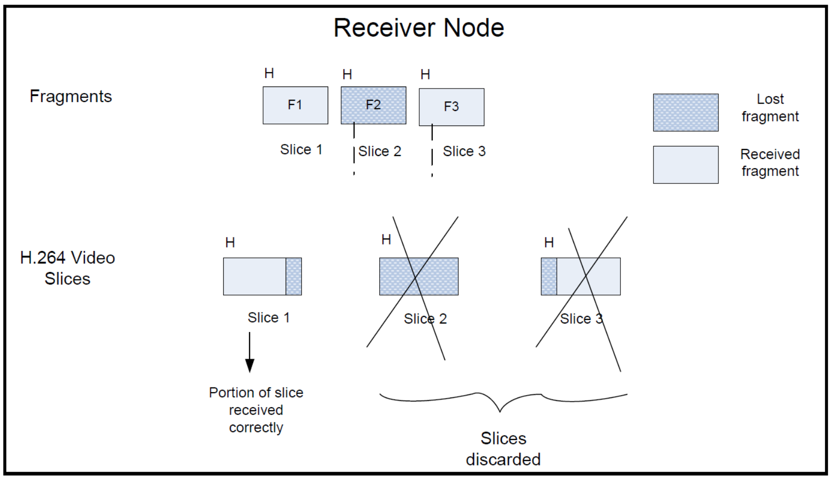


Figure 4.29: *Analysis of fragments received in a receiver node in proposed scheme*

Channel Bit Rate	2 Mbps
Video Bit Rate	256 kbps
RTS	44 bytes
CTS	38 bytes
ACK	38 bytes
SIFS	10 μs
DIFS	50 μs
MAC header	34 bytes
PHY header	24 bytes

Table 4.8: Simulation Parameters Used in Proposed Scheme

size while varying packet sizes from 200 to 1500 bytes. This is the point where maximum goodput is obtained for given number of connections and BER as explained in the optimal packet size scheme. These optimal goodput values calculated for each priority AC stream are plotted against number of connections in the figures that follow.

Each priority corresponds to the data sent through access category in IEEE 802.11e which contends with each other and with other nodes. We observe that the highest priority AC3 will get more opportunity to transmit data, and hence data sent over this stream will have the maximum goodput. The next chance of obtaining channel access will go to AC2, so its goodput will be higher than that of AC1 and AC0 streams. Then comes AC1 stream data with its goodput better than AC0.

In Figure 4.30, for BER=1 $\times$ 10^{-4}, the priority 4 data has the least goodput. It is around 50% when there are four connections in the network and drops to 10% when there are nine connections, whereas the priority 1 data decreases from 100% to 60% for nine connections. Overall, goodput for higher priority data is higher. We observe that when channel BER gets better, the goodput also improves because the probability of packets

Access Category	**Short Retry Limit**	**Long Retry Limit**	**Timeout value(ms)**
AC3-0 (Highest)	7	4	500
AC2-1 (High)	5	4	450
AC1-2 (Low)	4	4	400
AC0-3 (lowest)	2	2	250

Table 4.9: Retry Limits and Timeout Values for Four MAC Access Categories Used in the Proposed Scheme

being corrupted is lower.

In Figure 4.31, we observe that the priority 4 data has increased to 80% when there are four connections as compared to the previous Figure 4.30 but it is still 10% at nine connections because of high channel BER 5×10^{-5}.

Also in Figure 4.32, we observe that as channel BER goes lower, the optimal goodput obtained for each of the priorities is higher.

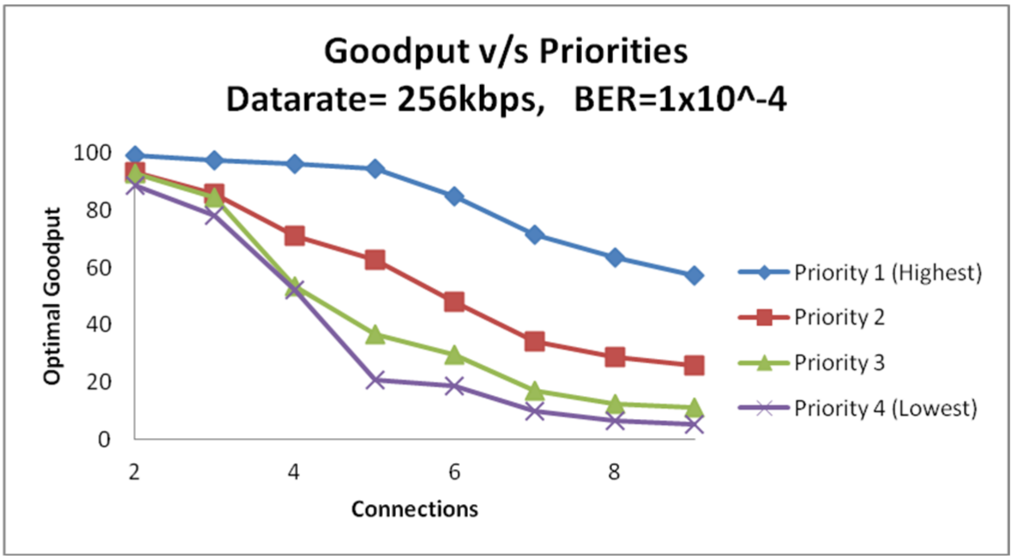


Figure 4.30: *Variation in optimal goodput with number of connections for four priorities for a data rate 256 kbps and BER* 1×10^{-4}

