İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

ENHANCING PERFORMANCE OF MULTIPLE IEEE 802.11 NETWORK ENVIRONMENT BY EMPLOYING COGNITIVE DYNAMIC CHANNEL ASSIGNMENT

M.Sc. Thesis by
Gökhan YILDIRIM

Department : Computer Engineering

Programme : Computer Engineering

JANUARY 2010

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

ENHANCING PERFORMANCE OF MULTIPLE IEEE 802.11 NETWORK ENVIRONMENT BY EMPLOYING COGNITIVE DYNAMIC CHANNEL ASSIGNMENT

M.Sc. Thesis by
Gökhan YILDIRIM
(504061514)

Date of submission : **25 December 2009**
Date of defence examination: **21 January 2010**

Supervisor (Chairman) : **Prof. Dr. Sema OKTUĞ (ITU)**
Members of the Examining Committee : **Assist. Prof. Dr. Feza BUZLUCA (ITU)**
Assoc. Prof. Dr. Fatih ALAGÖZ (BU)

JANUARY 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÇOKLU IEEE 802.11 AĞ ORTAMLARINDA BİLİŞSEL DİNAMİK KANAL ATAMASIYLA BAŞARIM ARTTIRIMI

YÜKSEK LİSANS TEZİ

Gökhan YILDIRIM
(504061514)

Tezin Enstitüye Verildiği Tarih : 25 Aralık 2009
Tezin Savunulduğu Tarih : 21 Ocak 2010

Tez Danışmanı : Prof. Dr. Sema OKTUĞ (İTÜ)
Diğer Jüri Üyeleri : Y. Doç. Dr. Feza BUZLUCA (İTÜ)
Doç. Dr. Fatih ALAGÖZ (BÜ)

OCAK 2010

FOREWORD

First, I would like to express my deep appreciation and thanks to my supervisor Prof. Dr. Sema OKTUĞ for her contribution and support. I also wish to thank Berk CANBERK for his guidance in cognitive radio networks and his help in implementing the test environment.

Second, I would like to thank to my friends Serda KASACI, Zeynep YURDAKUL and Tevfik YAPICIOĞLU for always sharing their knowledge and their friendship with me.

Third, I would like to thank to Fatma YILDIRIM for her support in creating the test environment.

Finally, I would like to thank to my family for their infinite support in my life and their infinite trust to my studies.

December 2009

Gökhan Yıldırım

Computer Engineer

TABLE OF CONTENTS

ABBREVIATIONS

AP	: Access Point
COF	: Channel Overlap Factor
CR	: Cognitive Radio
DSA	: Dynamic Spectrum Access
ETSI	: European Telecommunications Standard Institute
GHz	: Gigahertz
IEEE	: Institute of Electrical and Electronics Engineers
ISM	: Industrial, Scientific and Medical
Mbps	: Megabits per second
MHz	: Megahertz
NIC	: Network Interface Card
RF	: Radio Frequency
Wi-Fi	: Wireless Fidelity
WLAN	: Wireless Local Area Network

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF SYMBOLS

A	: Channel activity
c	: Channel number
CCA	: Cumulative channel activity
CIW	: Channel interference weight
D	: Difference of packet inter-arrival times
J	: Jitter
n	: Network number
N	: Network activity
P	: Pure channel activity
R	: Packet received time
S	: Packet send time
TA	: Total activity in channel
TCIW	: Total channel interference weight
W	: Interference weight

ENHANCING PERFORMANCE OF MULTIPLE IEEE 802.11 NETWORK ENVIRONMENT BY EMPLOYING COGNITIVE DYNAMIC CHANNEL ASSIGNMENT

SUMMARY

Many wireless networks are controlled by governmental agencies through a fixed spectrum assignment policy today. By this policy the governmental agencies regulate the spectrum and assign to license holders or services. A large part of the assigned spectrum is being used in an irregular manner which results in low utilization of the spectrum bands. Cognitive radio is a technology which is proposed to solve this problem with dynamic spectrum access techniques. The aim of cognitive radio is to operate in the best available channel to achieve a better performance. Spectrum sensing and spectrum decision are the main challenging functions of cognitive radio (CR). One of the research fields related to CR solutions is IEEE 802.11 based wireless networks. Infrastructured IEEE 802.11 based wireless networks require the use of a wireless access point (AP) that works on a certain channel to communicate with the nodes in the network.

In this thesis, we focused on selecting the best transmission channel for multiple AP environment by using the cognitive radio approach. The best transmission channel has the least activity on it. It is also the channel that is least affected by the interference. It is shown that interference is an important challenge for the neighboring wireless networks. A contribution to the channel selection system is made by selecting a channel not only to improve the performance of single AP (network), but also the performance of the neighboring networks forming the system. The channel utilization characteristic of the environment, the traffic load of the networks, and the interference are considered when deciding on the channel assignments. By using the test bed environment, the improvement of the system performance is shown. The test results also show that the proposed channel assignment scheme accomplishes performance improvement in throughput while balancing it among the networks. It also decreases jitter and packet loss.

ÇOKLU IEEE 802.11 AĞ ORTAMLARINDA BİLİŞSEL DİNAMİK KANAL ATAMASIYLA BAŞARIM ARTTIRIMI

ÖZET

Günümüzde pek çok kablosuz ağ, resmi birimler tarafından yapılan düzenlemeler ile kontrol edilen lisans sahiplerine ve servislere sabit spektrum dağıtım sözleşmeleri dikkate alınarak kullanılmaktadır. Bu dağıtılan spektrumun büyük bir kısmı düzensiz bir şekilde kullanıldığından spektrum bantları verimsiz kullanılmaktadır. Bilişsel radyolar dinamik spektrum erişim teknikleri ile bu sorunları çözmek için önerilmiş bir teknolojidir. Bilişsel radyonun amacı daha iyi performans elde etmek için mümkün olan en iyi kanalda hizmet vermeye çalışmaktır. Spektrum algılama ve spektrum karar verme bilişsel radyo ağlarında yerine getirilmesi gereken iki ana işlevdir. Bilişsel radyoların çalışma alanlarından biri de IEEE 802.11 tabanlı kablosuz ağlardır. Yapısal IEEE 802.11 tabanlı kablosuz ağlar, kendi ağlarındaki birimlerle haberleşmek için belirli bir kanalda çalışan erişim kontrol üniteleri kullanımını gerektirmektedir.

Bu tezde, tasarlanmış ve uygulanmış bilişsel erişim kontrol ünitesi için üzerinde en az trafik yükü olan ve kanallar arası parazitten en az etkilenen en iyi veri iletim kanalının seçilmesi üzerine odaklanılmıştır. Ayrıca kablosuz ağların ortak olarak kullandığı kablosuz ortamda kanallar arası parazitin önemli bir sorun olduğu gösterilmiştir. Bunun yanında kanal seçim sisteminde daha da iyileşme yapılarak, sadece bizim bilişsel erişim kontrol ünitemizin performansını arttırarak değil, komşu ağlarla birlikte tüm sistemin performansını arttıracak şekilde komşu ağlara da kanallar atayan bir sistem tasarlanmıştır. Kanal atamaları yapılırken ortamdaki kanalların kullanım oranı karakteristikleri, ağların trafik yükleri ve kanallar arasındaki parazit gerçeği göz önünde bulundurulmuştur. Gerçek test ortamında yapılan test uygulamalarıyla tüm sistemin performansının arttırıldığı gösterilmiştir. Test sonuçları ayrıca önerilen kanal atama uygulamasının, veri gonderim hızını arttırdığı gibi, bunu ağlar arasında dengelendirdiğini de göstermiştir. Bunların yanında paketlerin gecikme zamanlarının istatistiksel değişiminde azalma sağladığı gibi paket kaybında da azalma sağlamıştır.

1. INTRODUCTION

Most of the wireless networks use a fixed spectrum assignment policy that is regulation of the spectrum by governmental agencies and assignment to license holders or services today. A large part of the assigned spectrum is being used in an inefficient manner due to this spectrum management strategy [1].

This low utilization of the spectrum served well in the past. However, the requirement for additional bandwidth has increased dramatically by the fast growth of the widespread wireless services [2]. This resulted in bringing out a new communication form which is dynamic spectrum access.

Dynamic spectrum access (DSA) aims to implement cognitive radios that is defined as an intelligent communicaton technique that could reconfigure a user based on its needs and characteristics of the environment [3]. The aim of cognitive radio is to operate in the best available channel and to achieve high bandwidth by different kinds of wireless architectures and DSA techniques [1].

1.1 Purpose of the Thesis

Cognitive radios have been investigated for many wireless network communication services. One of the studying fields of cognitive radio solutions is IEEE 802.11 based wireless networks. Infrastructured IEEE 802.11 based networks require the use of a wireless access point that works on a certain channel to communicate with the nodes in the network. As IEEE 802.11 Wi-Fi networks are very popular in every aspects of our life, most IEEE 802.11 access points need to share the same communication medium that is one of the mostly used 11 channels in the 2.4 GHz band.

The purpose of this thesis is to select the best transmission channel for a specific cognitive access point. Another contribution of this thesis is in the channel selection system: Here, the proposed channel assignment will not only be applied to the specific network, but also to the neighboring networks to improve the whole performance of the system. It is also targeted to establish a test bed implementation to evaluate the performance of the proposed system.

1.2 Background

In conventional spectrum management strategy, most parts of the assigned spectrum are being used in an irregular manner. Figure 1.1 illustrates the inefficient utilization of the spectrum band.

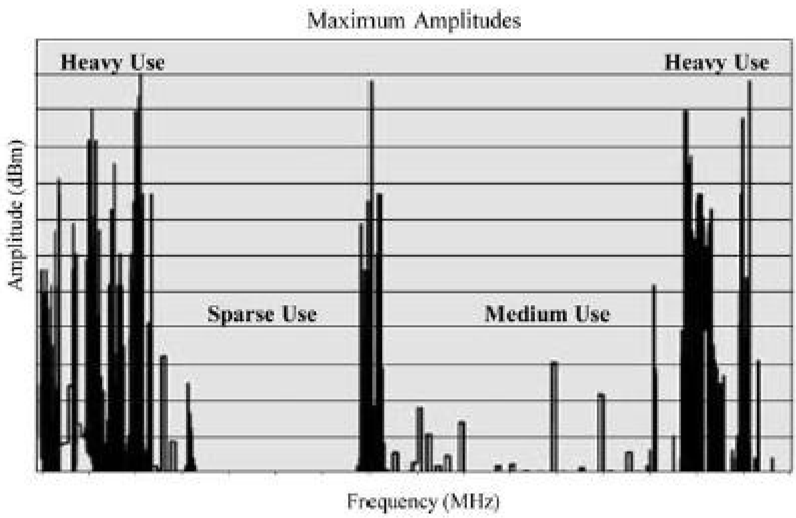

Figure 1.1 : Spectrum utilization [1]

Cognitive radio approach classifies the users into two different groups. The license holders of a certain spectrum band are considered as *primary users*. The access of the primary users to the spectrum band should not be affected by the unlicensed users. The second group of the users who are unlicensed are called *secondary users*. They could use the licensed spectrum bands in the absence of the primary users [1].

DSA communication paradigm intends to use the spectrum opportunistically. CR users are not allowed to interfere with the primary users that have settled licenses to operate on the related spectrum band. In [4], a two-phase channel and power allocation scheme, which is targeting to improve system throughput, defined as the number of simultaneous users, is introduced to avoid excessive interference to primary users by keeping the interference to primary users below a certain threshold.

Manoj et al. proposed a cognitive networking architecture that could use temporal patterns of the higher layer traffic information to achieve a better network throughput. They described a channel switching mechanism for the cognitive access points [5].

However, switching to the best available channel by sensing the environment could improve the performance of the related AP, on the other hand it could degrade the performance of the neighbouring networks. This work deals with designing a channel assignment scheme by increasing the performance of the whole system in the considered network area.

1.3 Structure of the Thesis

The remainder of this thesis is organized as follows: Chapter 2 gives basic information about IEEE 802.11 network standards and network design, and cognitive networking in IEEE 802.11 networks. Chapter 3 describes the proposed system model. Chapter 4 presents the observations and measurements made before testing the proposed system. Chapter 5 describes the test environment and traffic tools and gives the results obtained through the tests. Chapter 6 concludes the thesis by giving directions.

2. COGNITIVE NETWORKING in IEEE 802.11

2.1 IEEE 802.11 Network Standards

The IEEE 802.11b/g network standards use the unlicensed 2.4 GHz industrial, scientific and medical (ISM) band. The 2.4 GHz band is divided into 11 channels for the Federal Communications Comission or North American domain and 13 channels for the European or ETSI domain. These channels have a center frequency seperation of only 5 MHz and an overall channel bandwidth or frequency occupation of 22 MHz. Figure 2.1 shows the use of 2.4 GHz ISM band by IEEE 802.11 channels [6].

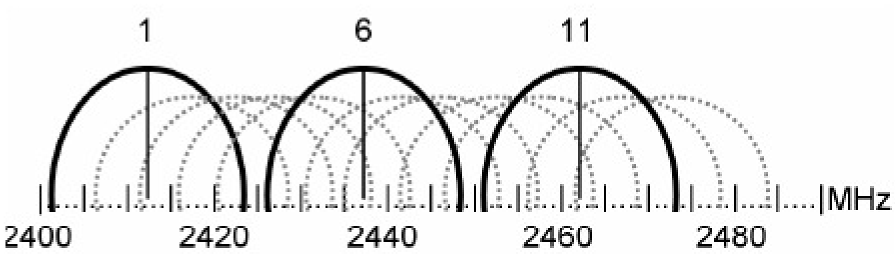

Figure 2.1 : 2.4 GHz ISM band [6]

2.2 Network Design in IEEE 802.11

An IEEE 802.11 wireless network must be designed regarding a number of interesting problems such as radio coverage of all the target space and having the required capacity to carry the expected load [7]. Besides that there are many interfering sources while having a limited number of non-overlapping frequency channels [6]. This introduces frequency assignment by avoiding interference.

2.2.1 Interference

The problem of having a negative effect on the network performance by the undesired effect of the interference becomes challenging as the number of neighbouring nodes increase [6].

Interference is determined by the level of RF energy that crosses between the channels in IEEE 802.11b/g. The energy extends beyond the edges of channel boundaries as radios do not have a precise edge to their channels. However, the level of the energy drops as we get farther from the channel center. Figure 2.2 shows the channelization scheme and how the energy of each channel spreads [8].

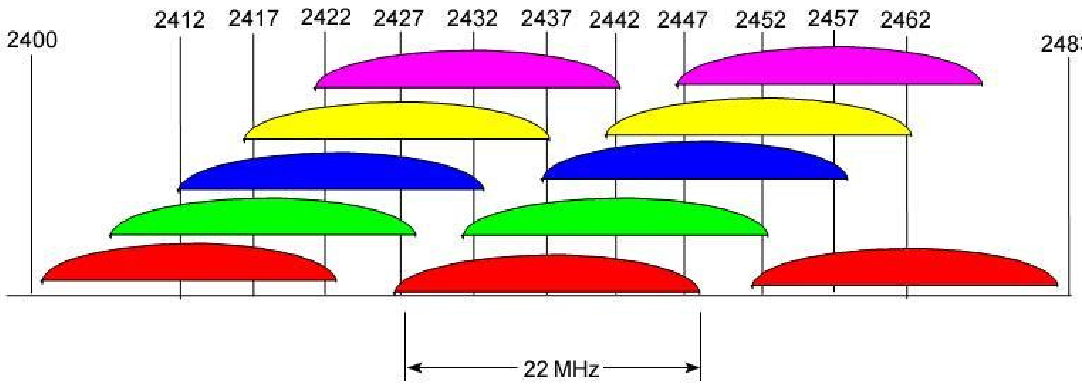

Figure 2.2 : Channels and energy spreads [8]

2.2.2 Channel selection

Most IEEE 802.11 wireless access points select as an opearting channel one of the three of the eleven channels that have minimum spectral overlap. These channels are 1,6 and 11 which are called orthogonal channels. However, when the traffic load of the nodes are very high, only three channels may not be enough to guarentee harmless coexistence of different networks with each other [6]. It is indeed only the adjacent three channels that are suffering the most from the interference [5]. This made people come up with the idea of using all available channels to improve performance. It is also studied emprically that a seperation of four channels could be used without degrading the performance [6][9].

In [5], a cumulative channel activity for the IEEE 802.11 channels is defined and the activity specific to each channel is summed with the activity of the neighboring channels that are within the channel overlap factor (COF). COF is taken as three for IEEE 802.11b networks.

2.3 Spectrum Sensing

Spectrum sensing is one of the main functions of cognitive radios with spectrum analysis and decision. Its purpose is to detect the unused spectrum so that secondary user share the spectrum without harmful interference on the other users. It is given in Figure 2.3 by the steps of cognitive cycle with spectrum analysis, that is estimating characteristics of the spectrum holes, and spectrum decision, that is choosing the appropriate spectrum with regard to the user requirements [1].

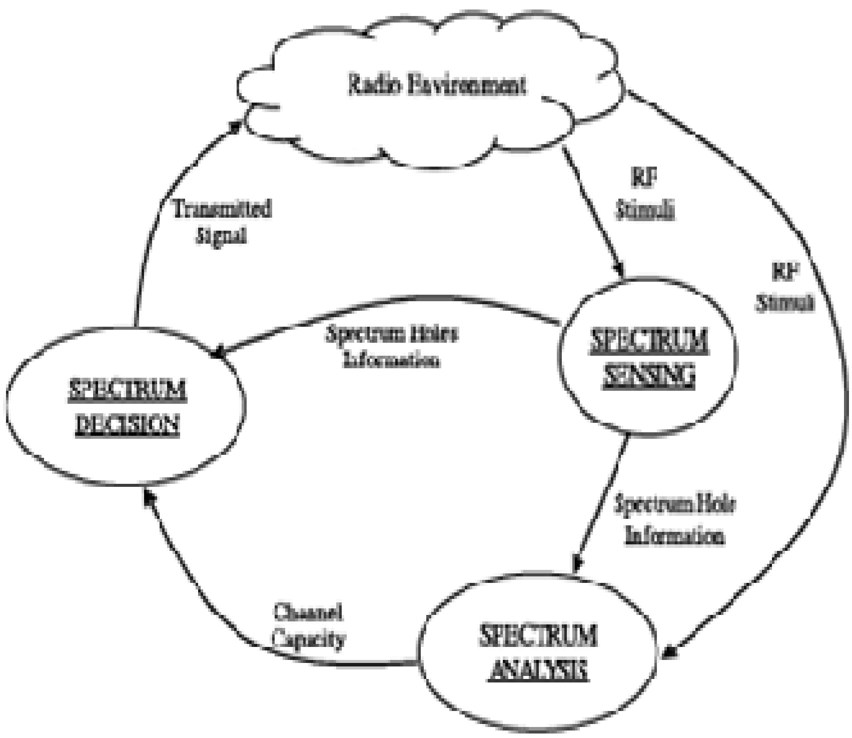

Figure 2.3 : Steps of cognitive cycle [1]

Although spectrum sensing in physical layer or physical layer cognition is very important, it is also complicated, expensive and provides only limited information about higher layer traffic information. It is argued in [5] that it is enough to have a cognitive networking information to achieve a better system performance even in the absence of physical layer cognition, with only higher layer traffic information.

To produce spectrum sensing information in IEEE 802.11 networks to obtain higher layer traffic information, the 11 channels in IEEE 802.11b/g could be monitored. This monitoring could be done through traffic sniffing on the wireless network.

