|  |
| --- |
| LHC-CT-ES-0004 rev 0.1 AB-OP 780566  2006-10-02    **CERN**  CH-1211 Geneva 23 Switzerland  the **Large Hadron Collider** project |
| Functional Specification |
| VideO STREAM NETWORK SIMULATION  Design Specifications |
| Abstract  *In order to efficiently use video stream application over a wireless network a novel scheduling algorithm has been developed, The performance of this algorithm will be evaluated by simulation. The simulator models Wi-Fi communication in which the nodes are capable of synchronous hopping between frequency channels. The simulator is developed using OMNeT++.*  Yaniv Fais  Version 0.1 – 15-May-2010 [Draft] |
|  |

**Table of Contents**

[1. IntroductioN 3](#_Toc261734369)

[1.1. Document Outline 3](#_Toc261734370)

[1.2. Document Description 3](#_Toc261734371)

[1.2.1. Introduction 3](#_Toc261734372)

[1.2.2. System Overview 4](#_Toc261734373)

[2. Design Considerations 4](#_Toc261734374)

[2.1. Assumptions and Dependencies 4](#_Toc261734375)

[2.2. General Constraints 5](#_Toc261734376)

[2.3. Goals and Guidelines 5](#_Toc261734377)

[2.4. Development Methods 5](#_Toc261734378)

[3. Architectural Strategies 6](#_Toc261734379)

[4. System Architecture 7](#_Toc261734380)

[5. Policies and Tactics 8](#_Toc261734381)

[6. Detailed System Design 9](#_Toc261734382)

[7. Glossary 10](#_Toc261734383)

[7.1.1. Mobile Ad Hoc Networks 10](#_Toc261734384)

[7.1.2. Video Stream: 10](#_Toc261734385)

[7.1.3. IEEE 802.11 : 10](#_Toc261734386)

[7.1.4. WiFi: 10](#_Toc261734387)

[7.1.5. MPEG: 10](#_Toc261734388)

[7.1.6. OMNET++: 10](#_Toc261734389)

[7.1.7. Mixim: 11](#_Toc261734390)

[8. Bibliography 11](#_Toc261734391)

# Introduction

A novel scheduling algorithm [1] has been developed in order to efficiently use a wireless network for streaming video transmission over the network, an OMENT++ [2] simulation model will be used to test and quantify this algorithm, this document would serve as a design specification for this network simulator subsystem.

## Document Outline

The following document will describe the wireless network simulation application, it will include a description of the application goals, system and simulation model as well as design details.

## Document Description

This document serves as a high level design for the network simulation sub-project of the video stream project. The document can serve for definition and overview of the project interfaces and goals. The design document is to be reviewed prior to implementation and then finalized and serve as an implementation guide for the project.

### Introduction

Wireless networks have become very popular in recent years through the IEEE 802.11 (Wi-Fi) standard and they are now very common in many devices for many applications.

Ad Hoc networks enable formation of changing network, while those networks are very robust they suffer from bandwidth problem due to interference from neighboring nodes.

Network traffic can be generally characterized into priorities according to band width requirements and delay considerations , while some traffic (such as email ) usually requires low bandwidth and can complete in some future time other forms of traffic such as video streams require both high bandwidth and require real-time performance thus highly effected by network congestion.

A novel scheduling algorithm has been developed in order to enable usage of video streams over wireless networks, the scope of this project is to use this algorithm and simulate its network video traffic using a network simulator and test the results in a video player.

### System Overview

Video File

Video Player

Wireless Network

Simulation

Scenario generator

Scheduler

The project system is composed of a scenario generator, a scheduling algorithm and network simulator, The first two components are external and are considered inputs to the simulator.

The output of the simulator is the actual network traffic, this output can be used to compose a video from input video stream traffic, and this video can be then played using a video player.

# Design Considerations

## Assumptions and Dependencies

In order to divide and distribute the video networking usage several assumptions need to be made:

* The main problem of network traffic is for "real-time" video data information, therefore control information such as nodes requirements and network formation are considered low bandwidth and can be synchronized in low frequency, we thus assume the network and loads are in steady state and this state change is ignored. (The synchronization is available through accurate clock in each node due to GPS availability, we assume clock skew is negligible).
* The traffic can be characterized by video, audio and data , we focus on video of MPEG H.264 type which can be partitioned into two types of frames: I and P… [TODO: need to clarify this after follow-up meeting]
* The network simulator is dependent on the existence of the scenario generator and the scheduler applications to provide inputs.
* The network simulator is dependent on OMNET++ and its wireless model (Mixim [3]) accuracy and speed.