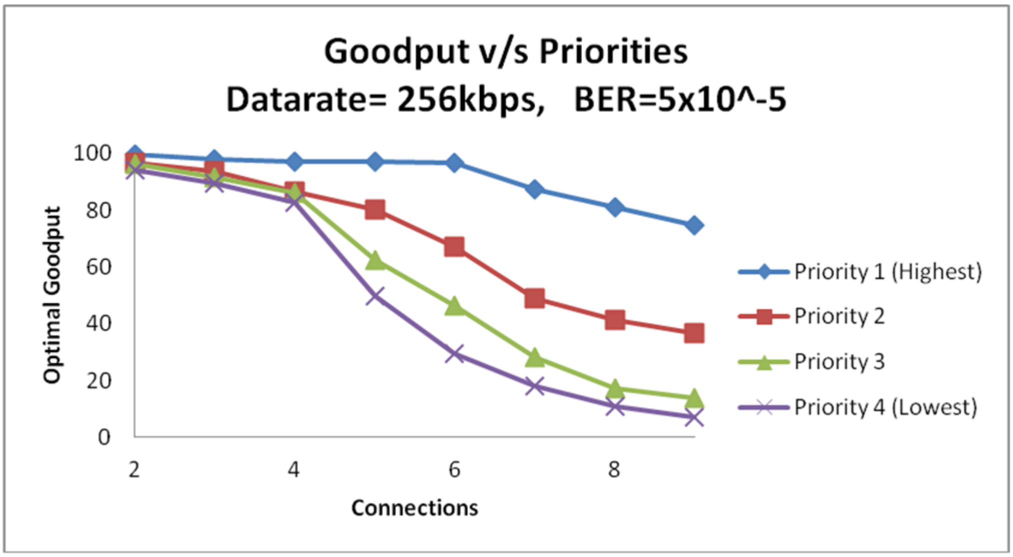


Figure 4.31: *Variation in optimal goodput with number of connections for four priorities for a data rate 256 kbps and BER* 1×10^{-5}

In Figure 4.33, we observe that for less than six connections, the goodput is almost the same for all the priorities because of very low channel BER. As the number of connections increases, the goodput drops drastically because the bandwidth available to transmit data gets lower.

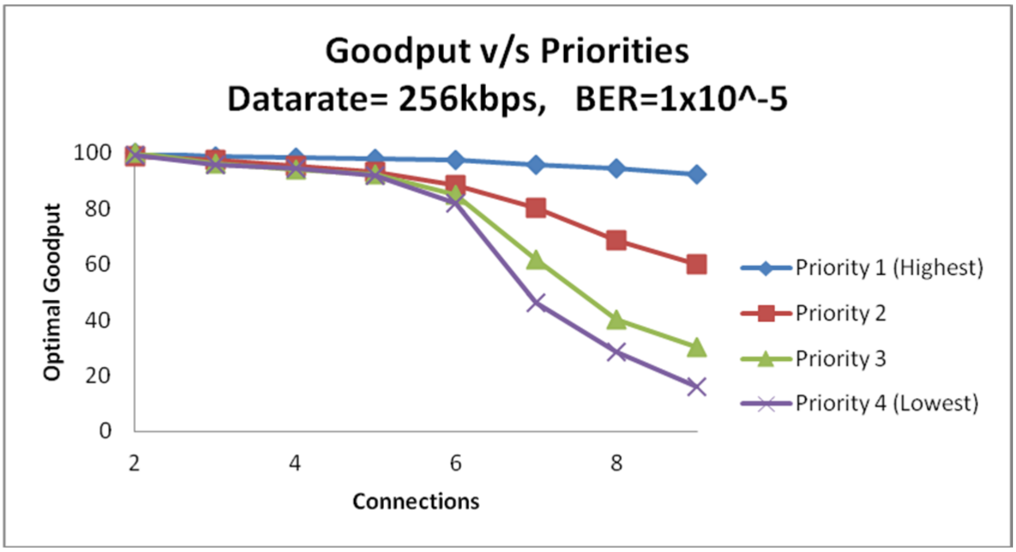


Figure 4.32: *Variation in optimal goodput with number of connections for four priorities for a data rate 256 kbps and BER* 1×10^{-5}