In order to monitor channels, the network interface should be in a special mode which is called monitoring mode. It is one of the modes for the wireless network interfaces. In this mode the network interface stops transmitting data and only sniffs the currently configured channel. The wireless network interfaces could also be configured in managed, ad-hoc or master modes. Managed mode is commonly used by most stations to connect to a local AP to gain connectivity to the wireless network. Ad-hoc mode is used for short-term connectivity between stations when an AP is not available. Finally, in master mode, the network interface operates as an AP.

It is necessary to monitor activities in all 11 channels for IEEE 802.11 b/g spectrum. However, a monitoring network interface could monitor only the currently configured channel at a time. A sampling approach is used in [5] to monitor each particular channel for a sampling period of time in a cyclic manner with channel hopping, since it would be a very resource-consuming activity to monitor all channels at the same time.

2.4 Design of a Cognitive Access Point

It is important to select a good channel in IEEE 802.11 spectrum for WLAN APs. Conventional APs do not have a mechanism to dynamically select the best available channel that has the least activity and least interference effect on it [5]. Most conventioanl APs come with a default preset channel by the manufacturers apart with providing an interface to enable users to change the operating channel manually. This preset channel is mostly one of the orthogonal channels that are 1,6 or 11. However, users generally do not modify the preset channels and this results in some channels being used heavily while some others are mostly unused.

Some manufacturers like Cisco and D-Link provided a mechanism for initial channel selection such that the AP selects the least used channel when it is powered up, but again although this provides a better performance, it is not able to cope with the changes in the environment.

CogNet AP is proposed in [5]. It provides a periodic update of the employed channel based on the traffic load obtained during the channel observations. In this thesis, we also tried to design and implement such a cognitive AP. We tried to move the work further to obtain a solution for all adjacent networks by selecting a channel not only for the considered AP, but also for the other networks.

As shown in Figure 2.4, our designed cognitive AP has two network interfaces:

1. AP service interface
2. Monitoring interface

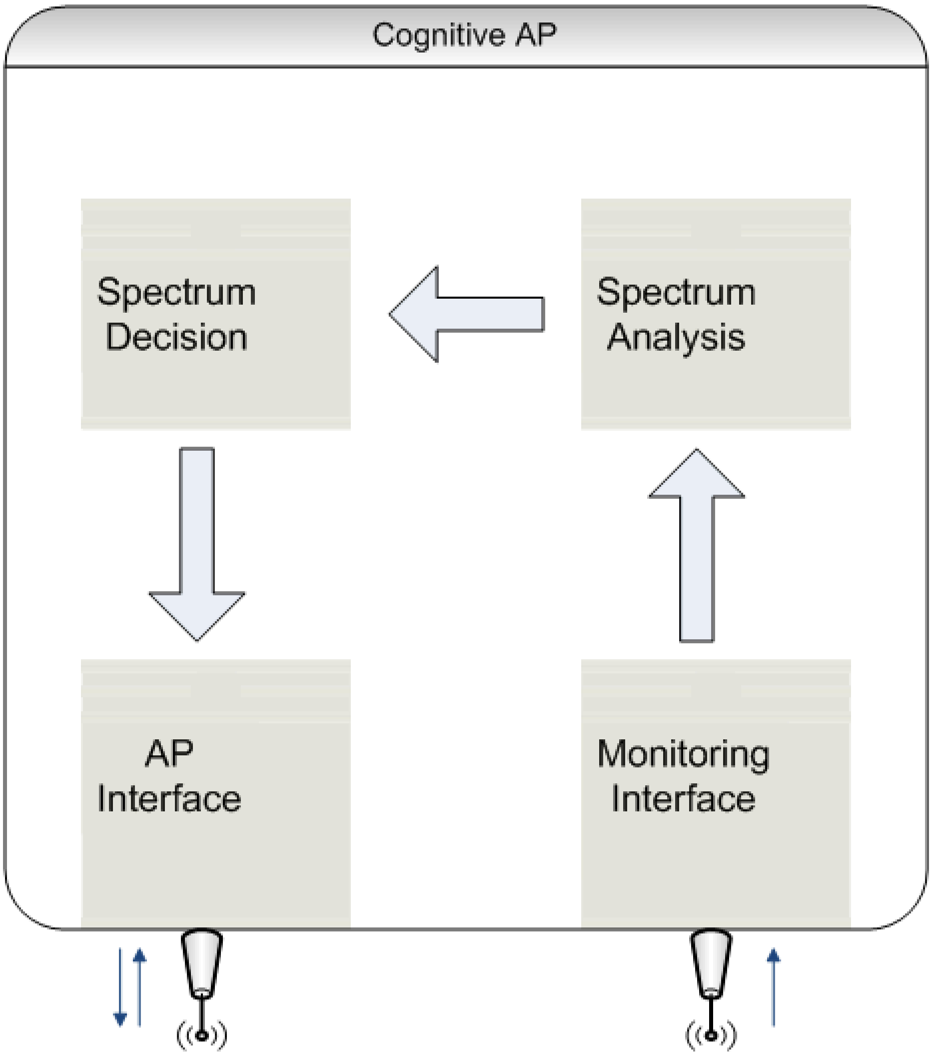

Figure 2.4 : The structure of the cognitive AP

The AP service interface works in the master mode. It provides the classical services to the users of the AP. On the other hand, the monitoring service interface works in the monitor mode. It monitors the traffic activity in all channels. The AP service interface selects the operating channel regarding the information provided by the cognitive analysis and decision modules.

The bahaviour of the designed cognitive AP is given in the following flow chart Figure 2.5:

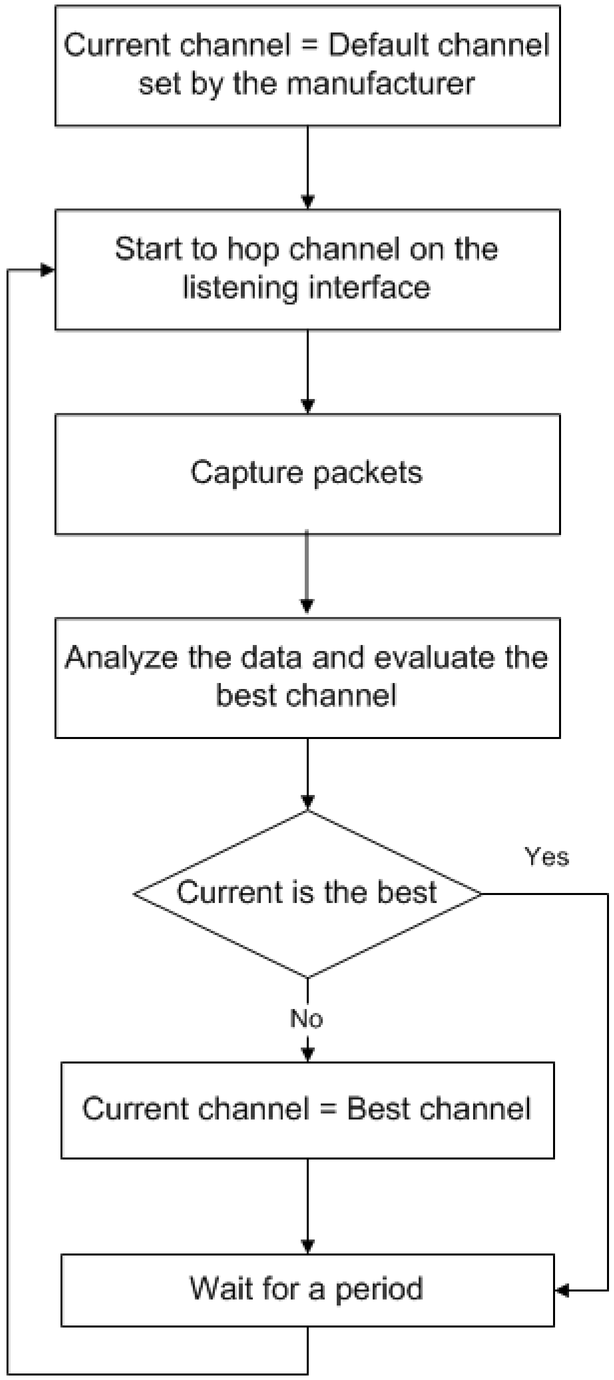

Figure 2.5 : Flow chart of the cognitive AP behaviour

The initial channel of the AP is the default channel set by the manufacturer. Before starting to capture packets, the monitoring interface starts to hop on channels and monitors each channel for a sampling period. The activity of each channel is measured by the average throughput of the captured packets in each channel. The channel that has the least activity is selected as the best channel. However, the AP interface is stopped and re-started prior to selecting the channel that has the least activity. As it takes 5-10 seconds for the AP to continue its operation after the channel selection, we could say that this is the drawback of the proposed system. However, this is a laboratory test bed environment, by using more complicated hardware which we aim to do in the future, this drawback could be removed. In this work, we try to keep the waiting period relatively long to avoid performance degradation. This designed cognitive AP is implemented and tested using the test bed at the Computer Networks Research Laboratory. The details of the setting on the implementation will be given in Section 4.

2.5 Channel Selection Regarding the Interference

Interference has always been a major problem in wireless communication techniques. Although some studies have been made in modeling the interference for cognitive radio networks [10] and techniques have been investigated to minimize the interference [11], it is a fact that it will always exist for IEEE 802.11 wireless networks and APs need to work on channels that are least affected by the interference to have a good performance in WLANs.

The activities of adjacent channels are taken into account using the following formula in [5]:

$$CCA_k = \sum_{i=1}^{N_{ch}} W_{ik} \times A_i \quad \textbf{(2.1)}$$

where CCA_k is the cumulative channel activity for channel k, N_{ch} is the number of channels. W_{ik} is the weight factor of the interference coming from neighboring channels and is defined with the following formula:

$$W_{ik} = \begin{cases} 1 & \text{if} \quad \text{k - COF} \leq \text{i} \leq \text{k} + \text{COF} \\ 0 & \text{otherwise} \end{cases} \quad \textbf{(2.2)}$$

where COF is equal to three for IEEE 802.11b/g networks.

A_i is the activity in channel i and is evaluated by the average throughput of the channel.

Then the channel having the minimum cumulative channel activity is selected as the operating channel.

3. PROPOSED SYSTEM

3.1 Spectrum Usage of the Cognitive AP

Maximizing WLAN performance is a difficult task since it needs some sort of coordination and frequent monitoring, because the wireless medium has a dynamic nature and is shared with many devices like APs in the same WLAN, APs in the other WLANs and other devices that may not be APs at all [12]. The sharing of the wireless medium is much more considerable for IEEE 802.11b/g networks, since an AP that is working on a certain channel affects not only the devices operating on the same channel, but also the devices operating in the adjacent channels within channel overlap factor. As a result of this, when a cognitive AP is aiming to work on the best available channel to improve its performance, it could degrade the other APs' performance. Thus, we tried to develop a cognitive and fair channel assignment scheme that is aiming to improve the performances of the neighboring networks along with our performance. The designed cognitive AP finds out the unused channels and assigns these channels to the neighboring networks so that the total system performance is improved.

3.2 Channel Assignment Scheme Regarding Total System

We have the same problems described in Section 2 to produce a cognitive channel assignment scheme for all networks. We again need to consider the channel overlapping degree and the traffic load. In addition to this, we need to consider the weight of the traffic load and interference to the channel and to its adjacent channels we bring, while we assign a channel to a network.

Since we again have to cope with the channel utilization and interference problems, we built our model on Manoj [5]'s studies that is described in the Section 2. However, in this model more than a precise cumulative channel activity we considered the weight that we could bring in each of the all channel selection possibilities, if we had selected that channel for an AP to operate on.

Before making the decision of selecting channels for each APs, the activities in all channels are monitored and the following two important information that will help us to make a cognitive decision is produced:

- Network activities: The activities of each considered network.
- Pure channel activities: The activities in each channel excluding the considered network activities.

The channel interference weight that we could bring if we select channel *k* for one of the considered APs is given with the following equation:

$$CIW_k = \sum_{i=1}^{N_{Ch}} W_{ik} \times TA_i \qquad \textbf{(3.1)}$$

where W_{ik} is again the weight to bring the interference effect within the COF that is again taken as three.

$$W_{ik} = \begin{cases} 1 & \text{if} \quad \text{k - COF} \leq \text{i} \leq \text{k + COF} \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{(3.2)}$$

This time a total activity TA_i is considered to evaluate each weight which is sum of the pure channel activity on channel *i* and the sum of network activities that are considered to be on channel *i* for that case.

$$\begin{aligned} &TA_i = P_i + N_i \\ &P_i \rightarrow \text{pure channel activity on channel } i \\ &N_i \rightarrow \text{sum of network activities working on channel } i \end{aligned} \qquad \textbf{(3.3)}$$

The channel interference weights are evaluated for all possibilities for selecting *n* channels for *n* networks out of *c* channels. The total number of possibilities are given as the permutation of *n* channels out of *c* channels.

$$\text{Number of channel assignment possibilities} \rightarrow \mathrm{P}(c, n) \qquad \textbf{(3.4)}$$

In our test bed environment, we studied the case of selecting channels for 4 networks from the available 11 channels of IEEE 802.11b/g. Thus, it is,

$$\mathrm{P}(11,4) = 14641$$

in case for our test bed environment.

The channel interference weights for the channels that the networks are assumed to be working are summed to gain a total channel interference weight for each case.

$$TCIW = \sum_{i=1}^{N_{ch}} \sum_{j=1}^{N_{ch}} \sum_{k=1}^{N_{ch}} \sum_{l=1}^{N_{ch}} CIW_i + CIW_j + CIW_k + CIW_l \qquad \textbf{(3.5)}$$

The case that has the minimum of all evaluated total channel interference weights out of all cases is selected.

$$OperatingChannels = \arg\min_{i,j,k,l} TCIW_{i,j,k,l} \qquad \textbf{(3.6)}$$

The channels *i, j, k* and *l* for the case that have the minimum total channel interference weight, could be assigned to the considered networks to improve total performance. The networks will have good performance in these assigned channels. In addition, the interference effect among the networks will be minimized.

Figure 3.1 depicts the steps of the proposed cognitive channel assignment model, that improves total system performance, in a flow chart diagram.

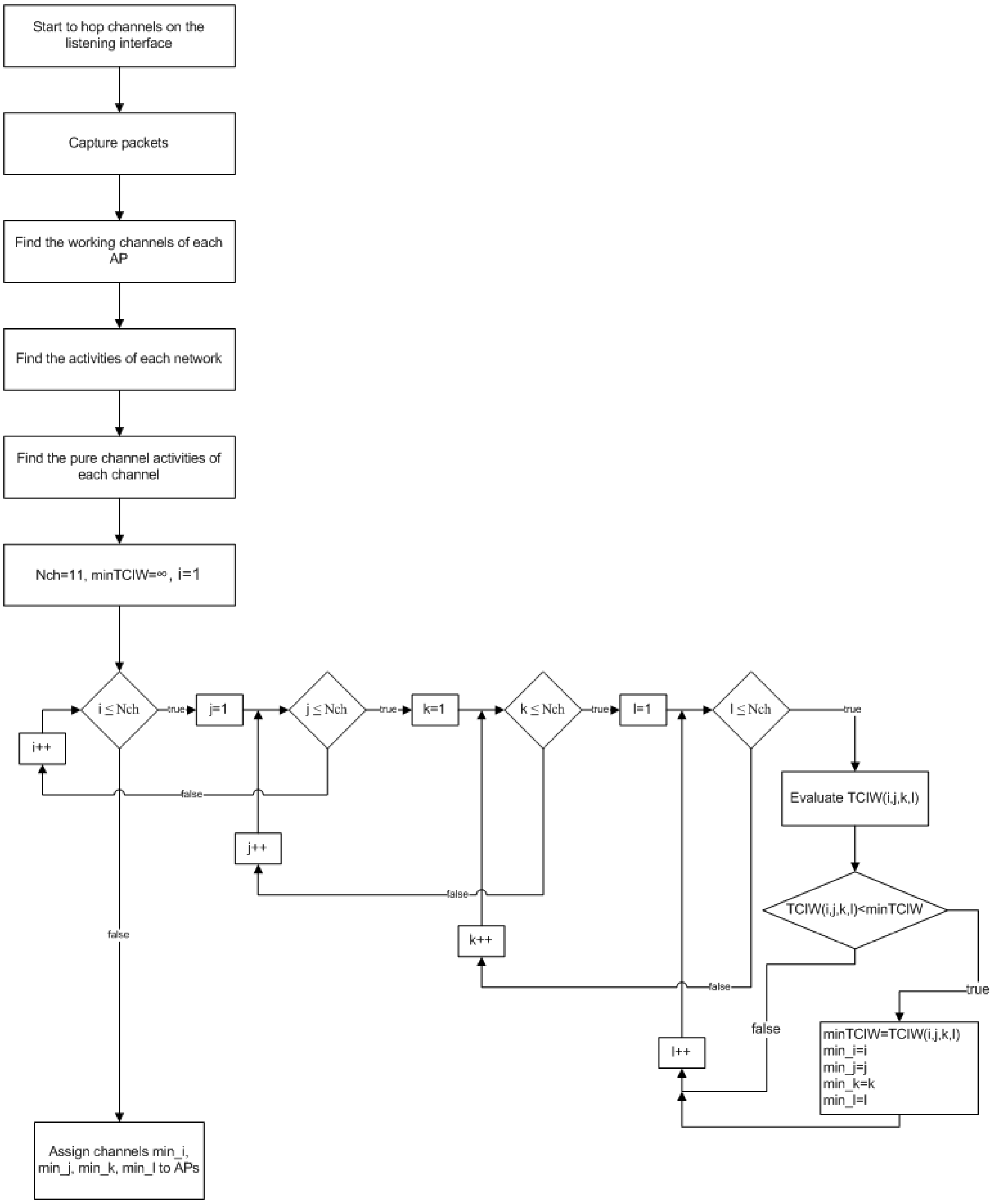

Figure 3.1 : Flow chart of the proposed cognitive AP

4. PRE-TEST OBSERVATIONS and IMPLEMENTATIONS

This thesis is mostly based on the observations and tests made on the real environment. Thus, the testing part will be documented in two different sections. The cognitive AP implementation details and the observations made with this AP in test environments are given in this section. The test bed environment details and the performance measurements of the proposed system will be given in the next section.

4.1 Implementation of Cognitive AP

Most IEEE 802.11 APs operate on a pre-configured channel and provide an interface to the user to configure the operating channel manually. To implement a cognitive AP, the channel selection was needed to be made automatically with regard to the cognitive decision mechanism. Our designed cognitive modules were needed to run on the AP itself, either in the driver level or in the application level.

Thus, we investigated the following both products in the market: Either an AP that we could run our own implemented modules on or a wireless network interface card, that we could configure as an AP to operate in the master mode. It is found out that *MadWiFi,* [13] which is one of the most advanced wireless network interface card drivers for Linux operating system, works only with wireless cards that have Atheros chipset on them. Therefore, we used a D-Link DWA-643 Xtreme N Notebook ExpressCard, which has a chipset Atheros AR5418, for our cognitive AP implementation. The card is given in the following Figure 4.1 [17].

Figure 4.1 : D-Link DWA-643 Xtreme N Notebook ExpressCard [17]

The card is installed on a laptop with Intel Core 2 Duo T7250 CPU operating at 2 GHz and with 2 GB RAM. The machine also had another on-board wireless NIC, Intel Wireless WiFi Link IWL4965AGN. This way, we had 2 NICs on our implemented cognitive AP.