## General Constraints

Processing video streams in a large network is done by several parallel and fast processors (CPU's,WiFi cards and etc..), a simulation model is used in order to simulate the entire system on a single processor in a serial way using an event based simulation algorithm, this means the performance of the simulation would be much slower than the real system.

In the real system the video should be viewed in real-time however in the simulation environment a single video can be processed , collected and later viewed, since only the viewing part is required to be in "real-time" the simulation can be much longer however to have meaningful results we need to have some reasonable ratio such that it may take hours/days to transfer seconds/minutes of video (in other words the transmission of several frames in a simulation wall clock month time isn't reasonable).

## Goals and Guidelines

The goal of the simulator is to enable quantifying the capabilities of the scheduling algorithm and in different scenarios, in order to enable this the system has been partitioned into a scenario generator and a scheduler application which provide inputs to the simulator, the simulator can then be used in order to check the network traffic according to the scenario and scheduling and accumulate a video stream in the destination nodes, this video stream can then be viewed by a video player, since video players employ advanced techniques to compensate on poor input data quality the subjective viewing of the video will be used to measure to effectiveness of the scheduler in the given scenario.

## Development Methods

The development would be done using the free and open-source network simulator OMNET++ and its wireless model Mixim, the video processing would be done using the free open-source Video LAN Player (VLC) [4].

The above components are publicly available free of charge for academic usage and enable easy adoption for multiple developers and are mostly suitable for academic research purposes.

The network simulator has been used by several research projects and its documentation and support is available publicly [5], several projects usage results has also been published [6].

The network simulator developments components can be added using C++ code to enable additional capabilities, this environment can be used in Linux or Windows systems.

The network simulator has a rich GUI development and run environment as well as statistics processing capabilities.

# Architectural Strategies

In order to test the system in different scenarios and for development to be done by different groups the system was decomposed into the scenario generator and the scheduler who feed the network simulator.

This architecture should enable the testing the system in different scenarios, the network simulator itself needs to allow flexibility in scenario building and network model.

The network model needs to be flexible in order to enable additional network flow control algorithms on nodes.

The simulation would be done in two modes – real video processing and generic data processing, in case a real video is processed it should be provide as input and in case a generic data is processed the transmitted data is assumed to be arbitrary, in this case the output would only be the rate statistics

# System Architecture

The network simulator system is partitioned into several components according to input/output processing and common network layers modeling.

Video

Stream

Accumulation

Application Layer

Input Video Stream

Mac Layer

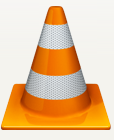
Scheduler

Phy Layer

Input Reader

Scenario Generator

Statistics processing

Video Player

1. Input scenario reader

The input of the simulator is the scenario generator and the scheduling table.

The scenario generator should provide parameters to the network model whereas the scheduler should provide the transmission table to nodes.

The input scenario should use reader from parameter files similar to the scheduler where possible, if not possible the input reader should generate the OMNET++ parameters and network configuration.

1. Optional input video reader

If the simulation is done in real video processing mode then an input video would be provided, in this case the video would be converted into packets and be correlated to the scenario and schedule table

1. Network Physical (PHY) Layer and Analogue Model

As described in [7] the Phy and Analogue Models are responsible for simulating the attenuation (like shadowing, fading and path loss) of a received signal. The Radio module simulates the physical characteristics of the radio, like switching times and simplex / duplex capabilities and the Decider is responsible for evaluation (classification as noise or signal) and demodulation (bit error calculation) of the received messages. Additionally, it provides channel sensing information to the MAC layer. These models will implement the specific path loss model as decided in the project definition and a decider based on SINR model , implement BER rate according to modulation and provides statistical information such as reception SINR and times.

1. Network Media Access Control (MAC) Layer

The MAC layer would simulate the 802.11 media access control protocol, this includes carrier sensing, request to send and etc.

As this protocol doesn't employ flow control additional enhancements to this layer may be needed in order to add flow control to the network if needed.