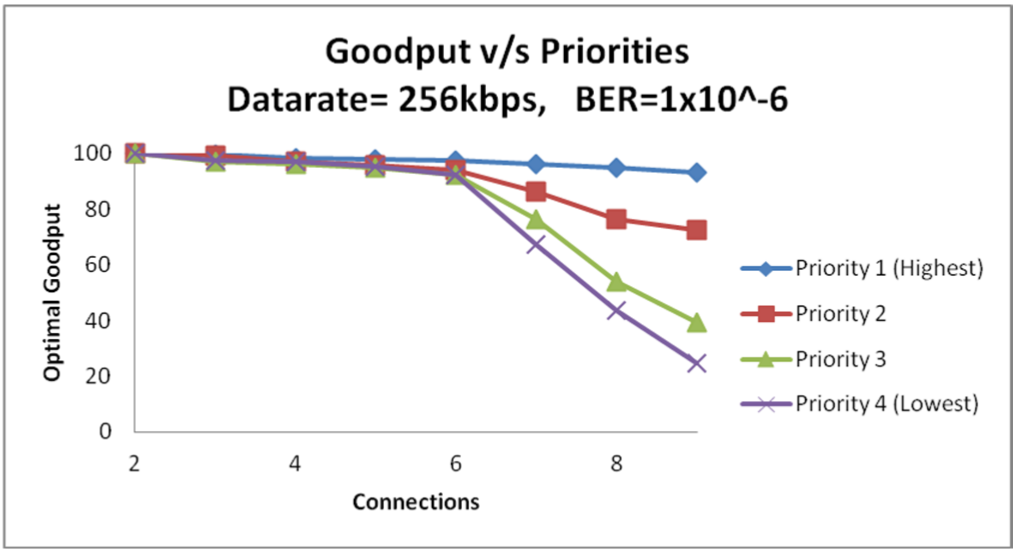


Figure 4.33: *Variation in optimal goodput with number of connections for four priorities for a data rate 256 kbps and BER* 1×10^{-6}

4.5.2 Performance Evaluation of Purposed Scheme

In this section, we compare the three schemes discussed in previous chapters. The proposed scheme is compared with the other two schemes, and performance is evaluated to show that the proposed scheme performs better than the other two. The weighted goodput calculated for the proposed scheme is plotted against the optimal goodput calculated for optimal packet size and baseline schemes. These comparisons are shown for data rates of 256 kbps and 512 kbps and four BERs. We observe an increase in the gain with increasing number of connections for a lower BER.

4.6 Comparison

Figure 4.34 shows the comparison plots for data rate 256 kbps and BER 1×10^{-6}. The weighted goodput decreases as the number of connections is increased for all schemes. However, the relative gain is higher with the proposed scheme than with other schemes for maximum connections.

The goodput obtained for optimal packet size scheme is nearly the same as that for the proposed scheme up to 6 connections. This is due to the fact that when the channel condition is good, the amount of data that could be successfully transmitted is high. Also, the fragment threshold values chosen for four priorities in the proposed scheme are almost the same as the optimal packet size obtained in the optimal packet size scheme because of low BER.

With more number of connections, the proposed scheme achieves higher gain because of the prioritization of data. Since the highest priority is given more weight in computation of the weighted goodput, it is boosted because of transmission opportunity obtained for higher priority data. The relative gain for the proposed scheme is higher than for the other schemes with higher channel BER even at fewer connections. This can be noticed in Figures 4.35, 4.36, 4.37.

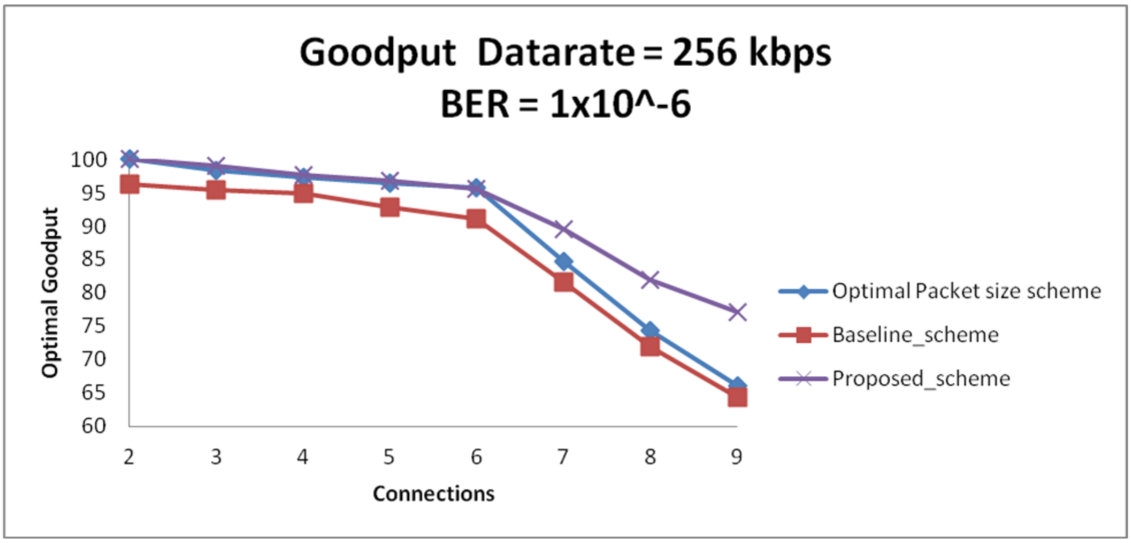


Figure 4.34: *Comparison of three schemes for data rate 256 kbps and BER*=1×10^{-6}

Similar gains are observed with higher data rate of 512 kbps in the proposed scheme as depicted in Figures4.38, 4.39, 4.40.