- D-Link DWA-643: Acts as an AP and provides network access to wireless clients.
- Intel IWL4965AGN: Configured in monitored mode to capture all packets on the air, acting as a wireless traffic sniffer.

The implemented cognitive AP runs on a Linux Ubuntu operating system.

Before starting to capture the packets of the wireless networks, the monitoring interface must be set to the monitor mode in which the wireless card stops transmitting data and sniffs the currently configured channel. However, the majority of the wireless cards are used in managed mode or ad-hoc mode. Wireless driver developers may not even include support for monitor mode access. In the case of Linux, many drivers support monitor mode and the others that do not support could be patched to support it. However, in the case of Windows, drivers are closed source, which prevents anyone except the driver developer from supplying the monitor mode functionality. Thus, it is a good choice to implement such an AP in a Linux environment, since it is easier to configure the monitoring part. The wireless network interface could be configured easily to work in the monitor mode with the following commands:

>*ifconfig wlan0 down*

>*iwconfig wlan0 mode monitor*

>*ifconfig wlan0 up*

The wireless NIC drivers, name the interface like "*wlan0*" for the above example. Before setting the interface in the monitor mode, the interface is stopped and then started again after setting its mode.

Wireshark [14] and TShark [15] tools, that have sophisticated wireless protocol analysis support, are used to monitor the traffic, capture the IEEE 802.11 packets on the air and analyze the captured data. Wireshark is the most famous network protocol analyzer in the world and is used in many industrial and educational institutions. It is very easy to sift through huge amounts of wireless traffic data with Wireshark's display filter and protocol decoders [16]. TShark is also another network protocol analyzer without the visualization support of Wireshark. It is also a good network traffic dumper and lets you get the necessary parameters easily for each packet from a huge data set.

There are some challenges for sniffing in a wireless environment. A wired network offers a single medium mechanism such as the wire for packet capture. However, wireless networks can operate on multiple wireless channels using different frequencies. It is needed to be identified the channel or the frequency that we are going to listen, to analyze the traffic for an AP. We need to consider that wireless cards can only operate on a single frequency at any given time.

The following command could be used to configure the operating channel of the wireless network interface:

>*iwconfig wlan0 channel 1*

where *wlan0* is again our monitoring network interface.

Since we would like to discover the unutilized channels among all available channels, we have to listen and get data from all available channels. Nevertheless, listening all channels at the same time would be a resource expensive solution, since it will need a separate wireless network interface card sniffing on each channel. Thus, we used a technique which is called channel hopping. With channel hopping, the wireless card is still operating on a single channel at any given time, but is rapidly switching between channels. Fortunately, Wireshark or TShark operates independently of the current channel selection. When we change to the desired channel while Wireshark is running, Wireshark will still continue to collect traffic on the changed channel.

The following shell script, similar to the script given in [16], is prepared and used for channel hopping:

```
#!/bin/bash
IFACE=wlan0
IEEE80211bg="1 2 3 4 5 6 7 8 9 10 11"

while true ; do
  for CHAN in $IEEE80211bg ; do
    echo "Switching to channel $CHAN"
    iwconfig $IFACE channel $CHAN
    sleep 5
  done
done
```

Figure 4.2 : Channel hopping shell script

4.2 Observations with the Implemented AP

The implemented cognitive AP is used to monitor different network environmets in different times to see the traffic characteristic difference between these environments. The observations are made in both a residential environment and an office environment. Channel sampling duration is selected as 5 seconds. The packets are captured for 20 minutes in each environment and then the activities of each channel are evaluated from these captured data. TShark’s capture-to-file functionality is used to store the captured data. The necessary parameters from each packet information, like frame length, radio frequency and time information, are taken out with the following command using TShark:

>tshark –r capture_file –e frame.len –e radiotap.channel.freq –e frame.time_relative –T fields > output

This command dumps the necessary information from the "capture_file" to the file named "output".

4.2.1 Observations in residential environment

Firstly, the observations are made in a residential environment in both a weekday and a weekend. The average values are taken out of a 20 minutes observation.

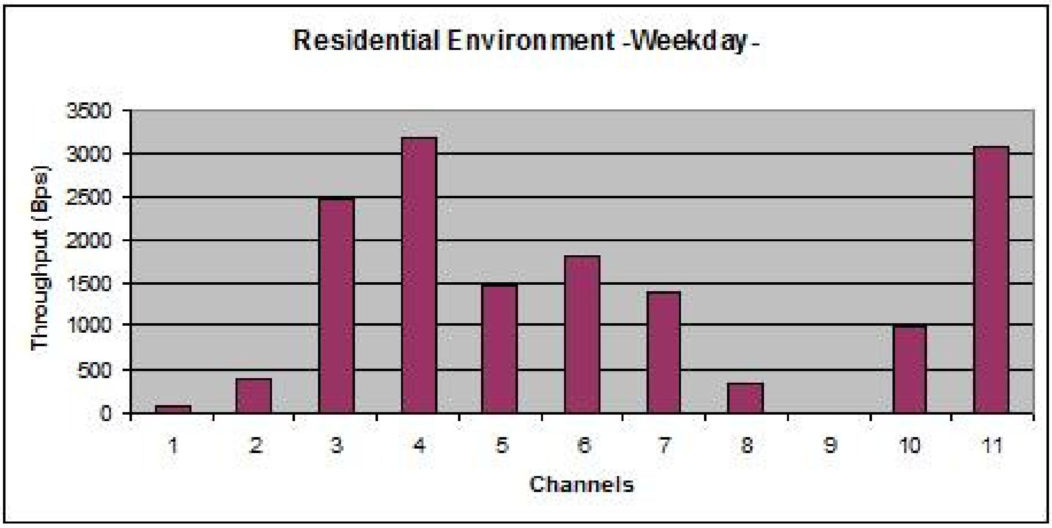

Figure 4.3 : Residential environment - weekday

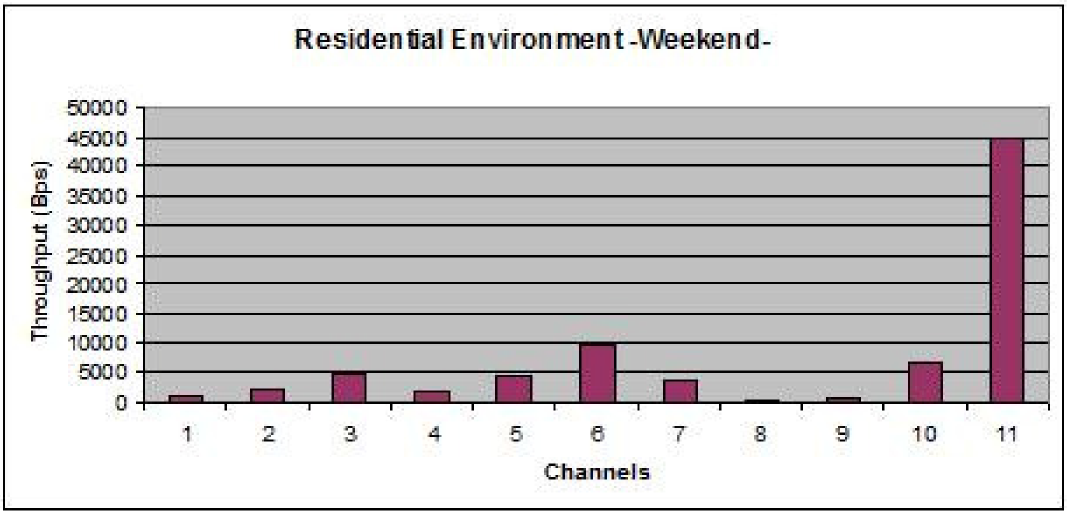

Figure 4.4 : Residential environment – weekend

It is observed that the traffic density at the weekend is much more than the traffic density at the weekday. There were not much data traffic at the weekday in the residential environment and most of the traffic was created by IEEE 802.11 management packets like beacon frames, probe request or probe response frames. However, some significant traffic could be observed on the weekend observations compared to the weekday traffic pattern and this traffic is mostly on channel 11, so we could come to a conclusion that most WLANs are operating at channel 11 in the observed residential environment.

The observations shown in Figure 4.3 and 4.4 were only results of a 20 minutes observations. It is also observed for 12 hours to see the daily change in the traffic behaviour.

It is observed that the activities are mostly on channels 1,6 and 11 in day time. The usage of every channel change in every hour. The activity in channel 11 is significantly higher than the other channels in the evening hours when most people are at home and using their wireless networks.

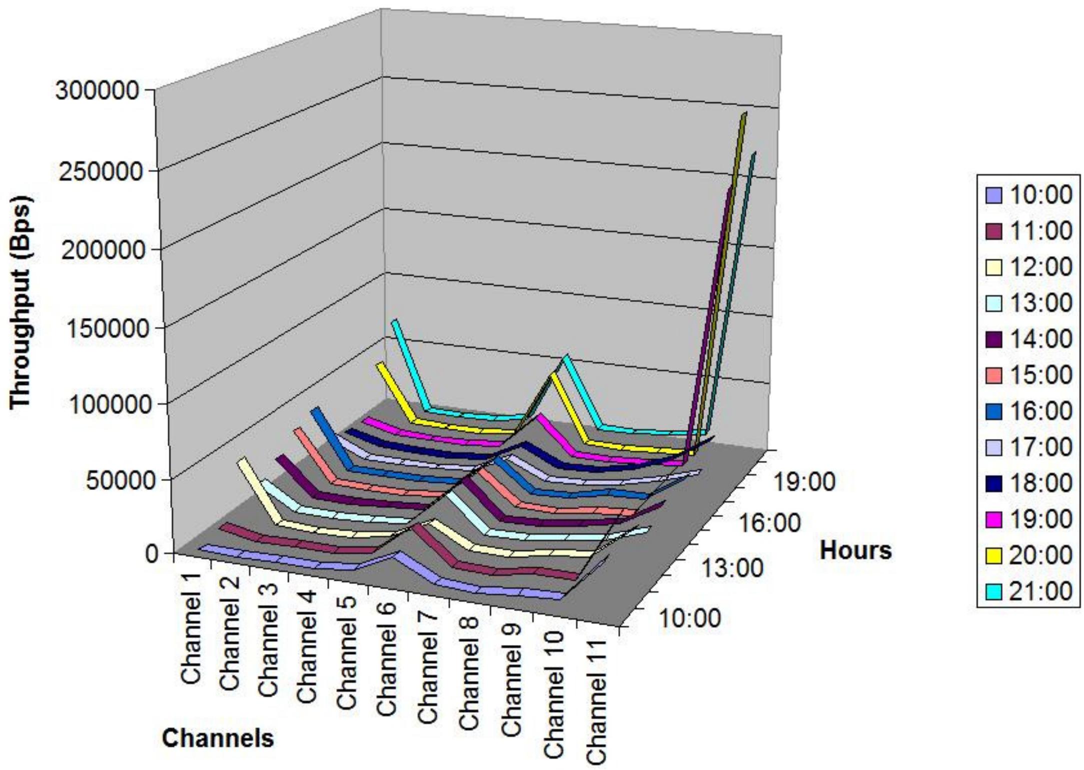

Figure 4.5 : Twelve hours observation -residential-

4.2.2 Observations in office environment

The same observations are repeated in an office environment to look for the characteristics of the channel usage in this environment.

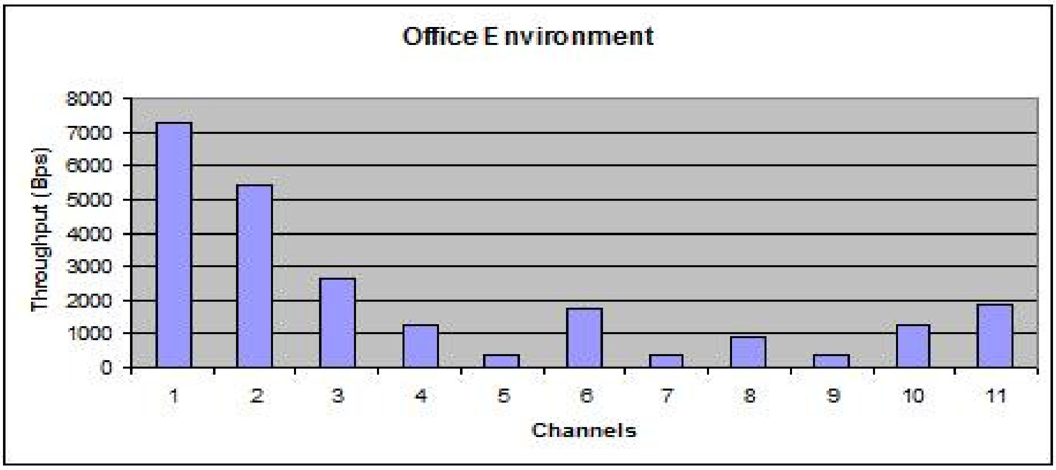

Figure 4.6 : Office environment

In the office environment, we observed a definite traffic pattern difference compared to the residential environment. It is observed that most WLAN APs are operating at channel 1 in the office environment.

The 12 hours observations of the office environment is given in Figure 4.7.

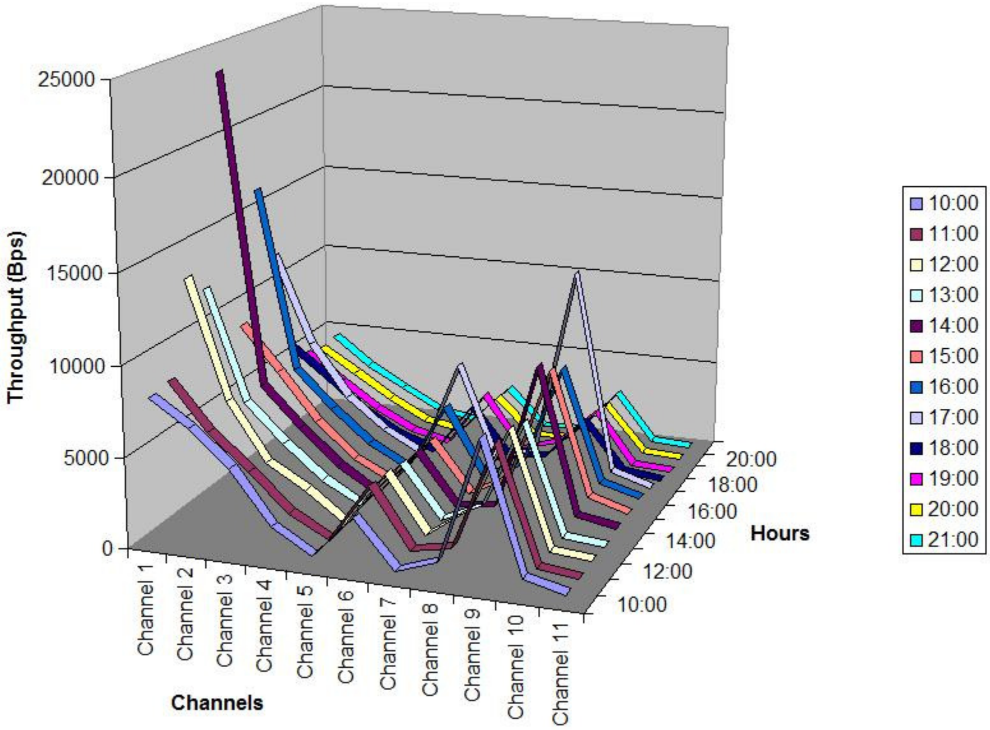

Figure 4.7 : Twelve hours observation -office-

It is observed that the use of the spectrum is much more sparse in the office environment compared to the residential environment. Although channel 1 is mostly used in most hours of the day, there are also significant activities in channels 2, 6 and 9 in some hours of the day.

4.2.3 Measurements after channel selection

As seen in the above measurements, traffic patterns could be different among different environments. This leads us to think about how the traffic load in each channel is affecting the performance on these channels. Thus, we made the same measurements in every channel from 1 to 11 to see the negative effect of the traffic load to the performance. We selected the working channel of the AP manually from the AP's user interface. Then we transferred 10 times an approximately 70 MB file by FTP and evaluated the average throughput in each test. The measurements are taken in both environments. The FTP throughput measurements in Figure 4.8 are made just after the channel activity measurements shown in Figure 4.4. Similarly, the FTP throughput measurements in Figure 4.9 are made just after the channels activity measurements shown in Figure 4.6.

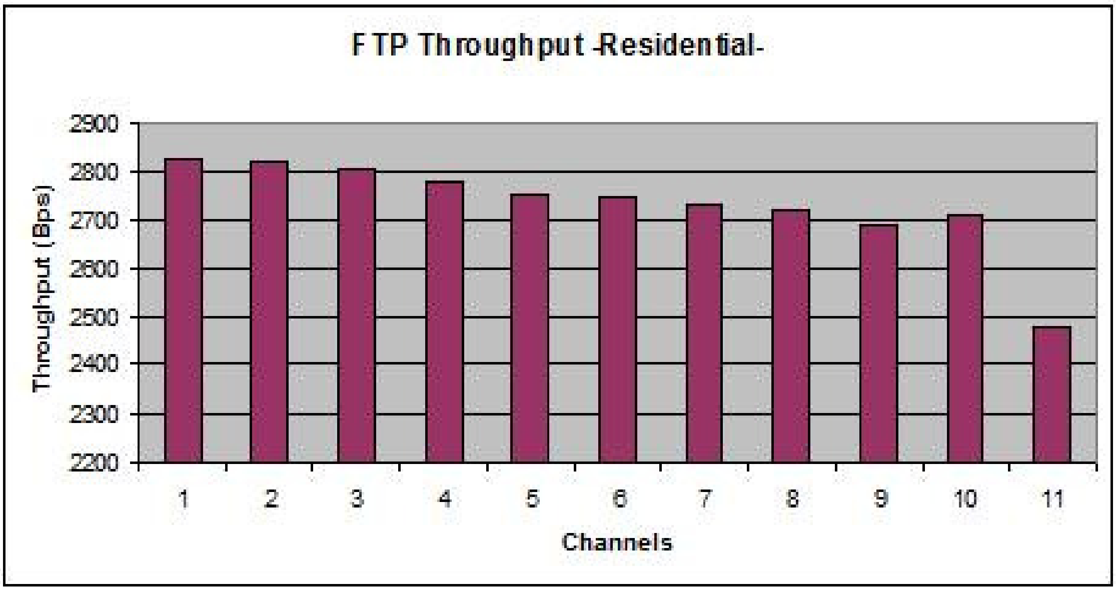

Figure 4.8 : FTP throughput -residential(weekend)-

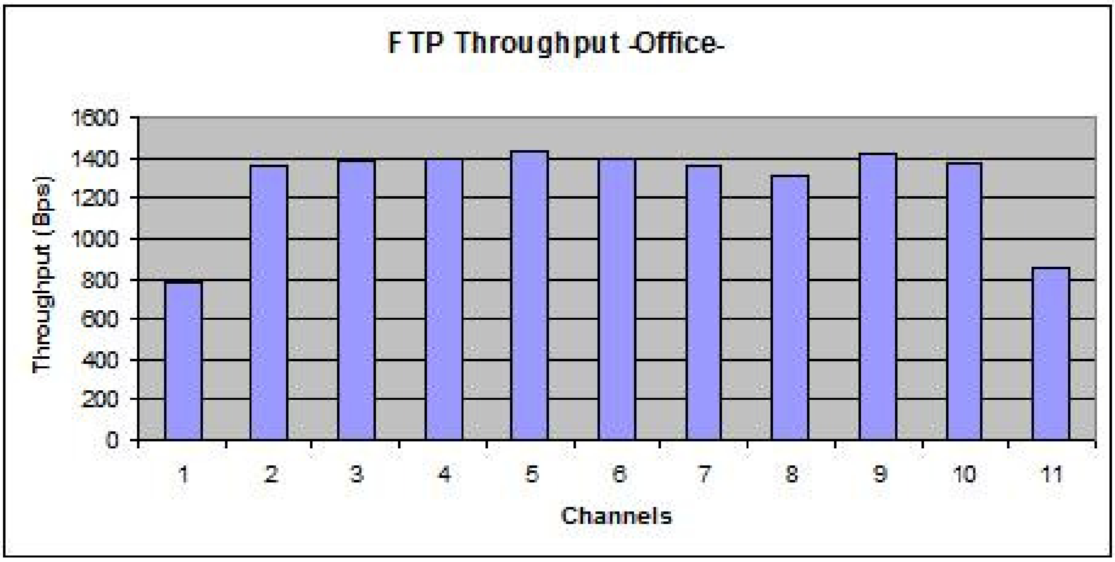

Figure 4.9 : FTP throughput -office-

In residential environment, the channel that has the best throughput for file transfer is channel 1 with 2829,72 KB/s and the worst channel (11) with 2475,70 KB/s. If we would have chosen the channel having best throughput instead of the worst, we would have had a performance improvement of about 14%.