1. Video Application Layer

The application layer should implement:

* 1. receive and accumulate of packets to compose a video frame
  2. scheduled send according to table of video data

In case the application mode is not real video then simple data would be send and receive according to defined bandwidth

1. Output video stream accumulation

Once video frames are accumulated in a receiver node application it would be composed as an input stream (time+data) in a format suitable for the video player

1. Statistics processing

Simulation results would be saved to file and later processes to report summary tables and graphs of needed statistics

# Policies and Tactics

OMNET++ statistics data format would be used to save any statistics of the network simulation, this output textual data format would then be processed in order to create summary tables and graphs.

The OMNET++ scavetool would be used in order to convert the data to CSV format or matlab in order to be processed.

The statistics which is required is:

* 1. Obtained bandwidth vs. required bandwidth per stream (Application Layer).
  2. Actual queue length per stream (Application Layer).
  3. Packet error rate per stream per node (Channel Layer). Every failure is labeled with the cause of the failure, i.e.,
     1. Due to interference (*P*/*N* ok but *P*/*(N+I)* too small).
     2. Due to low reception power (*P*/*N* too small).
  4. Packet delay from end to end (Channel Layer).
  5. Jitter buffer delay (Application Layer).
  6. "Black Screen" time as a function of the jitter buffer length (Application Layer).

# Detailed System Design

1. Input scenario reader
   1. NedCreator - class used for creating an input OMNET++ configuration file of type .ned

This class would parse scenario input files and convert it to the needed .ned format

1. Input video reader
   1. VideoReader – class responsible for reading video and forming data packets from them, these packets would be formed into queue of data which would then be used by the node application layer for sending
2. Phy Layer
   1. VidStreamPhyLayer – wrapping class for analogue models and path loss model,derived from 80211PhyLayer
   2. VidStreamDecider – class used to decide whether a Mixim AirFrame can be used to receive a packet or considered as noise
   3. VidStreamPathLossModel – class implementing the attenuation formula of the wireless network
3. Mac Layer
   1. VidStreamMac – class implementing MAC layer , derived from 80211MacLayer

Additional packet types would be added and the necessary receive frame methods would be overloaded if flow control implementation would be needed according to queue size

1. Application Layer
   1. VidStreamApplication – class implementing the video application
      1. This class would read the scheduler table, each node according to its ID would send if needed a packet according to the video stream data as filled by the video reader or route according to received packets
2. Video stream accumulator
   1. VidStreamCreator – a class responsible for creating a stream (data/time) format for the video player according to destination application received WiFi packets
3. Statistics processing – OMNET++ data format would be used for saving statistics of simulation, OMNET++ scavetool would be used for converting this format to CSV or MATLAB format where the summary statistics would be processesed

This requires an additional external component – possible a Perl or Matlab script

# Glossary

### Mobile Ad Hoc Networks

A mobile ad hoc network is a self configuring network of mobile devices connected by wireless links. Each device (called node) may act as an independent router and be free to move independently in any direction. Therefore, it allows reconfiguration around broken or blocked paths.

The node mobility causes a frequently change of transfer path.

### Video Stream:

Live video presented to the end-user, while being delivered by a streaming provider.

### IEEE 802.11 :

IEEE organization's set of standards for wireless local area network (WLAN) computer communication. Most popular standards are: 802.11a, 802.11b, 802.11g, 802.11n.

### WiFi:

Wi-Fi is a trademark owned by the Wi-Fi Alliance (WFA) trade group. Wi-Fi networks use the IEEE 802.11 standard for providing secure, reliable and fast wirless connectivity.

### MPEG:

The Moving Picture Experts Group (MPEG) is ISO's standards for audio and video compression and transmission. (Video compression is used for reducing the quantity of data used to represent digital video images.)

### OMNET++:

OMNET++ is a discrete event simulation environment. Its primary application area is the simulation of communication networks.

Although OMNET++ is not a network simulator itself, it is currently gaining widespread popularity as a network simulation platform in the scientific community as well as in industrial settings, and building up a large user community.

One of its strengths is being a currently developed open source environment, allowing the team to take part in the future development of the environment.

### Mixim:

MiXiM (mixed simulator) is a simulation framework for wireless and mobile networks using the [OMNeT++ simulation engine.](http://www.omnetpp.org/) Mixim Framework shall be used for IEEE 802.11 and complex routing implementations.

# Bibliography

* 1. Frequency and Time Slot Assignment Algorithm Report By Guy Even, Moni Shahar, Alexander Zadorojniy.