4.6.1 Summary

In this work we proposed a cross-layer priority aware packet fragmentation scheme to transmit pre-encoded H.264 video over bit rate limited error-prone wireless networks using IEEE 802.11e EDCA MAC protocol. Data fragmentation is done at MAC layer to adapt the packet size to varying channel conditions. We transmitted the UDP/CBR data by forming 200 to 1500 byte packet sizes and found an optimal packet size which maximizes the goodput for a given BER and number of connections. The optimal packet sizes were calculated for each priority and later used as fragment thresholds for transmitting data over each priority AC defined in IEEE 802.11e MAC protocol. We observed that the higher priority data used lower fragment threshold to achieve maximum goodput when the channel BER is high. The weighted goodput was calculated using pre-computed weights and the optimal goodput values obtained for each priority. It represented maximum goodput that could be achieved

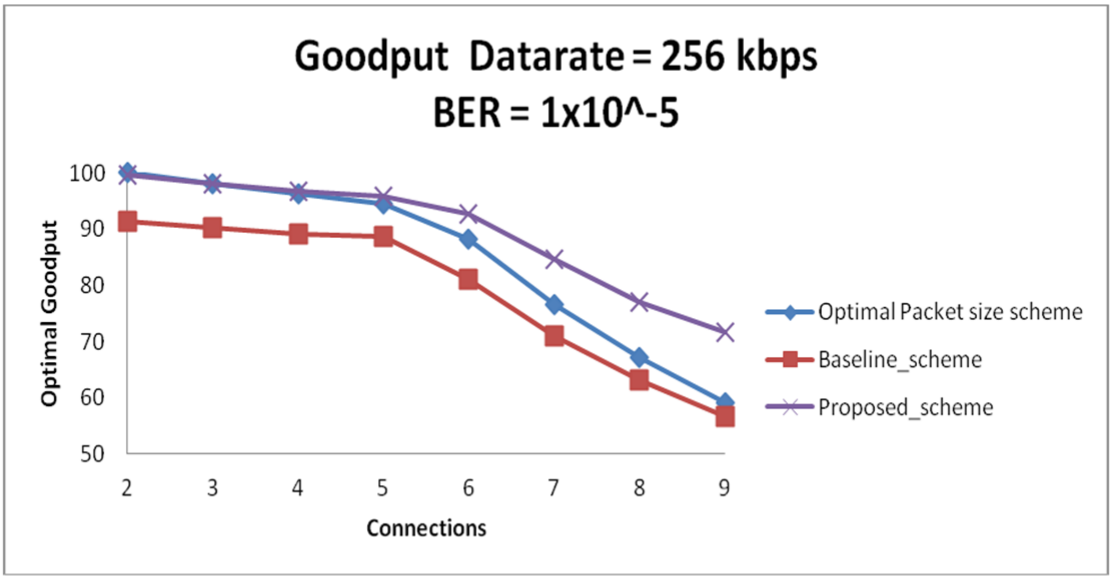


Figure 4.35: *Comparison of three schemes for data rate 256 kbps and BER*=1 × 10^{-5}

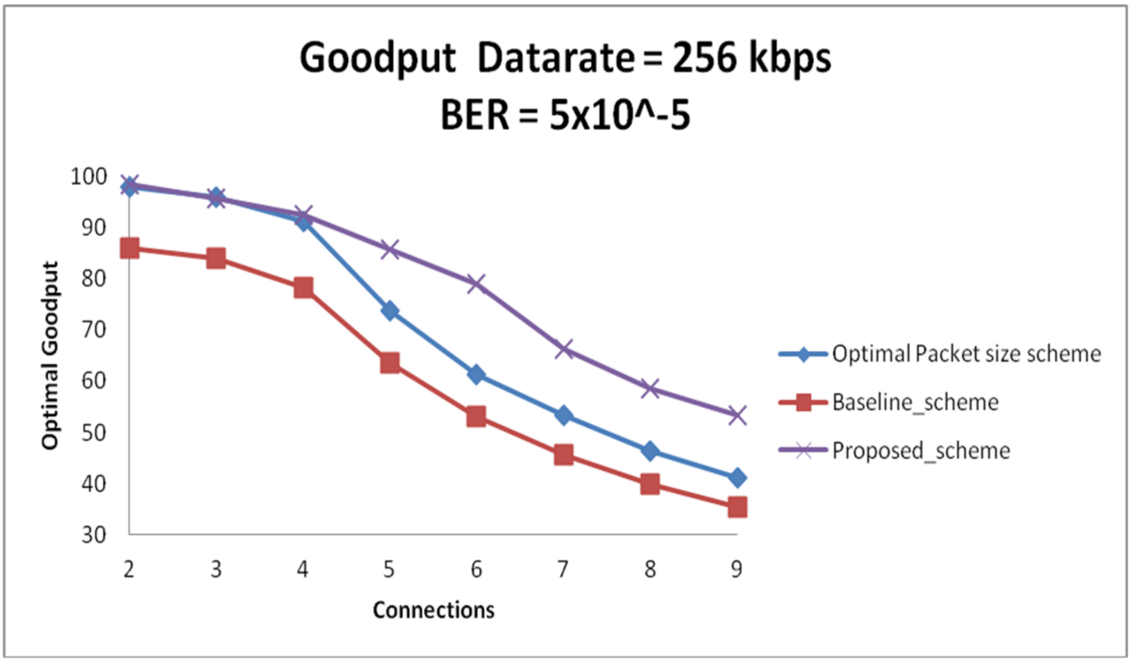


Figure 4.36: *Comparison of three schemes for data rate 256 kbps and BER*=5 × 10^{-5}

for a given number of connections and BER with a vector of four optimal packet size values obtained at four priorities.