In the office environment, the channel that has the best throughput for file transfer is channel 5 with 1428,05 KB/s and the worst channel (1) with 776,82 KB/s. If we would have chosen the best channel instead of the worst, we would have had a performance imrovement of about 84%. The reason beyond having hugely more improvement in the office environment compared to the residential environment is interference which we will be investigating in the following section. As it can be seen from Figure 4.7, the poor performance of the worst channel (1) is because of having a considerable traffic load in the neighbouring channels of channel 1.

4.3 Observations Related to Interference

As it is studied in section 2, interference affects the network performance negatively. In this section we tried to test this negative effect of the interference.

It is tried to test the effect of the interference coming from neighbouring channels within COF, which is 3 for IEEE 802.11 networks. Firstly, we settled 3 different networks having random traffic loads to operate in the following 3 different channels (1, 7, 11):

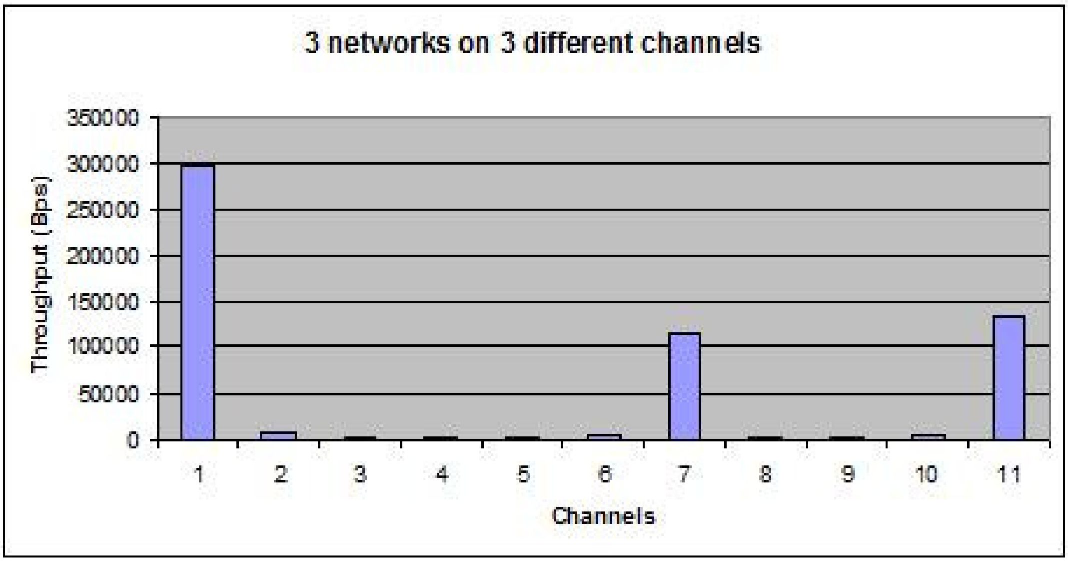

Figure 4.10 : Three networks on 3 different channels

Then, we measured the performance of an FTP transfer in every channel while the above traffics are active in the background. The following figure shows the time duration to complete an FTP transfer on each channel.

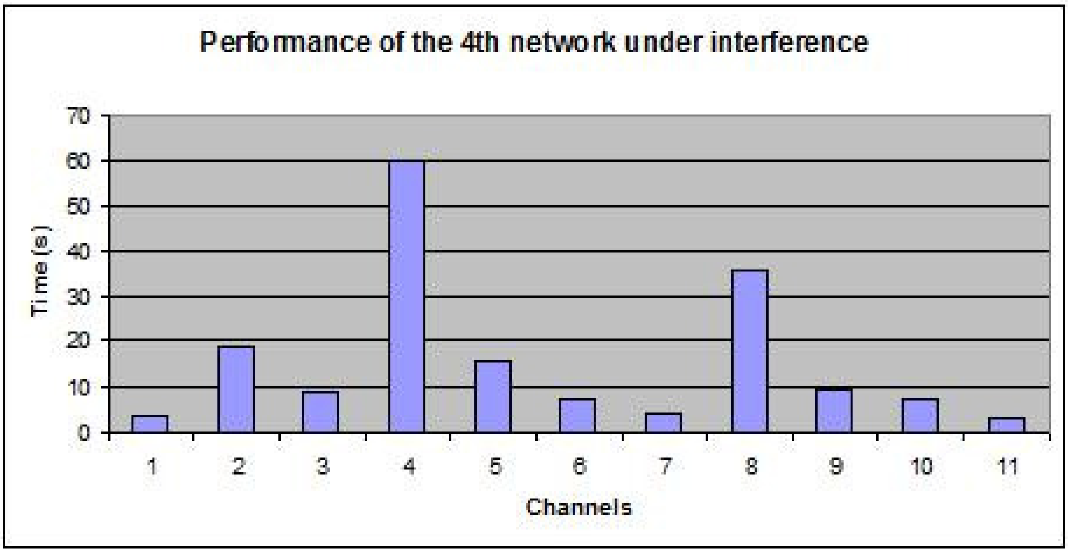

Figure 4.11 : The performance of the 4^{th} network under interference

It is shown in Figure 4.11 that the performance in channels 4 and 8 are very bad compared to the other channels. The reason behind this is that channels 4 and 8 are within the COF of 2 different background traffic creating networks. Channel 4 is affected by the background traffic on channels 1 and 7. Channel 8 is affected by the background traffic on channels 7 and 11.

We had made some more observations to clearly understand the effect of one network background traffic. We set the background traffic creating network to operate on channel 1 and runned a background TCP traffic on it.

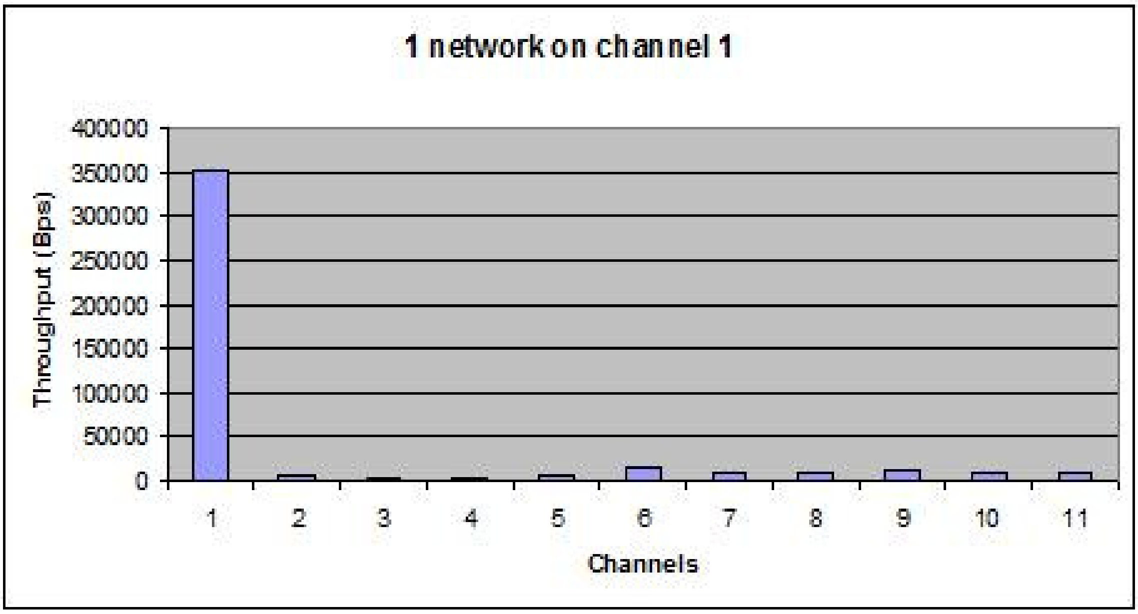

Figure 4.12 : One network on channel 1

Then, the throughput performance of FTP is measured on another network. The test is repeated on each channel respectively by configuring the second network to operate on each channel respectively.

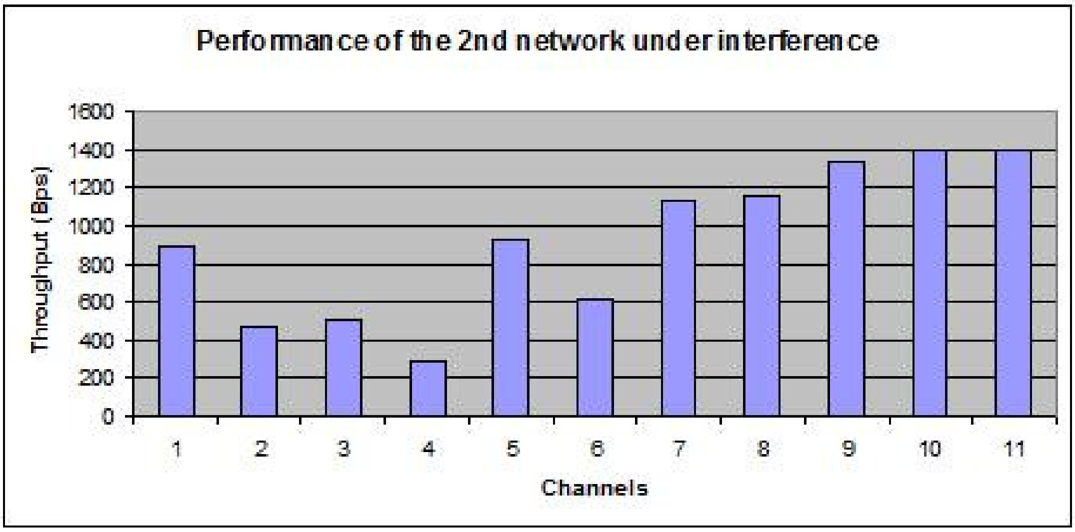

Figure 4.13 : The performance of the 2^{nd} network under interference

Figure 4.13 shows the throughput performance of FTP transfer made on each channel. It could be seen from the figure that the performance is worse than the other channels in the three adjacent channels of channel 1 where a background traffic is running. The poor performance in channel 6 is because of the characteristic of the test environment where channel 6 is normally more occupied than the other channels.

Then we repeated the same test by setting the background traffic on channel 11.

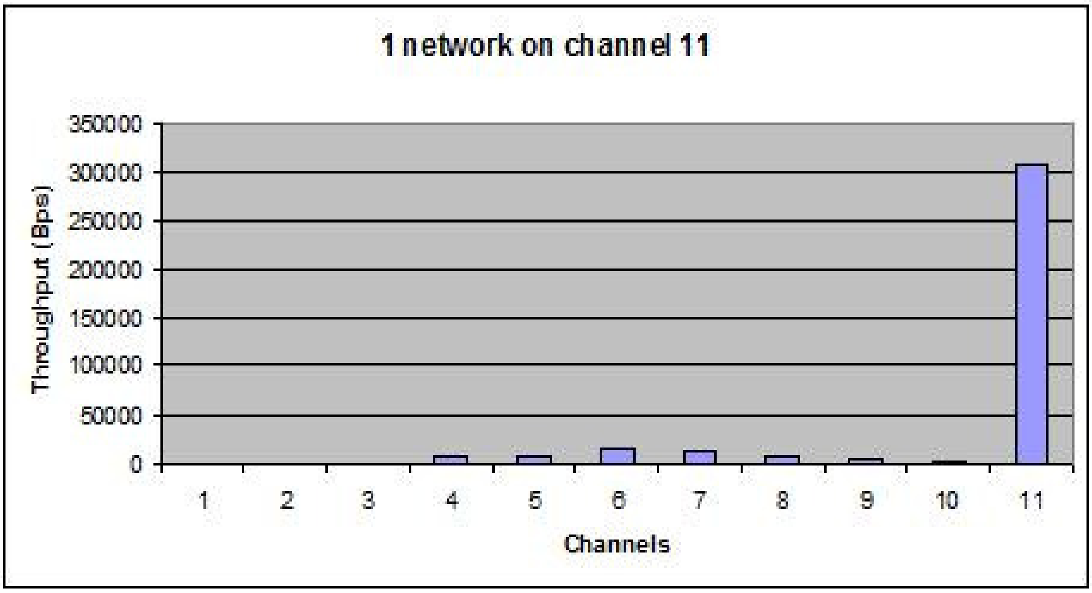

Figure 4.14 : One network on channel 11

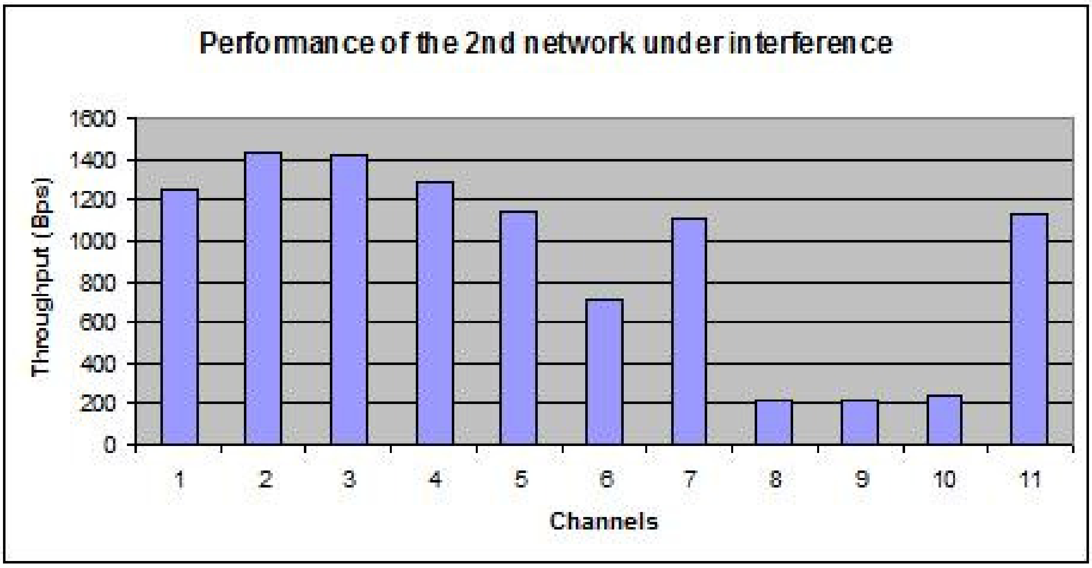

Figure 4.15 : The performance of the 2nd network under interference

The poor performance in the adjacent 3 channels could be seen clearly in Figure 4.15 again.

5. TEST BED USED and RESULTS OBSERVED

5.1 Test Environment

The proposed system described in Section 3 was implemented on a test bed deployed in Computer Networks Research Laboratory of Electrical and Electronics Faculty building at Istanbul Technical University, Istanbul. One base cognitive AP and a collection of other APs and client machines form the basis of our system. The base AP has 2 NICs one of which is D-Link DWA-643 card to provide the AP services. The other NIC on the base AP is an Intel IWLWIFI card to provide the monitoring services of the cognitive AP. It runs on Ubuntu Linux operating system. Atheros based D-Link DWA-643 card is used with the open source MadWiFi driver to have the capability to use the card as an AP in the master mode. MadWiFi is one of the most advanced WLAN drivers today with providing some tools to easily configure and get information about the network environment. The modules to implement the cognitive AP are developed with a combination of C modules and unix shell scripts. Other than the cognitive AP, 3 Linksys compact wireless-g wrt54gc APs are used to create different networks in the test environment. 8 client machines are used to connect to these 4 APs to have 2 nodes in each generated network. The client machines are a collection of PCs and laptops having wireless network adapters on them. The layout of the best bed environment is illustrated in the following figure:

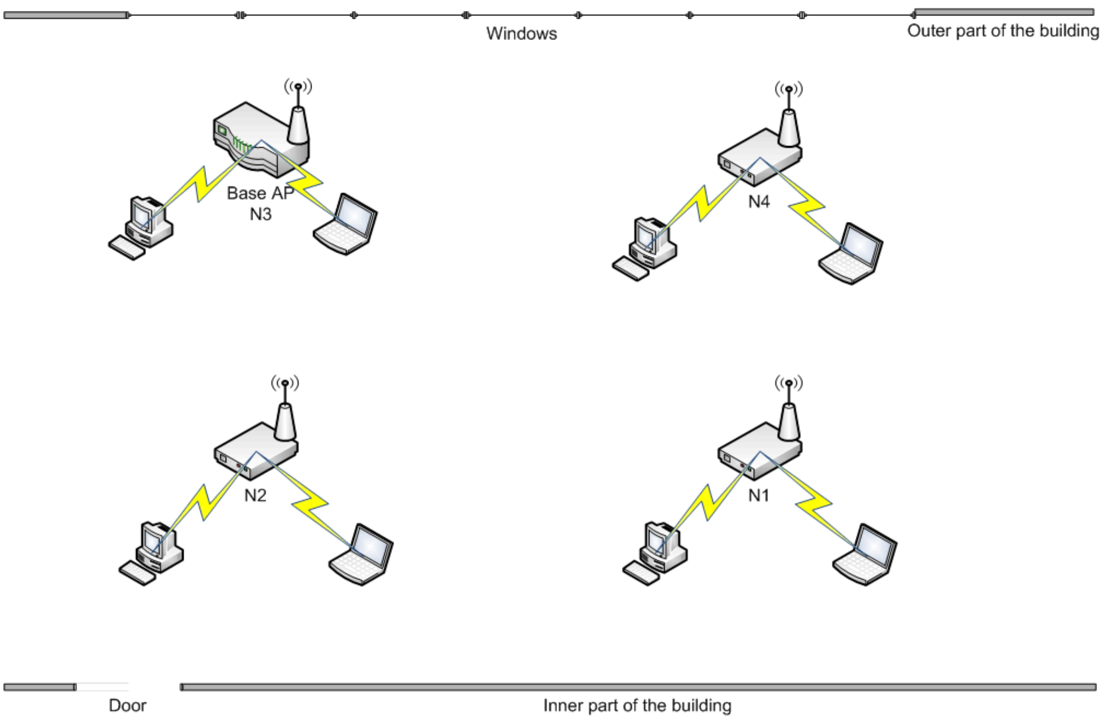

Figure 5.1 : Test bed environment

We observed the test environment for 10 hours to understand the general characteristic of channel utilization in this test environment.