[2] OMENT++ (<http://www.omnepp.org>)

[3] MIXIM (<http://mixim.sourceforge.net/>)

[4] VLC – (<http://www.videolan.org>)

[5] OMNET++ support group (<http://groups.google.com/group/omnetpp>)

[6] OMENT++ projects publications – ([http:://www.omnet-workshop.org](NULL))

[7] MiXiM – The Physical Layer An Architecture Overview (Karl Wessel, Michael Swigulski, Andreas Köpke, Daniel Willkomm - Technische Universität Berlin, Telecommunication Networks Group - Berlin, Germany)

[8] - Simulating Wireless and Mobile Networks in OMNeT++ - The MiXiM Vision

(A. Köpke, M. Swigulski,K. Wessel, D. Willkomm - Technische Universität Berlin , Germany)

[9] Simulation of Ad Hoc Routing Protocols using OMNeT++ - A Case Study for the DYMO Protocol (Christoph Sommer , Isabel Dietrich , Falko Dressle)

[10] An IEEE 802.11g simulation model with extended debug capabilities (Sorin Cocorada)

**Appendix A – input files format**

// input files

COMMUNICATION\_GRAPH\_FILENAME, string, CommGraph.txt

INTERFERENCE\_GRAPH\_FILENAME, string, InterfGraph.txt

COORDS\_FILENAME, string, Coords.txt

REQUESTS\_FILENAME, string, Requests.txt

N\_FREQS\_FOR\_SCHEDULE, int, 3

// output period and resolution

TIME\_PERIOD, int, 600

TIME\_RESOLUTION, int, 1

// number of clients

N\_CLIENTS, int, 100

// number of requests from each type

N\_NEAR\_REQUEST, int, 5

N\_FAR\_REQUEST, int, 7

// definition of when a request is considered far

FAR\_THRESHOLD, double, 0.3

// 3 km square - the program select points in the unit square,

// and then scales them up

SCALE\_IN\_METERS, int, 70000

// dynamic model (velocities are normalized to unit square)

// so 0.2 for a square of 30km means 6 km / h

MEAN\_VELOCITY, double, 0.2

STD\_VELOCITY, double, 0.1

MAX\_VELOCITY, double, 0.4

// communication parameters

// in GHZ

FREQUENCY, double, 2.4

TRANSMISSION\_POWER\_IN\_MILI\_WATTS, double, 0.1

NOISE\_IN\_DBM, double, -101.7

SNR\_TO\_PER\_FILE\_NAME, string, snr2per.txt

// In meters (different model before and after)

D\_BREAKPOINT, double, 20.

// STD in DB

SF\_BEFORE\_BREAKPOINT, double, 3.

SF\_AFTER\_BREAKPOINT, double, 6.

// Arbitrary - to get yes no for and edge instead of distribution

NUM\_STD\_FOR\_DECAY, double, 3.

// Sensitivity parameters - for communication and interference (in db)

MIN\_SNR\_FOR\_COMMUNICATION, double, 20.

MIN\_SNR\_FOR\_INTERFERENCE, double, 10.

// protocol parameters - data size are in kbyte, other lengths in bits, times are in micro-seconds, rates are in mbit/sec

HEADER\_TIME, double, 4.

PREAMBLE\_TIME, double, 16.

MAC\_SERVICE\_TRAILER\_TIME, double, 8

MAC\_HEADER\_LEN, double, 240.

MAC\_FCS\_LEN, double, 32.

MAC\_RTS\_LEN, double, 160.

MAC\_CTS\_LEN, double, 112.

MAC\_ACK\_LEN, double, 112.

DIFS\_TIME, double, 28.

SIFS\_TIME, double, 10.

CW\_TIME, double, 135.

// this means 2kbytes = 16 kbits of data before encoding

PAYLOAD, double, 2.

**Appendix B – additional parameters**

* Max transmission power [mW]
* Application – header length
* MAC - headerLength, slotDuration [sec], difs [sec] , maxTxAttempts, rtsCtsThreshold, neighborhoodCacheSize [sec] , neighborhoodCacheMaxAge [sec] , txPower [mW]
* Phy - thermalNoise [mW] , sensitivity [dBm] , timeRXToSleep [sec] , timeTXToRX [sec] ,timeTXToSleep [sec], timeSleepToRX [sec] , timeSleepToTX [sec]
* Decider – SINR threshold
* PathLossModel - Alpha , Beta