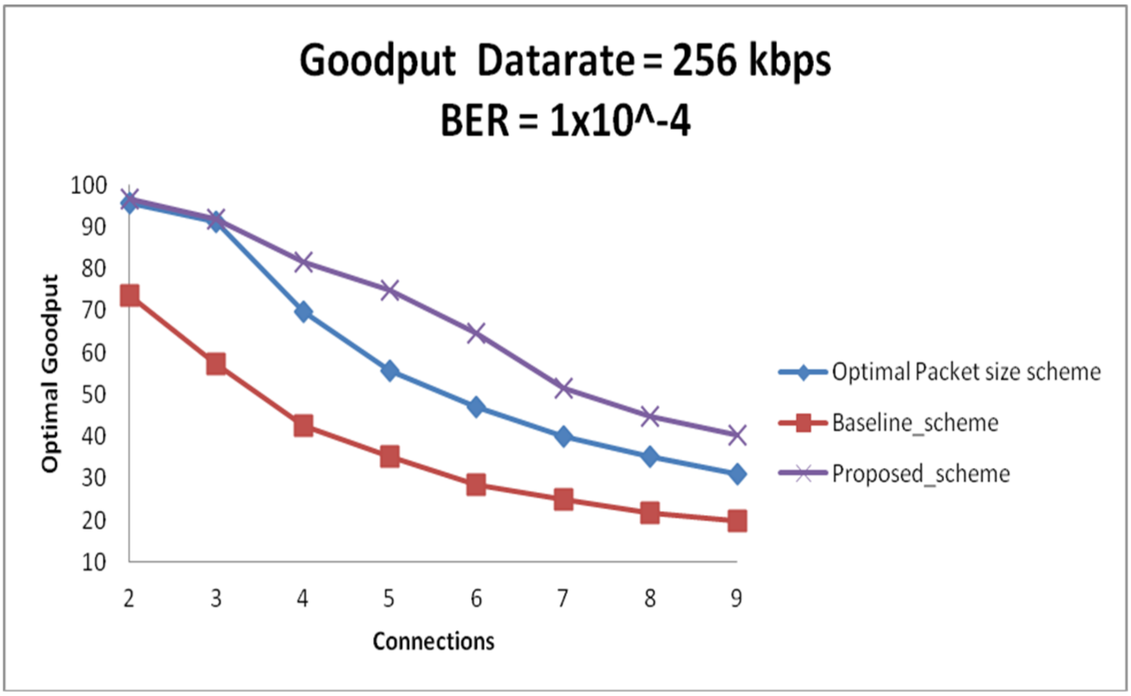


Figure 4.37: *Comparison of three schemes for data rate 256 kbps and BER*=1 × 10^{-4}

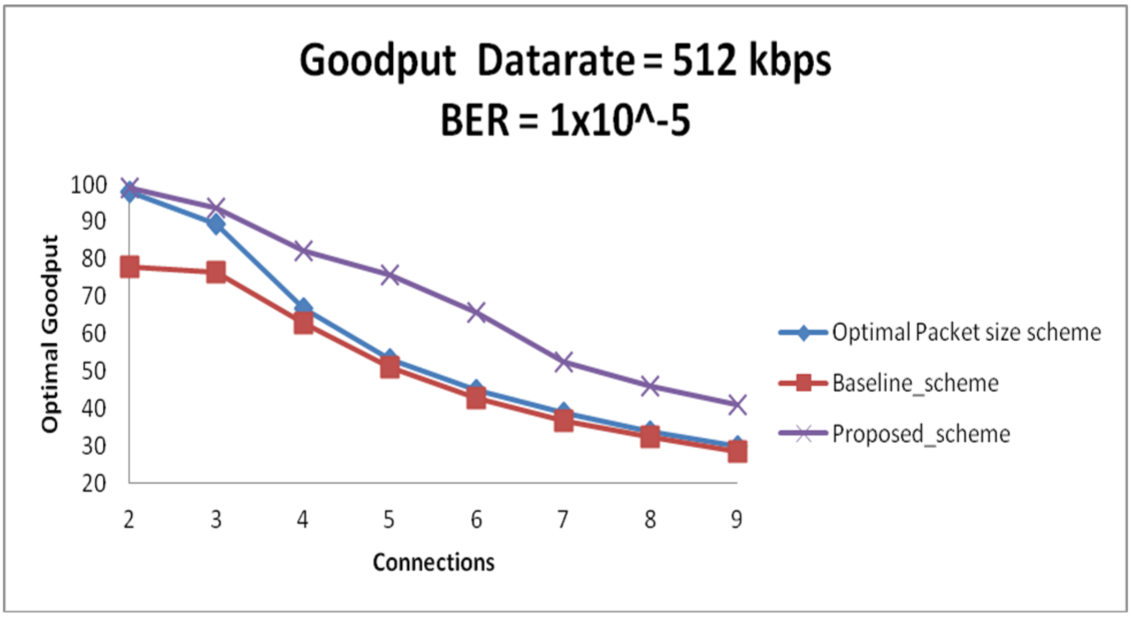


Figure 4.38: *Comparison of three schemes for data rate 512 kbps and BER*=1 × 10^{-5}