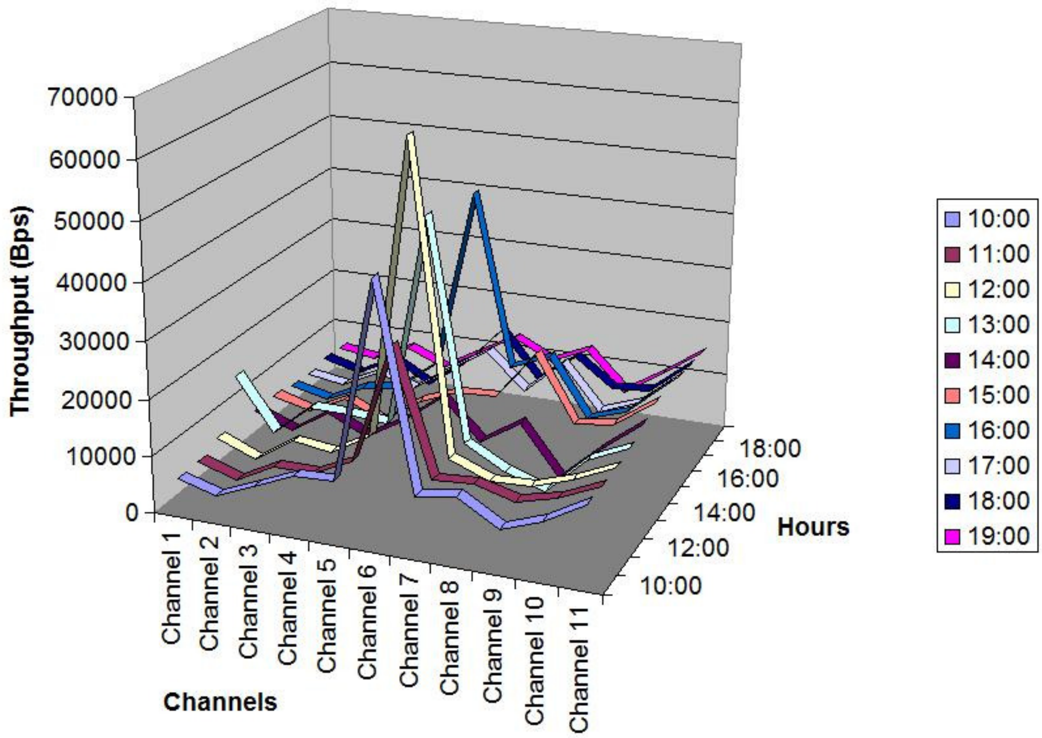

Figure 5.2 : Ten hours observation -ITU-

Figure 5.2 shows that channel 6 is more occupied compared to the other channels around our test environment which is in Computer Networks Research Laboratory in ITU.

5.2 Traffics Generated

After establishing the connections of the networks, it is required to have different types of traffic loads on these networks. Thus, we needed a useful traffic generator tool to generate different types of traffics and analyse the results on these traffics. We used NET IQ Corporation Chariot Console 5.0 tool as the traffic generator for the networks.

An endpoint software [18] is installed to each client that the traffic will run on in order to launch network traffics between the clients. Chariot Console application is installed to one of the computers in the networks to control and observe the performance of the tests. Chariot supports different kinds of traffic types with different network protocols and different built-in, but editable scripts to observe different parameters like throughput, response time, streaming parameters etc.

The following traffic scripts are used to create a background traffic in each network respectively:

Table 5.1: Network Traffic

Networks	Traffics
Network 1	HTTP Text traffic
Network 2	HTTP Gif traffic
Network 3	POP3 E-mail traffic
Network 4	Real Media Stream, Audio-Video traffic

5.2.1 HTTP text traffic

This script emulates the transfer of text files from an HTTP server. The number of transaction records is changed from its default value and made larger to have a long term running background HTTP traffic. The steps that each endpoint is performing a part of this script is given below:

```
HTTP Text Transfer

Endpoint 1                                      Endpoint 2
SLEEP
  time = initial_delay (0)
LOOP                                            LOOP
  count = number_of_timing_records (500)          count = number_of_timing_records (500)
  START_TIMER
  LOOP                                            LOOP
    count = transactions_per_record (10)            count = transactions_per_record (10)
    CONNECT_INITIATE                                CONNECT_ACCEPT
      port = source_port (AUTO)                       port = destination_port (AUTO)
    SEND                                            RECEIVE
      size = size_of_record_to_send (300)             size = size_of_record_to_send (300)
      buffer = size_of_record_to_send (300)           buffer = size_of_record_to_send (300)
      type = control_datatype (trans.cmp)
      rate = send_data_rate (UNLIMITED)
                                                    SLEEP
                                                      time = delay_before_responding (100)
    RECEIVE                                         SEND
      size = file_size (1000)                         size = file_size (1000)
      buffer = receive_buffer_size (DEFAULT)          buffer = send_buffer_size (DEFAULT)
                                                      type = send_datatype (news.cmp)
                                                      rate = send_data_rate (UNLIMITED)
    DISCONNECT                                      DISCONNECT
      type = close_type (Reset)                       type = close_type (Reset)
    INCREMENT_TRANSACTION
  END_LOOP                                        END_LOOP
  END_TIMER
  SLEEP
    time = transaction_delay (0)
END_LOOP                                        END_LOOP
```

Figure 5.3 : HTTP text transfer script

This script is runned through the endpoints in Network 1. It is supposed that Network 1 users are surfing through mostly the text based web pages. The traffic that is generated in the background of Network 1 is given in Figure 5.4.

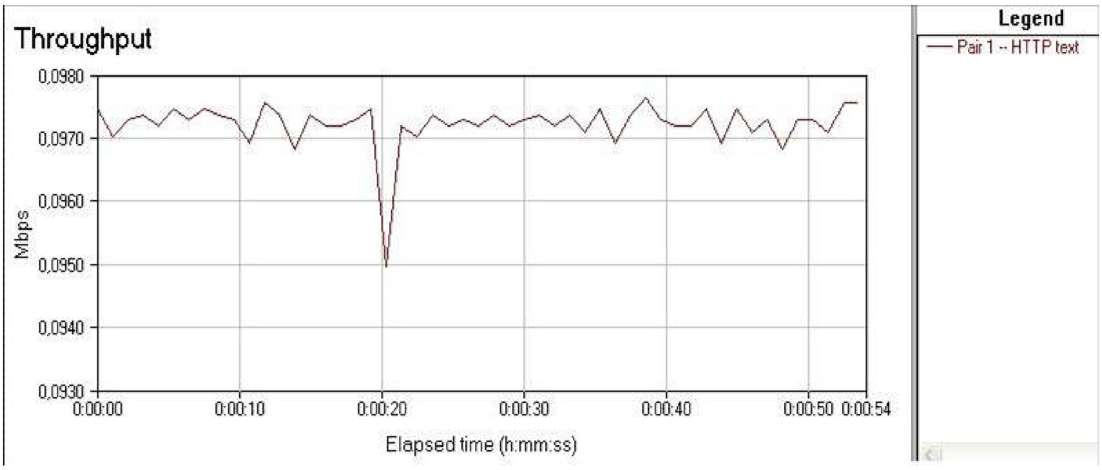

Figure 5.4 : HTTP text traffic

5.2.2 HTTP gif traffic

This script emulates the transfer of graphics from an HTTP server. The script is very similar to the HTTP text scipt. The only difference is in the file size of the transferred file. The number of transaction records is changed again. Also the script is configured to have some delay between transactions, since graphics transfer may not be made as frequent as text transfer. The steps that each endpoint is performing a part of this script is given below:

HTTP GIF Transfer

Endpoint 1	Endpoint 2
SLEEP	
time = initial_delay (0)	
LOOP	LOOP
count = number_of_timing_records (500)	count = number_of_timing_records (500)
START_TIMER	
LOOP	LOOP
count = transactions_per_record (10)	count = transactions_per_record (10)
CONNECT_INITIATE	CONNECT_ACCEPT
port = source_port (AUTO)	port = destination_port (AUTO)
SEND	RECEIVE
size = size_of_record_to_send (300)	size = size_of_record_to_send (300)
buffer = size_of_record_to_send (300)	buffer = size_of_record_to_send (300)
type = control_datatype (trans.cmp)	
rate = send_data_rate (UNLIMITED)	
	SLEEP
	time = delay_before_responding (100)
RECEIVE	SEND
size = file_size (10000)	size = file_size (10000)
buffer = receive_buffer_size (DEFAULT)	buffer = send_buffer_size (DEFAULT)
	type = send_datatype (lena.cmp)
	rate = send_data_rate (UNLIMITED)
DISCONNECT	DISCONNECT
type = close_type (Reset)	type = close_type (Reset)
INCREMENT_TRANSACTION	
END_LOOP	END_LOOP
END_TIMER	
SLEEP	
time = transaction_delay (200)	
END_LOOP	END_LOOP

Figure 5.5 : HTTP gif transfer script

This script is runned through the endpoints of Network 2. It is supposed that Network 2 users are surfing through mostly picture based web pages. The traffic that is generated in the background of Network 2 is given in Figure 5.6.

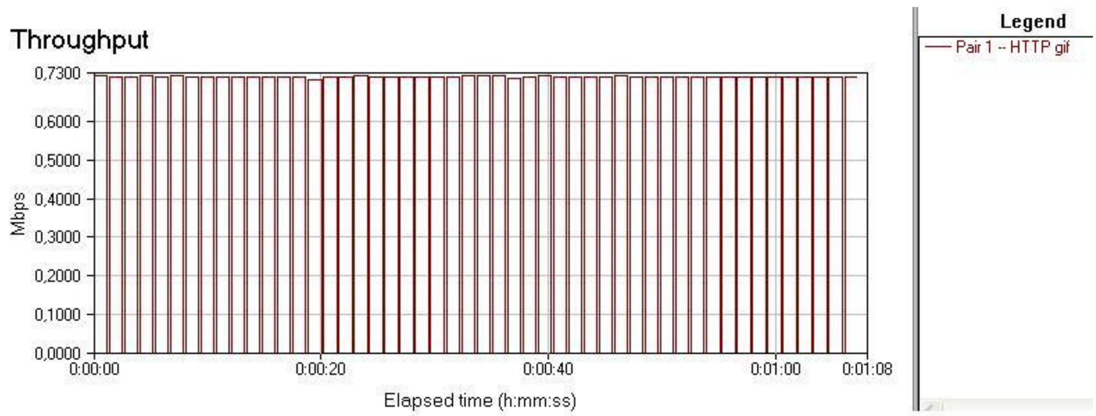

Figure 5.6 : HTTP gif traffic

As it could be seen from the above figure that the average throughput of this traffic is much more than the HTTP text traffic.

5.2.3 POP3 e-mail traffic

This script emulates typical receive e-mail transfers. The number of transaction records and the transaction delay parameters are changed from their default value for this script too. Since this script is a little bit longer than the other one. The details of the script will be given in Appendix A.1.

This script is runned through the endpoints of Network 3. It is supposed that Network 3 users are mostly reading their e-mails. The traffic that is generated in the background of Network 3 is given in Figure 5.7.

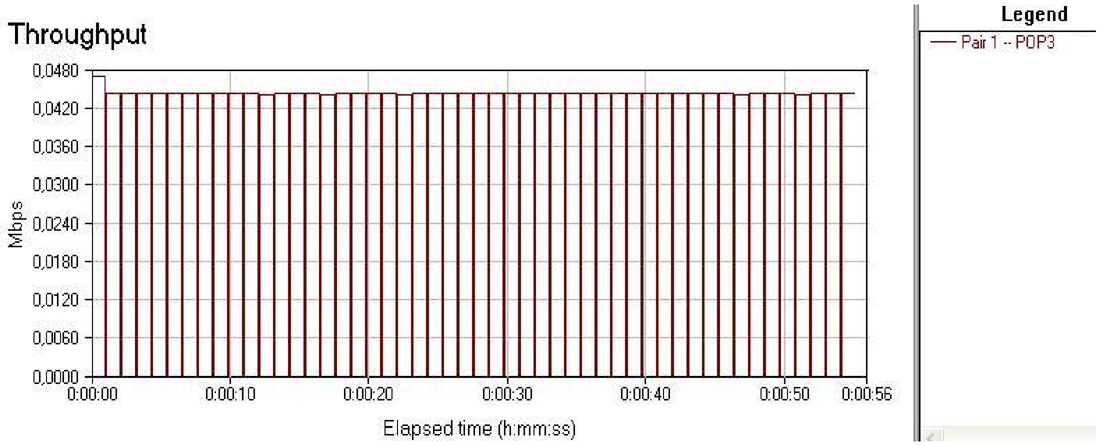

Figure 5.7 : POP3 e-mail traffic

The average throughput of this traffic is less than both the HTTP text and gif traffics.

5.2.4 Real time media traffic

This script emulates an encoded streaming traffic at rate 300 kbps. The details of the script is given in Figure 5.8.

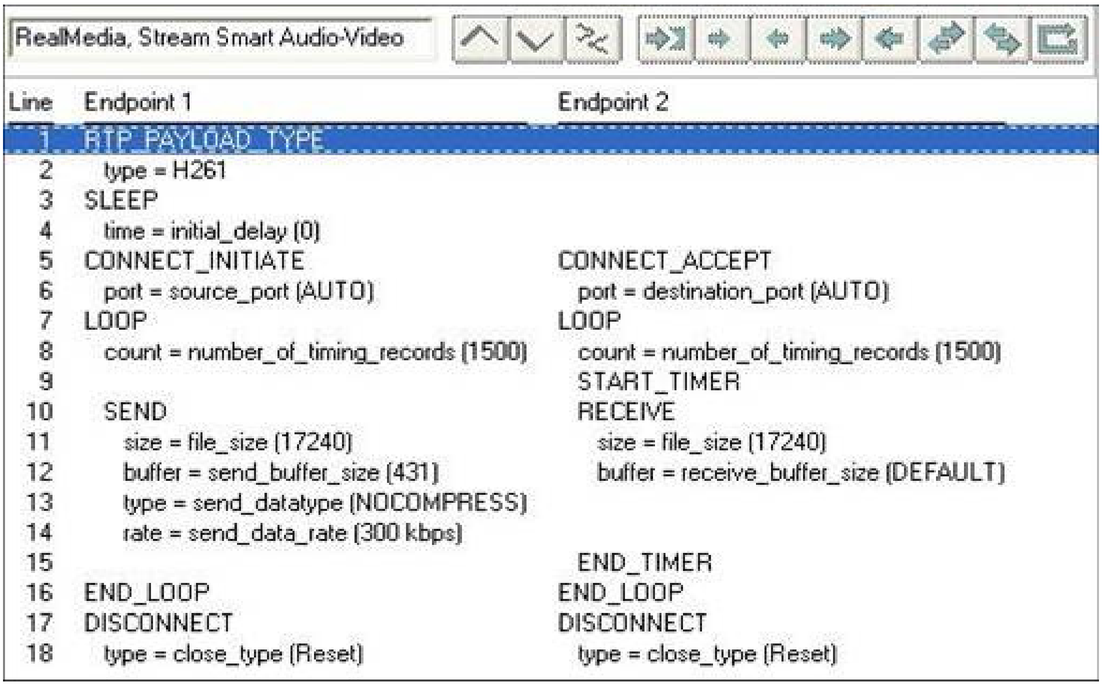

Figure 5.8 : Real time media traffic script

This script is runned through the endpoints of Network 4. It is supposed that Network 4 users are mostly listening or watching real time audio or video. The traffic that is generated in the background of Network 4 is given in Figure 5.9

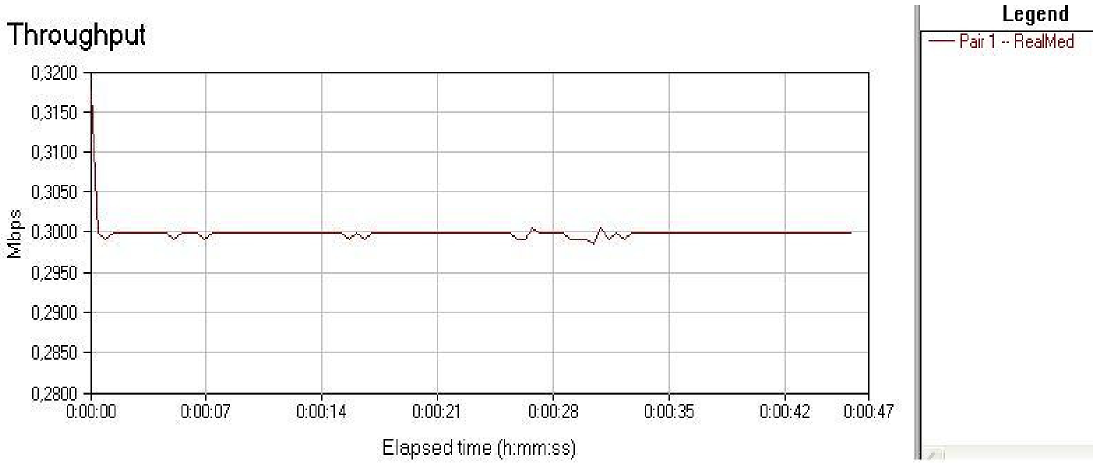

Figure 5.9 : Real time media traffic

5.3 Performance Metrics

While these 4 background traffics are running on each 4 networks, the performances of other TCP, UDP and RTP traffics are observed for different channel selections.

The real media transfer script, which is very similar to the background RTP traffic, is used for RTP traffic performance evaluations. A throughput script, details of which are given in Figure 5.10 is used for TCP and UDP traffic performance evaluations.

[Throughput]

Line	Endpoint 1	Endpoint 2
1	SLEEP	
2	time = initial_delay (90000)	
3	CONNECT_INITIATE	CONNECT_ACCEPT
4	port = source_port (AUTO)	port = destination_port (AUTO)
5	LOOP	LOOP
6	count = number_of_timing_records (100)	count = number_of_timing_records (100)
7	START_TIMER	
8	LOOP	LOOP
9	count = transactions_per_record (1)	count = transactions_per_record (1)
10	SEND	RECEIVE
11	size = file_size (100000)	size = file_size (100000)
12	buffer = send_buffer_size (DEFAULT)	buffer = receive_buffer_size (DEFAULT)
13	type = send_datatype (NOCOMPRESS)	
14	rate = send_data_rate (UNLIMITED)	
15	CONFIRM_REQUEST	CONFIRM_ACKNOWLEDGE
16	INCREMENT_TRANSACTION	
17	END_LOOP	END_LOOP
18	END_TIMER	
19	SLEEP	
20	time = transaction_delay (0)	
21	END_LOOP	END_LOOP
22	DISCONNECT	DISCONNECT
23	type = close_type (Reset)	type = close_type (Reset)

Figure 5.10 : Script used to evaluate throughput

Performance metrics used are throughput, lost data percent and jitter. The traffic generator tool Chariot uses the following equation to calculate the throughput:

$$(Bytes\_Sent + Bytes\_Received\_By\_Endpoint1) / (Throughput\_Units) / Measured\_Time \quad \textbf{(5.1)}$$

The throughput unit is selected as Mbps.

Chariot keeps track of the lost data, data that was discarded and not received by the receiving endpoint. From the header information in the RTP and UDP packets, receiving endpoint determines how much data was sent and calculates lost data by substracting the amount of data it actually received.

Jitter is the statistical variance of the datagram interarrival time expressed as a mean deviation for a single pair. It is calculated by using the following formulas defined in RFC 1889 [19].

$$D(i, j) = (Rj - Sj) - (Ri - Si) \quad \textbf{(5.2)}$$

Where R is the receive time and S is send time of the datagrams.

$$J = J + (|D(i-1,i)| - J)/16 \quad \textbf{(5.3)}$$

The tests are repeated in each of the following channel selections to compare the results with the channel selections proposed by our cognitive AP.

- **(2,6,7,11):** Supposed that each AP is working on a random selected channel.
- **(1,1,1,1):** Supposed that all APs are working on the same orthogonal channel.
- **(1,1,6,6):** Supposed that half of the APs are working on one orthogonal channel and half of the APs are working on another.
- **(1,1,11,11):** Supposed that half of the APs are working on one orthogonal channel and half of the APs are working on another.
- **(1,6,11,1):** Supposed that each AP is working on a random selected orthogonal channel.
- **(1,7,1,11):** Proposed channels by our AP in the lab environment.

5.4 Results Obtained

The performance parameters are measured in each of the channel selections described in the previous section. The graphs containing the test results for 4 networks in each channel selection are given in Appendix A.2. In this section, a summary of the test results and obtained graphs to compare the channel selections for all networks with the total values are given.

Before making its channel assignment decisions, our cognitive AP monitors the test environment and evaluates the network throughputs and the pure channel throughputs which are given in Figure 5.11 and Figure 5.12.

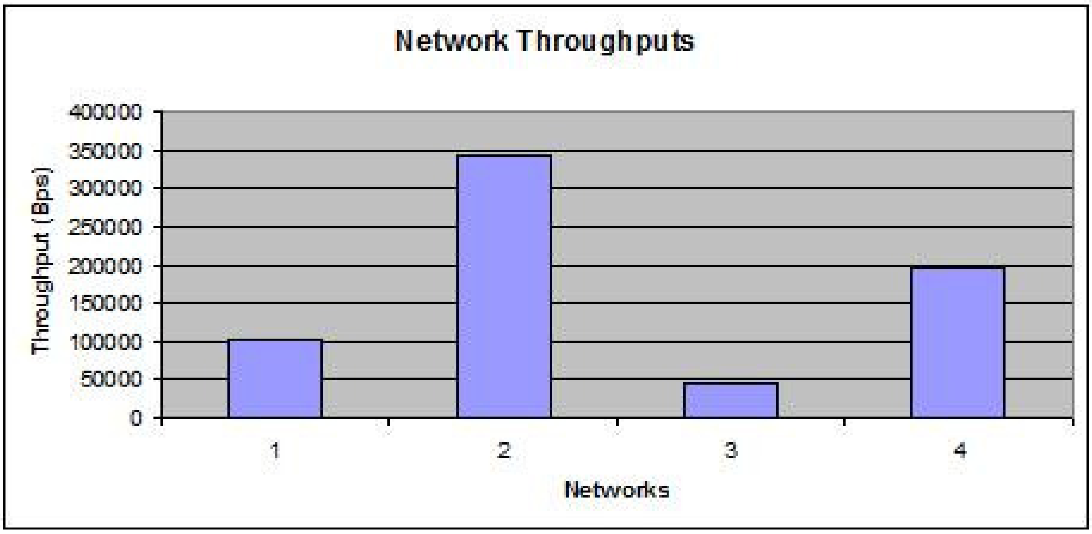

Figure 5.11 : Network throughputs

It could be seen from Figure 5.11 that the least traffic is running on Network 3 which is the network where POP3 e-mail traffic is running on. On the other hand, the most traffic is running on Network 2 where HTTP gif traffic is running on.

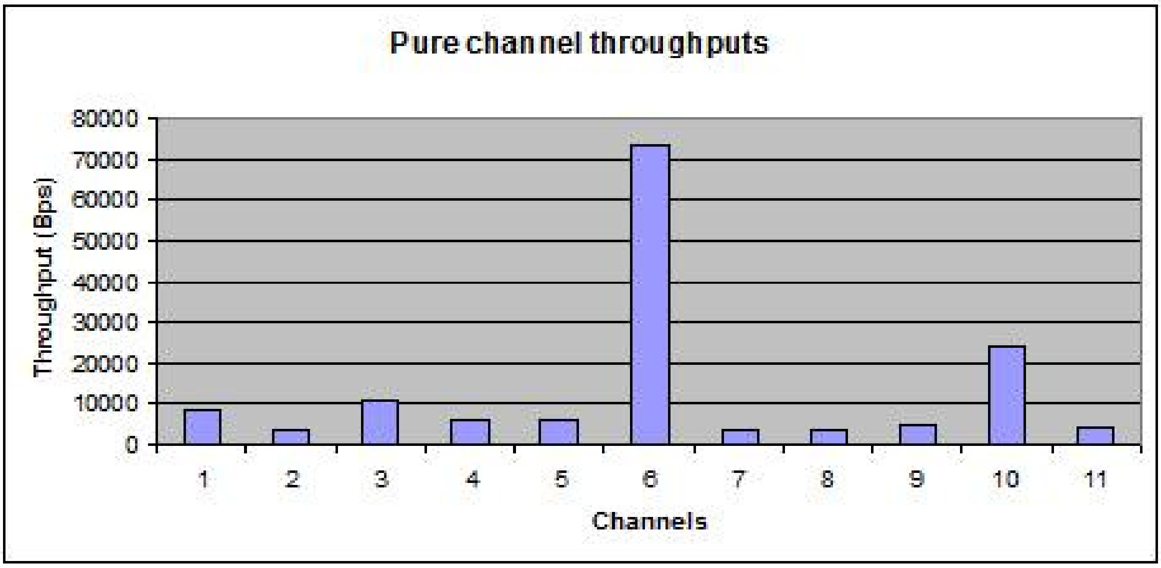

Figure 5.12 : Pure channel throughputs

Figure 5.12 shows the characteristics of the test environment excluding the generated test traffics. It could be seen that channel 6 is heavily used around the laboratory that the tests are performed.

Our cognitive AP uses this network throughputs and pure channel throughputs to evaluate the minimum total interference weight and decides that the networks should be using channels **1,7,1,11** respectively. It took the channel decision algorithm 1.606 seconds to decide on working channels on our implemented AP.

The complexity of the algorithm is

$$O(c^n) \quad \textbf{(5.5)}$$

where c is the number of channels and n is the number of networks.

c=11 and n=4 for our case.

The following are the comparions of the performace metrics for all channel selections.

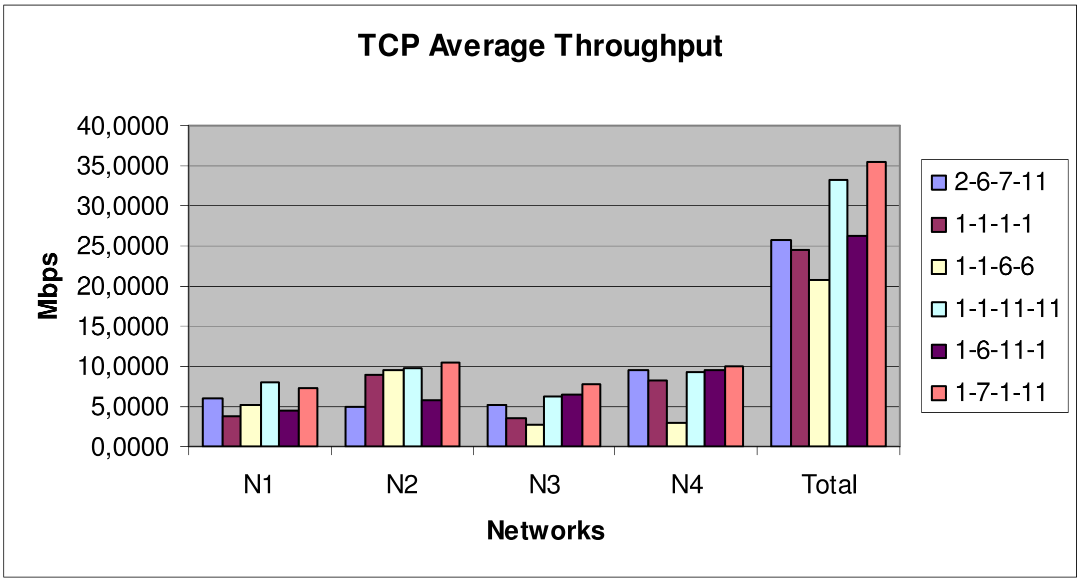

Figure 5.13 : TCP average throughput

Figure 5.13 shows that the total throughput for TCP is much better for our channel assignment compared to the other channel selections. It could also be seen from the figure that although there could be a channel selection that one network could have a better throughput on, the total throughput of the networks for that channel selection is not as good as the total throughput of our proposed channel assignments.

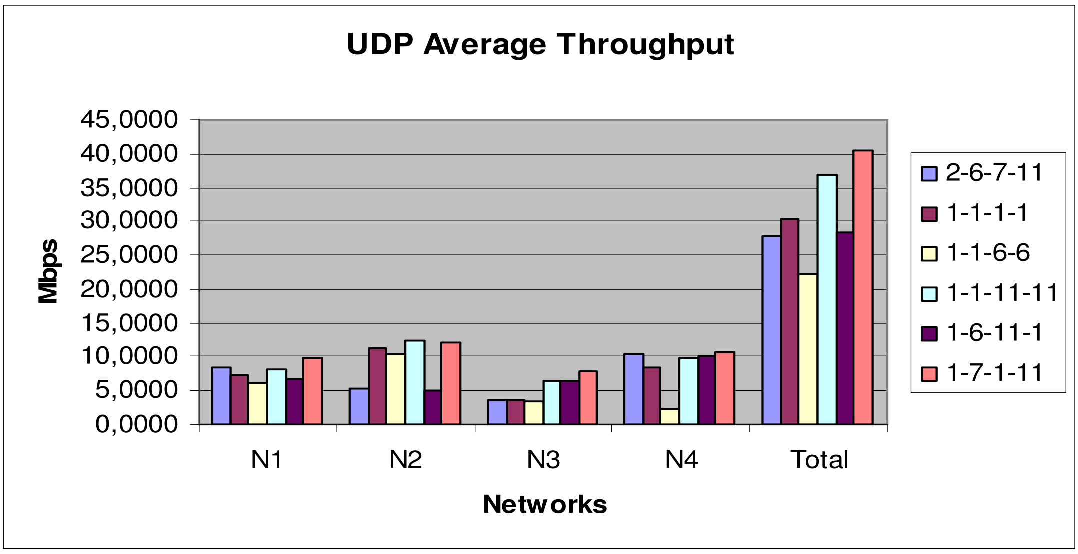

Figure 5.14 : UDP average throughput

UDP shows a similar performance to TCP by having a better throughput in total. UDP throughput performance is given in Figure 5.14. It could be seen from both figures that both performances are very bad for 1-1-6-6 selection whereas the performance for 1-1-11-11 selection is very good by having the closest results to our channel assignments. The performance for 1-1-6-6 selection is very poor, because two networks are using channel 6, although channel 6 is already heavily used in this test environment. The performance for 1-1-11-11 is better than many selections, because the networks are using orthogonal channels 1 and 11 that have less traffic load compared to 6 and they are working almost at the edges of the spectrum. This test case was an optimistic test case where the performance results are closest to our proposed system results.

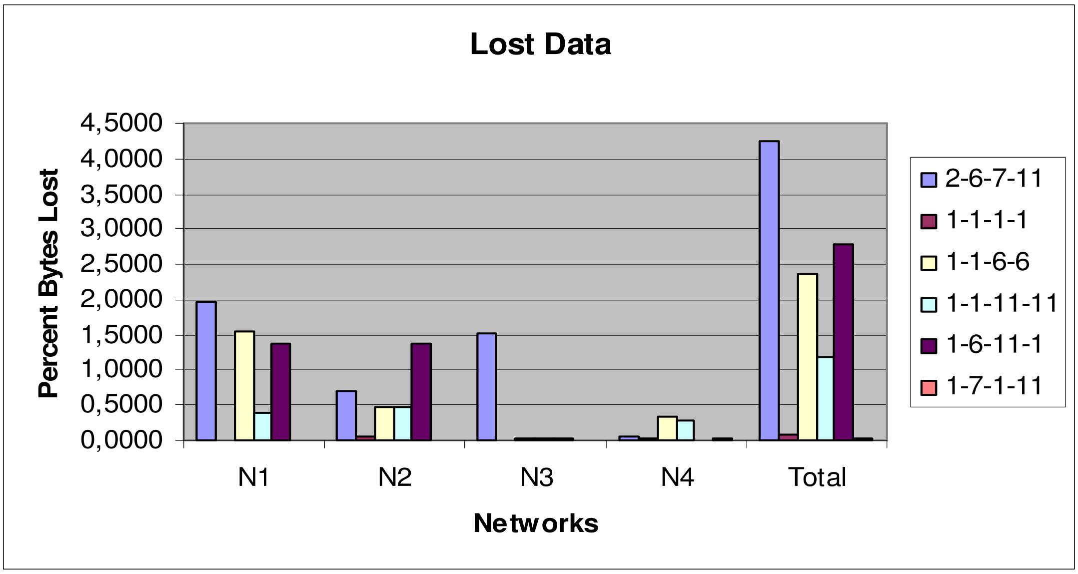

Figure 5.15 : Lost data for RTP

Figure 5.15 shows that the performance of our proposed system for lost data performance metric is again better than the other channel selections. Another point observed for lost data performance metric is that in the 3rd and 4th networks, which are close to windows and in the outer part of the building, the lost data percentage is very less compared to the 1st and 2nd networks, which are in the inner part of the building where there are more other networks that could create interference.

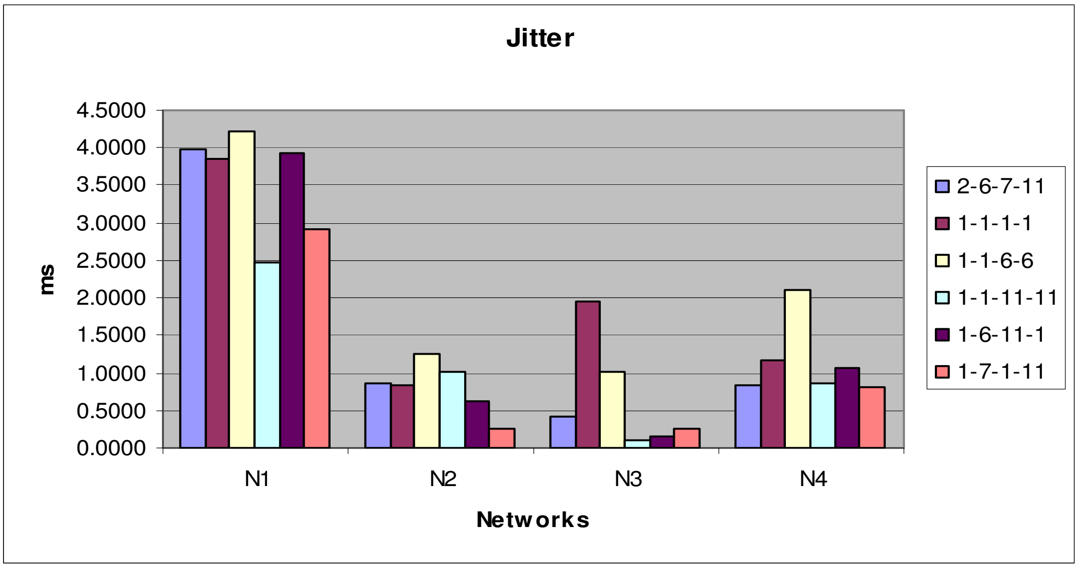

Figure 5.16 : Jitter for RTP

Figure 5.16 shows the jitter performance of the RTP traffic. Having less jitter is a good performance indicator for real time traffics. It could again be seen from the graph that the jitter performance of our proposed system is better in most of the networks.

The dynamic behaviour of the proposed system is also observed by adding some new traffic to the environment. The test results are given in Figure 5.17.

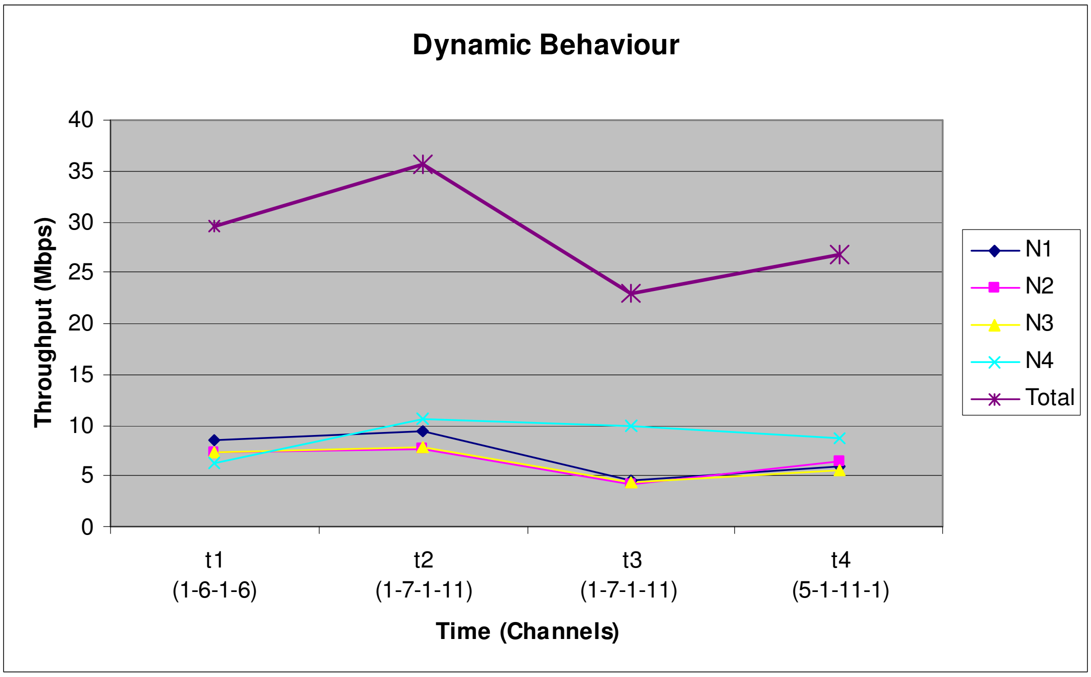

Figure 5.17 : Dynamic behavior for TCP throughput

The networks are selected to operate on random channels 1-6-1-6 initially while the traffics described in Section 5.2 are running on them. Then our cognitive AP assigns new channels 1-7-1-11 to the networks and the total TCP throughput increases after this assignment. At time t3, one new FTP traffic starts on Network 1 and one new HTTP Gif traffic starts on Network 3. Thus, the total TCP throughput decreases as these new traffics cause packet losses in Network 1 and Network 3 along with the neighbor networks that are operating on a channel close to the channel of Network 1 and Network 3. Then, our cognitive AP assigns new channels 5-1-11-1 to the networks regarding the new traffics in the environment and the total TCP throughput increases as it could be seen in Figure 5.17. It can not increase up to its previous value as there are more traffic loads in the environment now.

6. CONCLUSION

In this thesis, we implemented a cognitive AP that have the ability to sense the environment, analyze the sensed data and decide on the operating channel with regard to the sensed data to improve performance. In addition, we implemented this cognitive, dynamic channel assignment scheme regarding interference avoidance by targeting the total system performance.

We made observations in different environments to show how the performance degrades by the traffic load in channels and by the interference caused by the neighboring channels. Then we established a test bed to show the improvements obtained using the proposed scheme. The test results showed that the proposed technique increases throughput while balancing it among the networks. The proposed method also decreases jitter and packet loss.