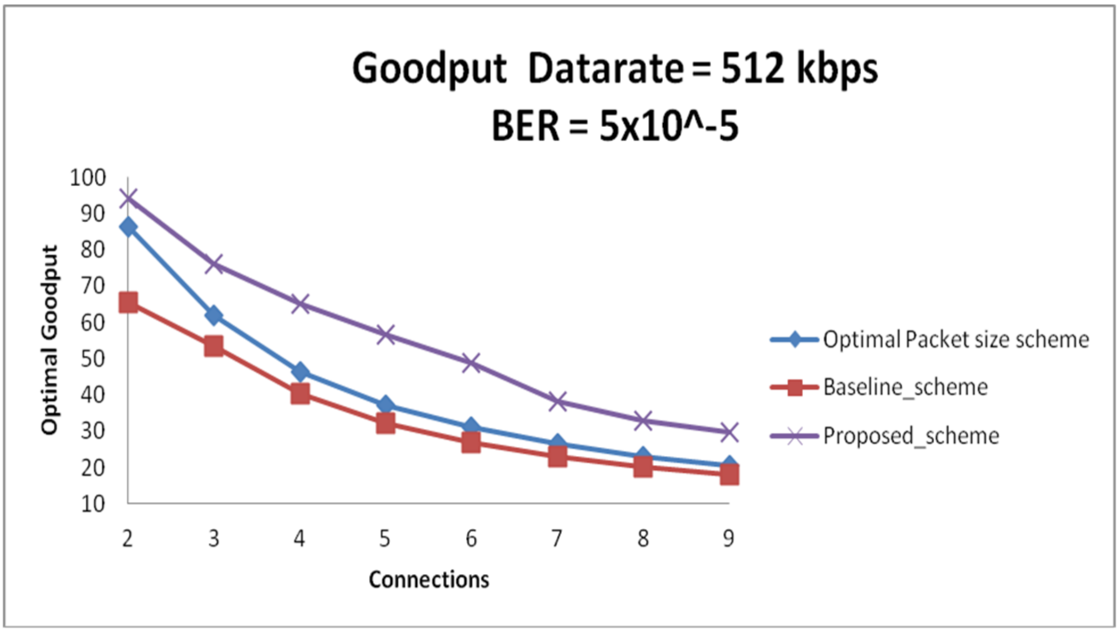


Figure 4.39: *Comparison of three schemes for data rate 512 kbps and BER*=5 × 10^{-5}

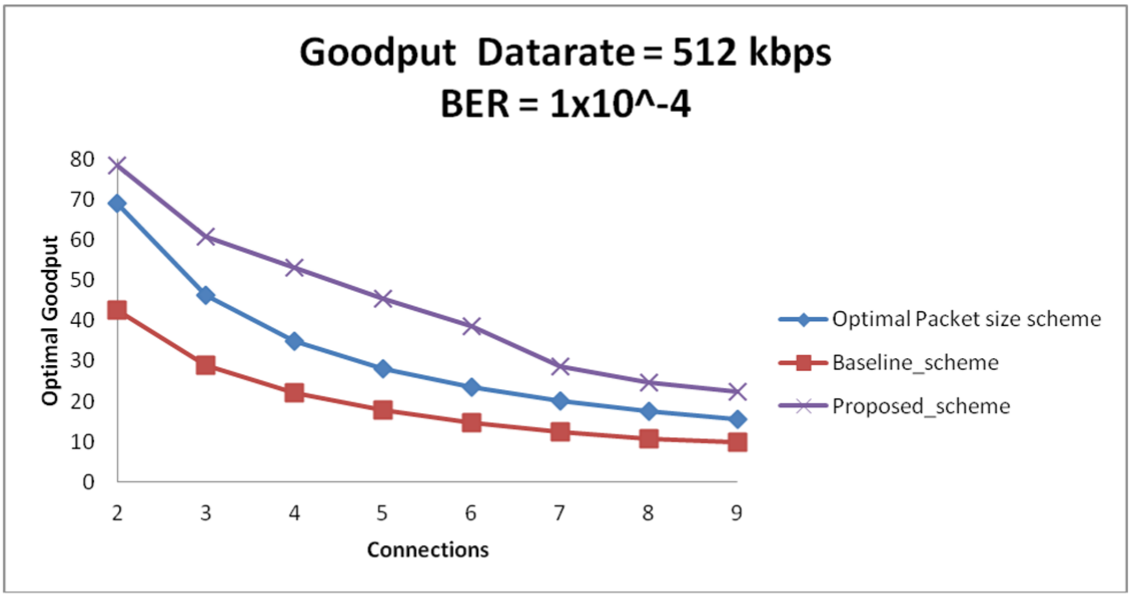


Figure 4.40: *Comparison of three schemes for data rate 512 kbps and BER*=1 × 10^{-4}

Chapter 5

Conclusion

Together with emergence of MANETs, new routing approaches are required since MANETs are highly dynamic and distributed in nature. The researches done have shown that nature inspired routing protocols can remove at least one or several problems in the area such as battery life, scalability, maintainability, survivability, adaptability and so on.

Through this work we tried to explore the potential of the nature-inspired networking in developing solutions to the wide array of challenges in routing and network management. We presented a literature survey about basics of multimedia data compression, introduction to H.264 and its architecture. Along with this we explored the different Swarm Intelligence inspired routing algorithm (mainly focused on Ant Colony). Undoubtedly, part of the ACOs popularity is due to the fact that it is a Nature-inspired meta-heuristic with a quite simple and modular structure based on the use of ant-like agents and stigmergy. That is, an optimization meta-heuristic which promises to generate optimal or near-optimal solutions on the basis of a recipe made of a set of possibly simple and computationally light agents, that independently and repeatedly construct solutions according to a stochastic decision policy locally dependent on pheromone variables. Part of the appeal of ACO precisely comes from this expectation of generating extremely good solutions out of the simplicity of the main actors (the ant-like agents) and from the collective and fully distributed learning activities in which they are involved in, similarly to what happens in the case of real ant colonies.