For future work, the performance of the channel assignment scheme could be studied regarding to the QoS requirements of the network traffic. In order to satisfy QoS requirements of the connections, we should pay attention to the current network traffic and the interference caused by the neighboring networks. Currently, the effect of the interference caused by the neighboring networks is evaluated using COF. More precise evaluation of interference may help us about the channel selection when multiple connections with different QoS requirements arrive at the same time. Another future work which we target to improve the proposed system is to use more complicated hardware to remove the drawback of the proposed system related to the channel switching duration.

REFERENCES

[1] **Akyıldız, I. F., Lee, W., Vuran, M. C., and Mohanty, S.,** 2006. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey, *Elsevier Computer Networks*, **50**, 2127-2159.

[2] **Su, H., and Zhang, X.,** 2007. Cognitive radio based multi-channel MAC protocols for wireless ad hoc networks, *IEEE Globecom 2007*, Washington, USA, 4857-4861.

[3] **Le, B., et al.,** 2007. A public safety cognitive radio node, *2007 SDR Forum Technical Conference*, Denver, USA, 1-7.

[4] **Hoang, A. T., and Liang, Y.,** 2006. A two phase channel and power allocation scheme for cognitive radio networks, *IEEE PIMRC'06*, Helsinki, Finland, 1-5.

[5] **Manoj, B. S., Rao, R. R., and Zorzi, M.,** 2007. On the use of higher layer information for cognitive networking, *IEEE Globecom 2007*, Washington, USA, 3568-3573.

[6] **Villegas, E. G., et al.,** 2007. Effect of adjacent-channel interference in IEEE 802.11 WLANs, *IEEE CROWNCOM 2007*, Orlando, USA, 1-8.

[7] **Hills, A.,** 2001. Large-scale wireless LAN design, *IEEE Communications Magazine*, **39**, 98-107.

[8] <*http://www.cisco.com/en/US/i/100001-200000/120001-130000/121001-122000/121140.jpg* >, last accessed at 24.12.2009

[9] **Leskaroski, D., and Mikhael, W. B.,** 2006. Frequency planning and adjacent channel interference in a DSSS wireless local area network (WLAN), *Wireless Personal Communications*, 169-180, Springer Netherlands.

[10] **Hong, X., Wang, C., and Thompson J.,** 2008. Interference modeling of cognitive radio networks, *IEEE VTC 2008*, Marina Bay, Singapore, 1851-1855.

[11] **Rahman, M. J., Wang, X., and Primak S. L.,** 2009. Efficient mutual interference minimization and power allocation for OFDM-based cognitive radio, *IEEE Globecom 2009*, Hawaii, USA, 1-6.

[12] **Tamma, B. R., Manoj, B. S., and Rao R.,** 2009.An autonomous cognitive access point for Wi-Fi hotspots, *IEEE Globecom 2009*, Hawaii, USA, 1-6.

[13] <*http://madwifi-project.org*>, last accessed at 24.12.2009.

[14] <*http://wiki.wireshark.org*>, last accessed at 24.12.2009.

[15] <*http://www.wireshark.org/docs/man-pages/tshark.html*>, last accessed at 24.12.2009.

[16] **Orebaugh, A., et al.,** 2007. *Wireshark & Etheral Network Protocol Analyzer Toolkit*, Syngress Publishing, 267-370.

[17] <*http://www.dlink.com/products/?pid=550*>, last accessed at 24.12.2009.

[18] <*http://download.netiq.com/products/pe/documentation/NetIQ_PE_Microsoft_Windows_NT_2000_XP_2003.pdf*>, last accessed at 24.12.2009.

[19] <*http://www.ietf.org/rfc/rfc1889.txt*>, last accessed at 24.12.2009.

APPENDICES

APPENDIX A.1 : Traffic Scripts
APPENDIX A.2 : Performance Test Graphs

APPENDIX A.1: Traffic Scripts

Line	Endpoint 1	Endpoint 2
1	SLEEP	
2	time = initial_delay (0)	
3	LOOP	LOOP
4	count = number_of_timing_records (500)	count = number_of_timing_records (500)
5	START_TIMER	
6	CONNECT_INITIATE	CONNECT_ACCEPT
7	port = source_port (AUTO)	port = destination_port (AUTO)
8	RECEIVE	SEND
9	size = file_control_size (70)	size = file_control_size (70)
10	buffer = receive_buffer_size (1024)	buffer = file_control_size (70)
11		type = control_datatype (trans.cmp)
12		rate = send_data_rate (UNLIMITED)
13	SEND	RECEIVE
14	size = reply_size (20)	size = reply_size (20)
15	buffer = reply_size (20)	buffer = server_buffer_size (8192)
16	type = control_datatype (trans.cmp)	
17	rate = send_data_rate (UNLIMITED)	
18	RECEIVE	SEND
19	size = reply_size (20)	size = reply_size (20)
20	buffer = receive_buffer_size (1024)	buffer = reply_size (20)
21		type = control_datatype (trans.cmp)
22		rate = send_data_rate (UNLIMITED)
23	SEND	RECEIVE
24	size = reply_size (20)	size = reply_size (20)
25	buffer = reply_size (20)	buffer = server_buffer_size (8192)
26	type = control_datatype (trans.cmp)	
27	rate = send_data_rate (UNLIMITED)	
28	RECEIVE	SEND
29	size = file_control_size (70)	size = file_control_size (70)
30	buffer = receive_buffer_size (1024)	buffer = file_control_size (70)
31		type = control_datatype (trans.cmp)
32		rate = send_data_rate (UNLIMITED)
33	SEND	RECEIVE
34	size = size_of_record_to_send (6)	size = size_of_record_to_send (6)
35	buffer = size_of_record_to_send (6)	buffer = server_buffer_size (8192)
36	type = control_datatype (trans.cmp)	
37	rate = send_data_rate (UNLIMITED)	
38	RECEIVE	SEND
39	size = reply_size (20)	size = reply_size (20)
40	buffer = receive_buffer_size (1024)	buffer = reply_size (20)
41		type = control_datatype (trans.cmp)
42		rate = send_data_rate (UNLIMITED)
43	LOOP	LOOP

Figure A.1 : POP3 e-mail transfer script -Part1-

count = transactions_per_record (5)	count = transactions_per_record (5)
SEND	RECEIVE
size = size_of_record_to_send (6)	size = size_of_record_to_send (6)
buffer = size_of_record_to_send (6)	buffer = server_buffer_size (8192)
type = control_datatype (trans.cmp)	
rate = send_data_rate (UNLIMITED)	
RECEIVE	SEND
size = reply_size (20)	size = reply_size (20)
buffer = receive_buffer_size (1024)	buffer = reply_size (20)
	type = control_datatype (trans.cmp)
	rate = send_data_rate (UNLIMITED)
RECEIVE	SEND
size = file_size (1000)	size = file_size (1000)
buffer = receive_buffer_size (1024)	buffer = file_size (1000)
	type = send_datatype (news.cmp)
	rate = send_data_rate (UNLIMITED)
SEND	RECEIVE
size = size_of_record_to_send (6)	size = size_of_record_to_send (6)
buffer = size_of_record_to_send (6)	buffer = server_buffer_size (8192)
type = control_datatype (trans.cmp)	
rate = send_data_rate (UNLIMITED)	
RECEIVE	SEND
size = reply_size (20)	size = reply_size (20)
buffer = receive_buffer_size (1024)	buffer = reply_size (20)
	type = control_datatype (trans.cmp)
	rate = send_data_rate (UNLIMITED)
INCREMENT_TRANSACTION	
END_LOOP	END_LOOP
SEND	RECEIVE
size = size_of_record_to_send (6)	size = size_of_record_to_send (6)
buffer = size_of_record_to_send (6)	buffer = server_buffer_size (8192)
type = control_datatype (trans.cmp)	
rate = send_data_rate (UNLIMITED)	
RECEIVE	SEND
size = file_control_size (70)	size = file_control_size (70)
buffer = receive_buffer_size (1024)	buffer = file_control_size (70)
	type = control_datatype (trans.cmp)
	rate = send_data_rate (UNLIMITED)
DISCONNECT	DISCONNECT
type = close_type (Reset)	type = close_type (Reset)
END_TIMER	
SLEEP	
time = transaction_delay (100)	
END_LOOP	END_LOOP