Based on the principles of basic ant colony algorithm we proposed a new tree-based ant colony algorithm and apply it to solving multicast routing problem. With only one ant the new algorithm can directly find a multicast tree on the network, overcoming drawbacks of basic ant colony algorithm in multicast routing such as the misleading of pheromones and repetitive search of paths, and speeding convergence of the algorithm. Experimental results indicate that in solving multicast routing the new algorithm has higher convergence speed, less consumption of time than, and similar performance of finding the best solution to the basic ant colony algorithm. The new algorithm, however, is aimed at multicast routing. We also proposed a cross-layer priority aware packet fragmentation scheme to handle H.264 video transmission over error-prone wireless networks using IEEE 802.11e EDCA MAC protocol. Packet fragmentation is done at MAC layer to adapt the packet size to varying channel conditions. The fragmentation is combined with the prioritization on the basis of Access Classes. The optimal packet sizes were calculated for each priority and later used as fragment thresholds for transmitting data over each priority AC defined in IEEE 802.11e MAC protocol. We observed that the higher priority data used lower fragment threshold to achieve maximum goodput when the channel BER is high.

Chapter 6

Bibliography

Bibliography

[1] Nukhet Ozbek, Turhan Tunali",A Survey on the H.264/AVC Standard", "Turk J Elec Engin", 13,pp.288-290,2005

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra."Overview of the H.264/AVC Video Coding Standard" In IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 560-576, 2003

[3] N.S.Narkhede,Naveeta Kant, "The Emerging H.264 /Advanced Video Coding Standard And Its Applications",In International Conference on Advances in Computing, Communication and Control, Vol. 12, Jan, 2009

[4] Narkhede, N. S. and Kant, Naveeta, "The emerging H.264/advanced video coding standard and its applications",In Proceedings of the International Conference on Advances in Computing, Communication and Control,Jan 2009

[5] Muhammad Saleem, Gianni A. Di Caro, Muddassar Farooq",Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions",In Information Science Elesvier,May 2010

[6] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra",Overview of the H.264/AVC Video Coding Standard",IEEE Transactions on Circuits and Systems for Video Technology,2003

[7] I. E. G. Richardson, H.264 and MPEG-4 Video Compression, UK Wiley, 2003.

[8] Ralf Schfer, Thomas Wiegand and Heiko Schwarz, "The emerging H.264/AVC standard" EBU TECHNICAL REVIEW January 2007 Heinrich Hertz Institute, Berlin,Germany

[9] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264 /MPEG4 Advanced Video Coding standard and its applications", IEEE Communications Magazine, vol. 44, no.8, pp. 134-144, Aug. 2006

[10] P. Xue and S. Chandra, "Revisiting multimedia streaming in mobile ad hoc networks, " in Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video, pp. 1-7, 2006.

[11] M. Tarique, K. E. Tepe, S. Adibi, and S. Erfani, "Survey of multipath routing protocols for mobile ad hoc networks", Journal of Network and Computer Applications, vol. 32, no. 6, pp. 1125 - 1143, 2009

[12] Muhammad Shaffatul Islam, Md. Adnan Riaz, Mohammed Tarique",Performance Analysis Of The Routing Protocols For Video Streaming Over Mobile Ad Hoc Networks",International Journal of Computer Networks and Communications (IJCNC) Vol.4, No.3, May 2012

[13] Anurag Malik, Shivanshu Rastogi, Sajendra Kumar, ”Performance Analysis of Routing Protocol in MobileAd Hoc Network using NS-2. ” MIT International Journal of Computer Science and Information Technology, January 2011, Vol. 1 No. 1, pp. 47-50.

[14] B. Baran and R. Sosa, ”A new approach for AntNet routing” , presented at the Proc. 9th Int. Conf. Computer Communications Networks, Las Vegas, NV, 2000

[15] Tarek Sheltami”,Performance Evaluation of H.264 Protocol in Ad hoc Networks,Journal of Mobile Multimedia,Vol.4,No.1, 2008 pp. 59-70

[16] E. Bonabeau, M. Dorigo, and G. Theraulaz, ”Inspiration for optimization from social insect behaviour”, Nature, vol. 406, pp. 3942, July 2000

[17] H.264 Advanced Video Coding, http://www.vcodex.com/h264.html.

[18] H.264/MPEG-4 AVC, http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC

[19] J. Baras, H. Mehta,A Probabilistic Emergent Routing Algorithm for Mobile Ad hoc Networks (PERA), 2003

[20] J. Klaue, B. Rathke, and A. Wolisz, ”EvalVid - A Framework for Video Transmission and Quality Evaluation”, in Proceedings of the 13th Conference on Modelling Techniques and Tools for Computer Performance Evaluation, ser. Lecture Notes in Computer Science. Springer, 2003, pp. 255272.

[21] Daniel Hertrich, ”Mpeg4 video transmission in wireless lans basic qos support on the data link layer of 802.11b” Minor Thesis, 2002.

[22] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, ”Overview of the H.264/AVC video coding standard”, IEEE Trans. Circuits Syst. Video Technol.,vol. 13, no. 7, pp. 560-576, Jul. 2003.

[23] LAN MAN Standards Committee of the IEEE Computer Society, Sponsor,ANSI/IEEE Std 802.11, 1999 Edition: Part 11. Piscataway, NJ: IEEE, 1999.

[24] T. Stockhammer and M. M. Hannuksela, ”H.264/AVC video for wireless transmission”, Wireless Communications, vol. 12, no. 4, pp. 6-13, Aug. 2005.

[25] ”ns2.” [Online]. Available: http://www.isi.edu/nsnam/ns/

[26] ”802.11-2012 IEEE Standard for Information Technology, Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks, Specific Requirements Pt 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.” [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6178209

[27] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, ”IEEE 802.11e wireless LAN for quality of service”, in Proc. European Wireless, Feb. 2002, pp. 32-39.

[28] LAN MAN Standards Committee of the IEEE Computer Society, Sponsor, ANSI/IEEE Std 802.11D, 1998 Edition: Part 11. Piscataway, NJ: IEEE, 1998.

[29] ”IEEE 802.11e QoS on WLANs.” [Online]. Available: http://www.docstoc.com/docs/84960443/IEEE-80211e-QoS-on-WLANs

[30] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", IEEE Journal Selected Areas in Communications, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[31] J. Yin, X. Wang, and D. P. Agrawal, "Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function", in Proc. IEEE WCNC, Mar. 2004, pp. 1654-1659.

[32] B. S. Kim, Y. Fang, and T. F. Wong, "Throughput enhancement through dynamic fragmentation in wireless LANs", IEEE Trans. on Veh. Technol., vol. 54, no. 4, pp. 1415-1425, Jul. 2005.

[33] F. Zheng and J. Nelson, "Cross-layer adaptive design for the frame length of IEEE 802.11 networks", in IEEE, WiOPT, Apr. 2008, pp. 437-442.

[34] P. R. Jelenkovic and J. Tian, "Dynamic packet fragmentation for wireless channels with failures", in ACM MobiHoc, May 2008, pp. 73-82.

[35] Y. Chang, C. Lee, and J. A. Copeland, "Dynamic optimal fragmentation for goodput enhancement in WLANs", in Proc. TridentCom, Orlando, Florida, May 2007.

[36] A. Ksentini and M. Naimi, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross layer architecture", IEEE Commun. Mag., vol. 44,no. 1, pp. 107-114, Jan. 2006.

[37] A. Ksentini, M. Naimi, and A. Gueroui, "Improving H.264 video transmission in 802.11e EDCA", in Proc. ICCCN, Oct. 2005, pp. 381-386.

[38] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard", J. Visual Commun. and Image Representation, vol. 17, no. 2,pp. 183-185, Apr. 2006.

[39] Y. P. Fallah, P. Nasiopoulos, and H. Alnuweiri, "Efficient transmission of H.264 video over multirate IEEE 802.11e WLANs", EURASIP J Wireless Commun. and Networking, vol. 2008, no. 11, pp. 1-14, Jan. 2008.

[40] Y. P. Fallah, K. Darrell, S. Avideh, K. Faizal, and P. Nasiopoulos, "A cross layer optimization mechanism to improve H.264 video transmission over WLANs", in IEEE, CCNC, Jan. 2007, pp. 875-879.

[41] A. T. Connie, P. Nasiopoulos, V. C. M. Leung, and Y. P. Fallah, "Video packetization techniques for enhancing H.264 video transmission over 3G networks", in IEEE, CCNC, Jan. 2008, pp. 800-804.

[42] R. MacKenzie, D. Hands, and T. OFarrell, "An evaluation of quality of service for H.264 over 802.11e WLANs", in Proc. 17th Intl. PV 2009, Seattle, Washington,May 2009.

[43] K. K. R Kambhatla, S. Kumar, S. Paluri, and P. C. Cosman, "Wireless H.264 video quality enhancement through optimal prioritized packet fragmentation", IEEE Trans.on Multimedia, vol. PP, no. 99, p. 1, Apr. 2012.

[44] S. Kumar, K. K. R Kambhatla, and P. C. Cosman, "Prioritized packet fragmentation for H.264 video", in IEEE Int. Conf. on Image Processing, Sep. 2011, pp. 3233-3236.

[45] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming", Wireless Commun., vol. 12, no. 4, pp. 59-65,Aug. 2005.

[46] S. Shakkotai, T. Rappaport, and P. Karlsson, "Cross-layer design for wireless networks", IEEE Commun. Mag., vol. 41, no. 10, pp. 74-80, Oct. 2003.

[47] M. van der Schaar and N. S. Shankar, ”Cross-layer wireless multimedia transmission:challenges, principles and new paradigms”, IEEE Wireless Commun. Mag., vol. 12,no. 4, pp. 50-58, Aug. 2005.

[48] J. Klaue, B. Rathke, and A. Wolisz, ”EvalVid: A framework for video transmission and quality evaluation”, in Proc. Int. Conf. on Modeling Techniques and Tools for Comp. Perf. Evaluation, Sep. 2003, pp. 255-272.

[49] S. Kumar, A. Janarthanan, M. M. Shakeel, S. Maroo, J. D. Matyjas, and M. Medley”,Robust H.264/AVC video coding with priority classification, adaptive NALU size and fragmentation”, in IEEE, MILCOM, Jun. 2009, pp. 1702-1707.

[50] T. Wedi, ”Motion compensation in H.264/AVC”, IEEE Trans. Circuits Systems Video Technology., vol. 13, pp. 577586, July 2003.

[51] S.K. Bisoyi,S. Sahu, ”Performance analysis of Dynamic MANET On demand (DYMO) Routing protocol”, IJCCT Vol.1 Issue 2, 3, 4; 2010 International Conference ACCTA-2010, 3-5 August 2010.