Figure A.2 : POP3 e-mail transfer script -Part2-

APPENDIX A.2: Performance Test Graphs

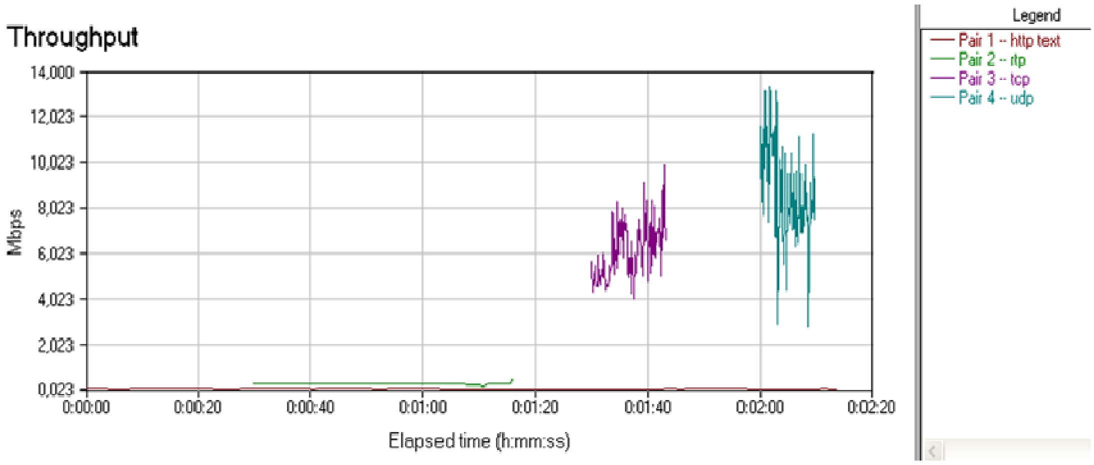

Figure A.3 : 2-6-7-11 channel selection -N1- all

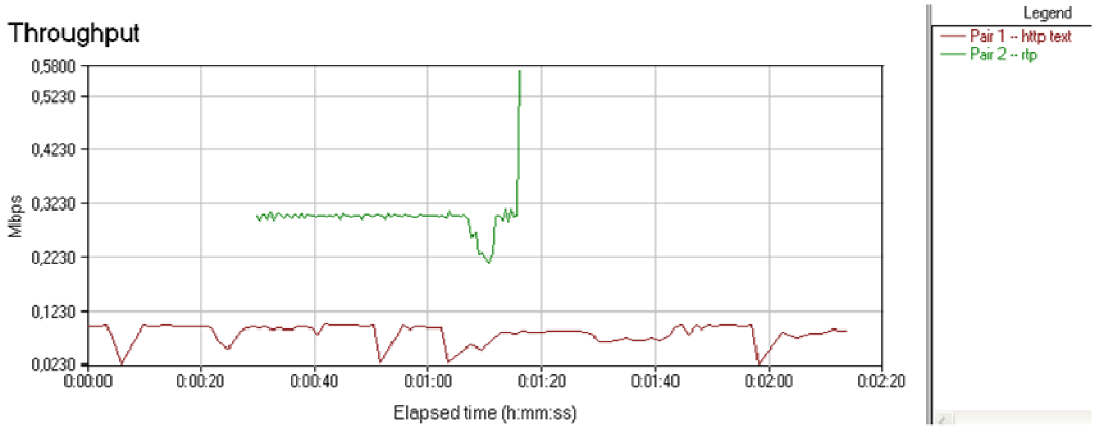

Figure A.4 : 2-6-7-11 channel selection -N1- rtp

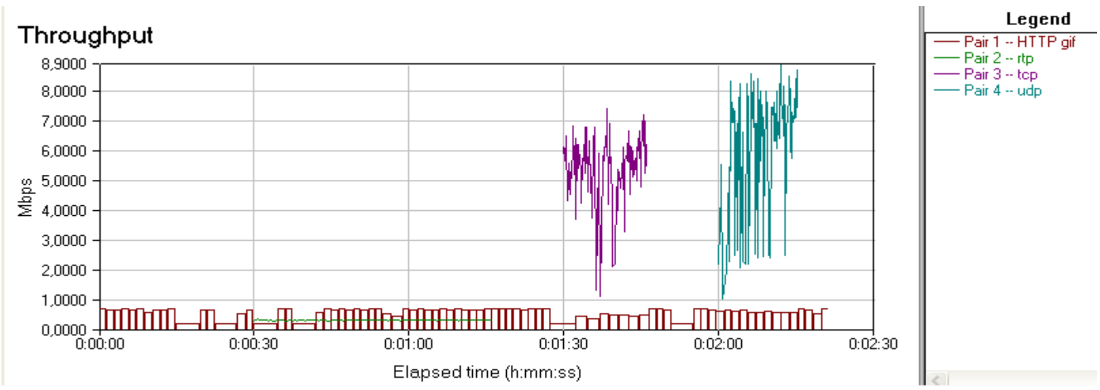

Figure A.5 : 2-6-7-11 channel selection -N2- all

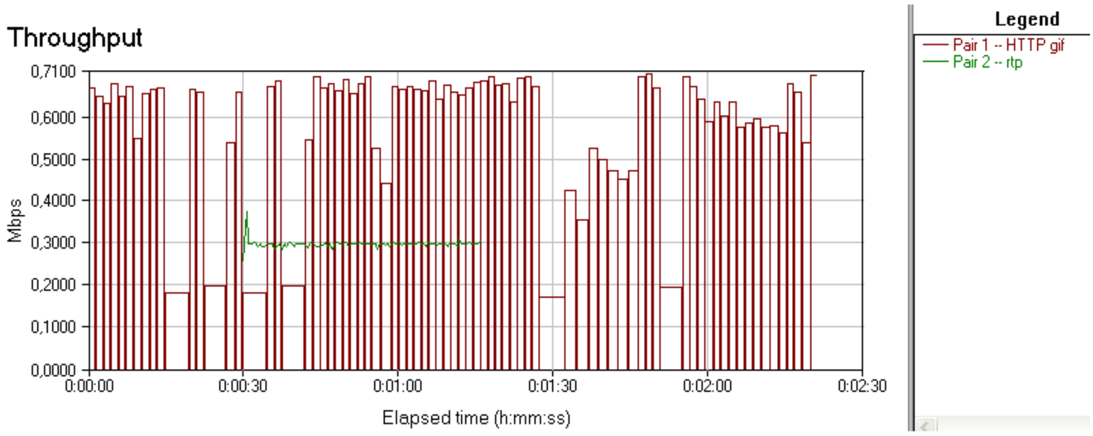

Figure A.6 : 2-6-7-11 channel selection -N2- all rtp

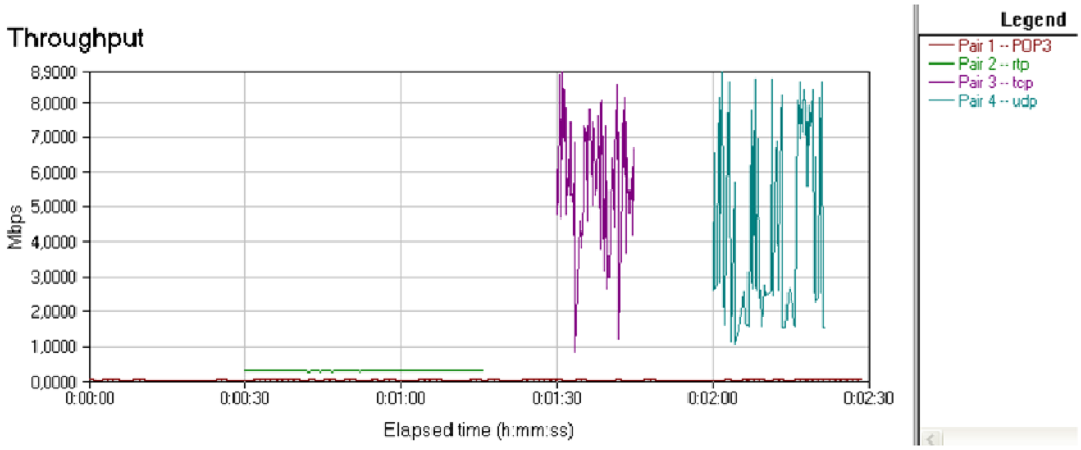

Figure A.7 : 2-6-7-11 channel selection -N3- all

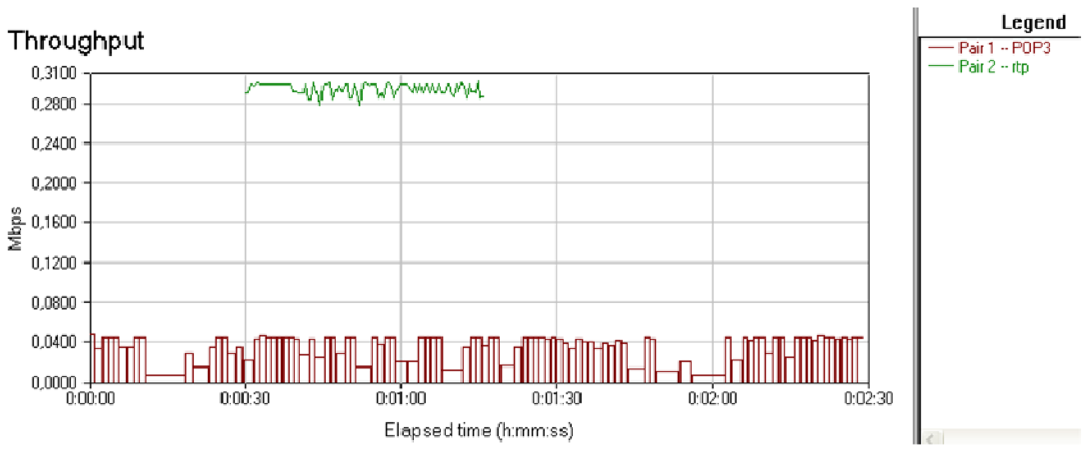

Figure A.8 : 2-6-7-11 channel selection -N3- rtp

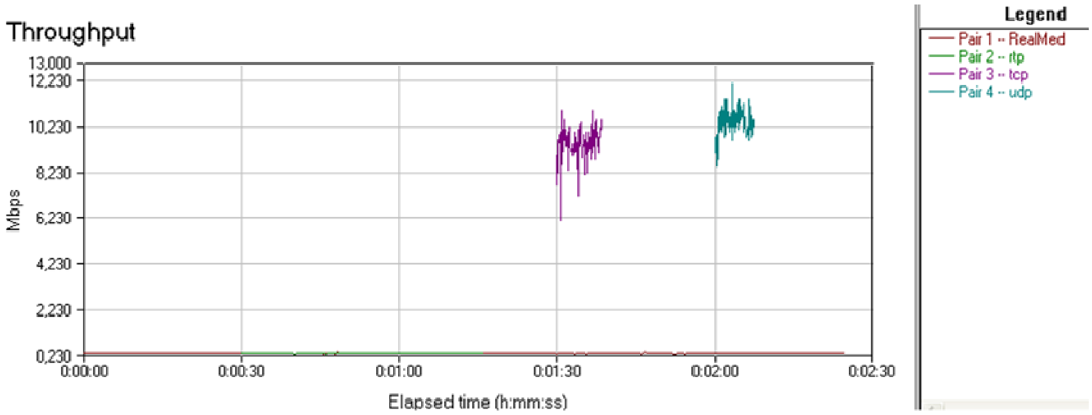

Figure A.9 : 2-6-7-11 channel selection -N4- all

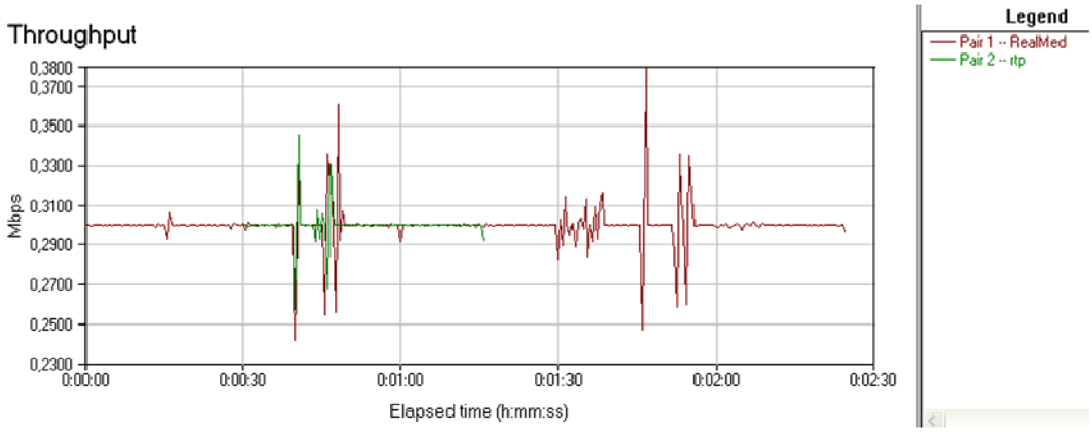

Figure A.10 : 2-6-7-11 channel selection -N4- rtp

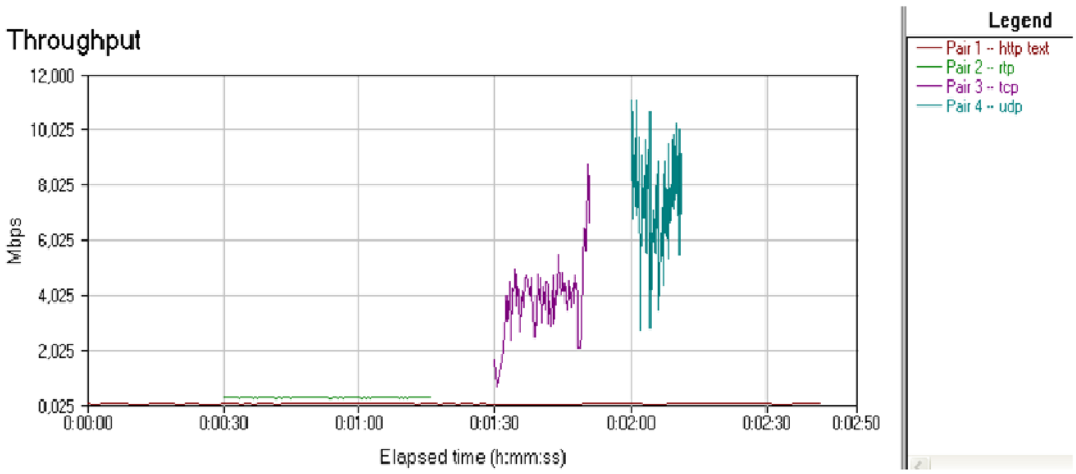

Figure A.11 : 1-1-1-1 channel selection -N1- all

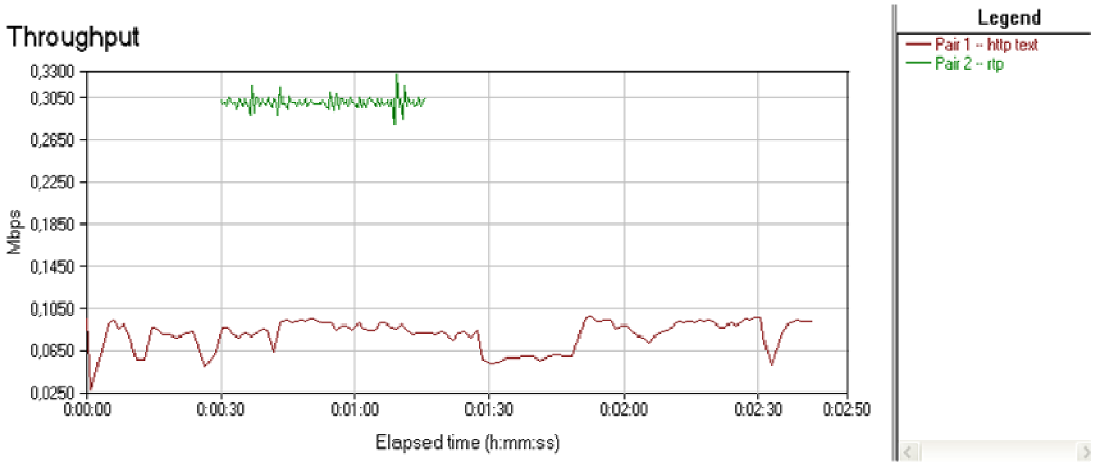

Figure A.12 : 1-1-1-1 channel selection -N1- rtp

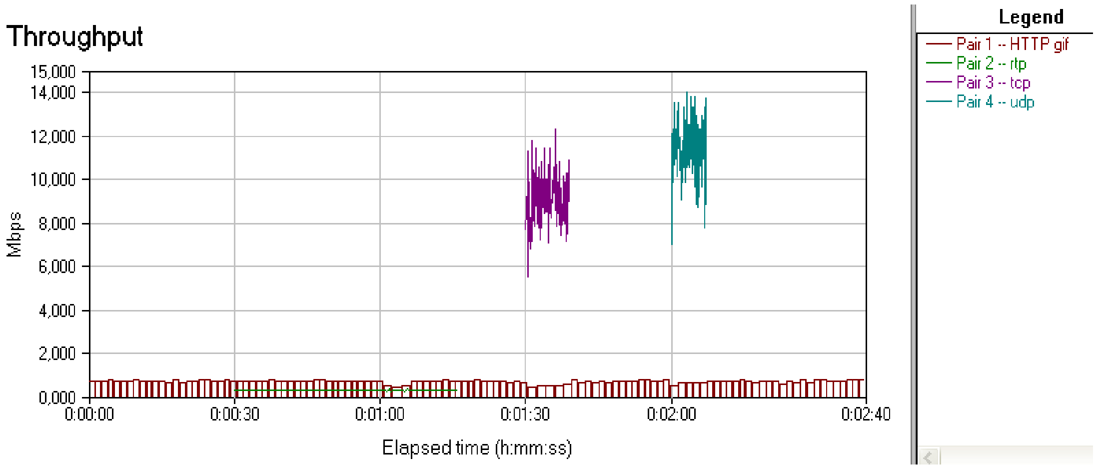

Figure A.13 : 1-1-1-1 channel selection -N2- all

Figure A.14 : 1-1-1-1 channel selection -N2- rtp

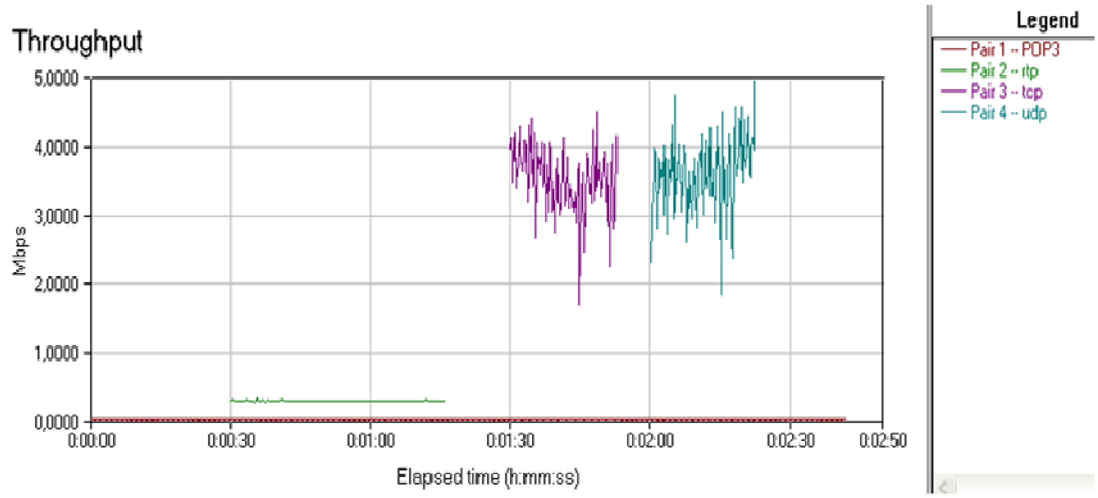

Figure A.15 : 1-1-1-1 channel selection -N3- all

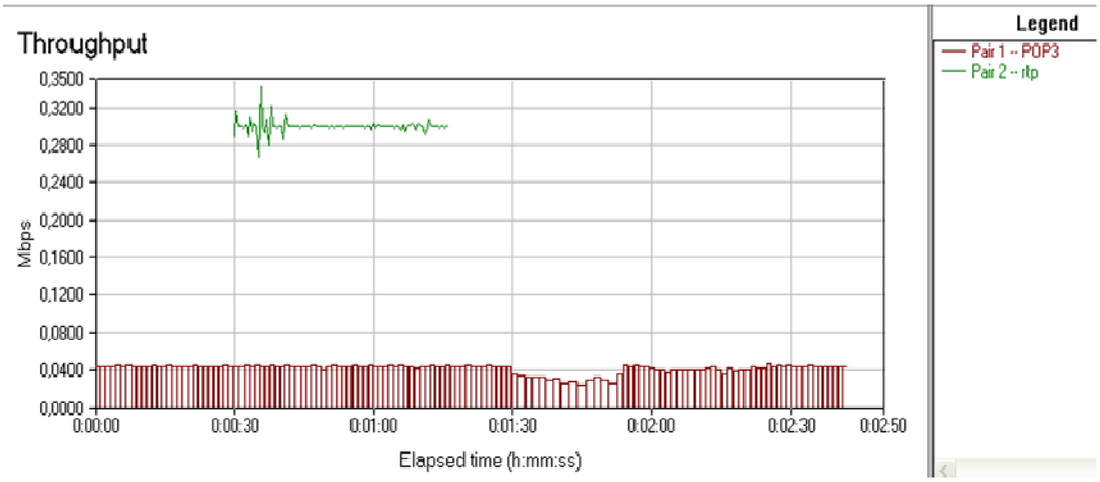

Figure A.16 : 1-1-1-1 channel selection -N3- rtp

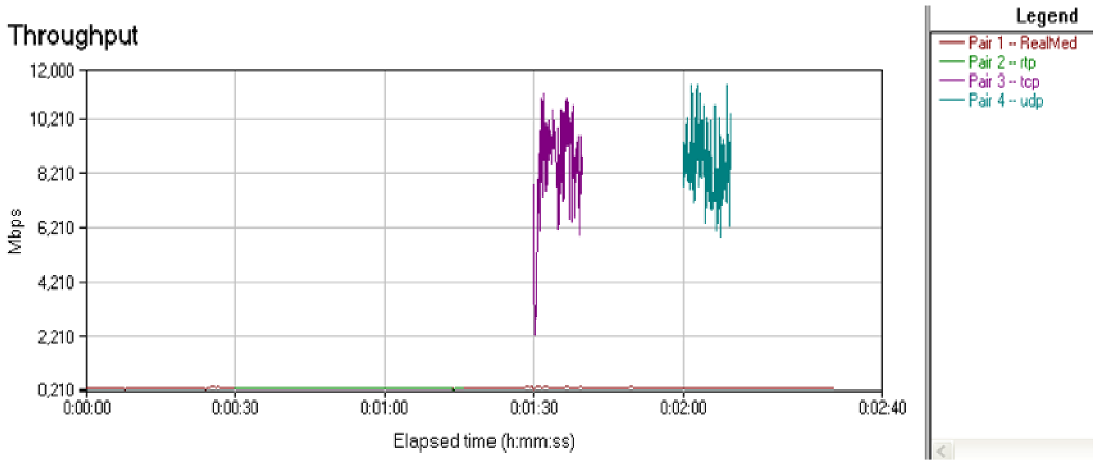

Figure A.17 : 1-1-1-1 channel selection -N4- all

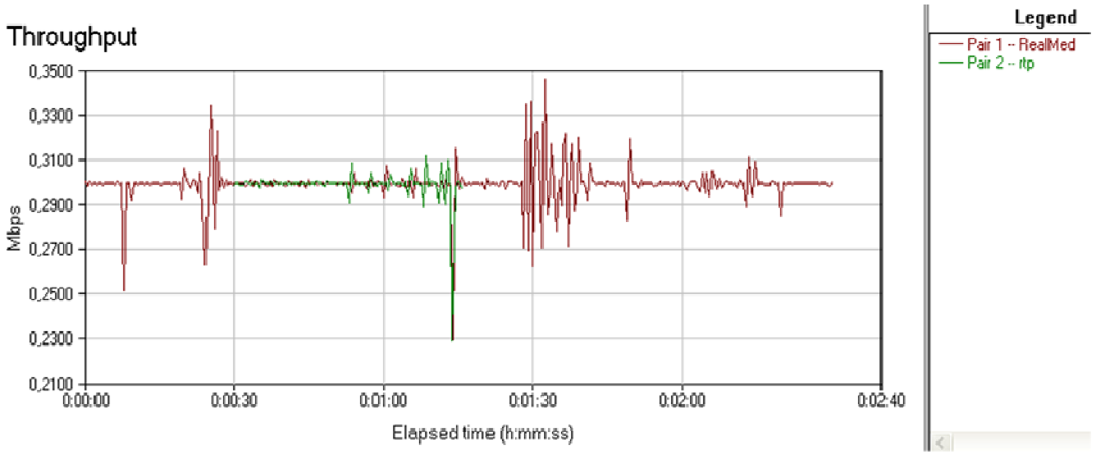

Figure A.18 : 1-1-1-1 channel selection -N4- rtp

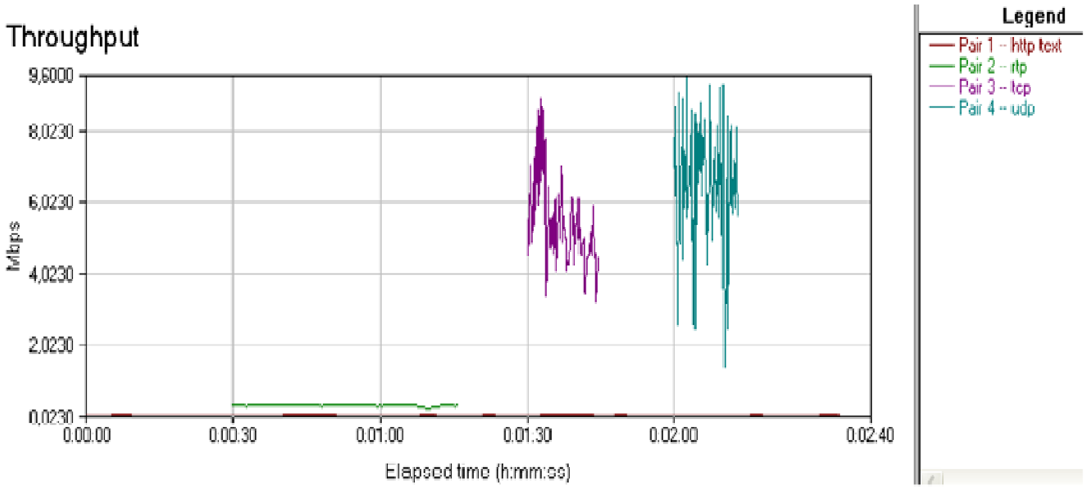

Figure A.19 : 1-1-6-6 channel selection -N1- all

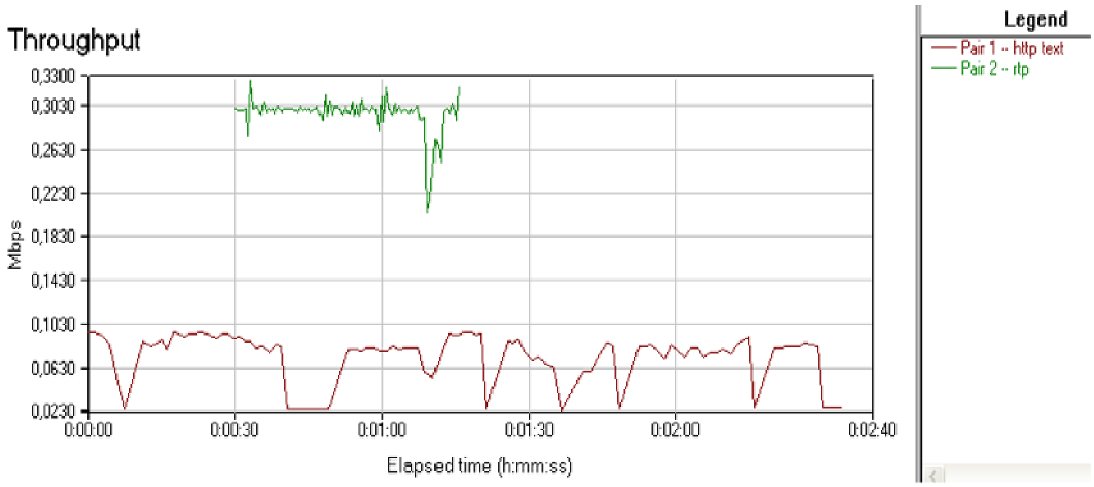

Figure A.20 : 1-1-6-6 channel selection -N1- rtp

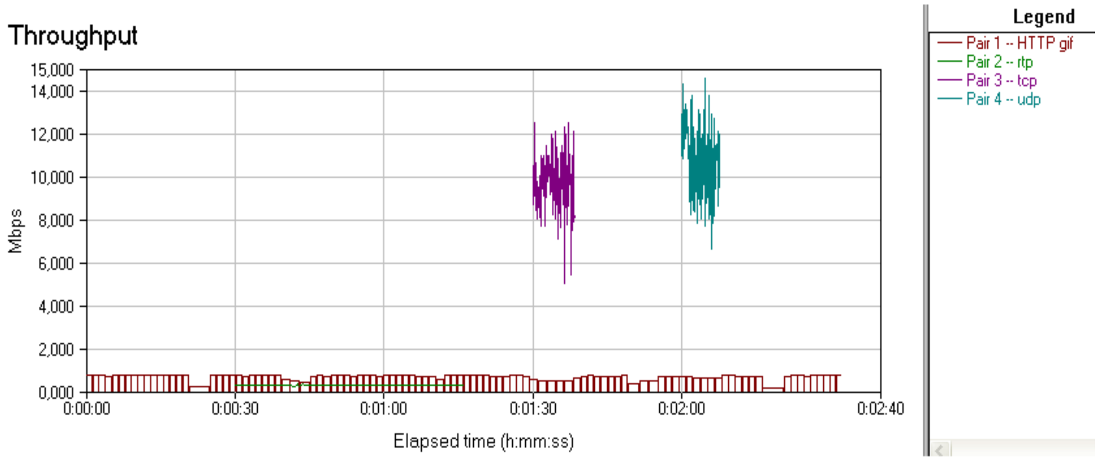

Figure A.21 : 1-1-6-6 channel selection -N2- all

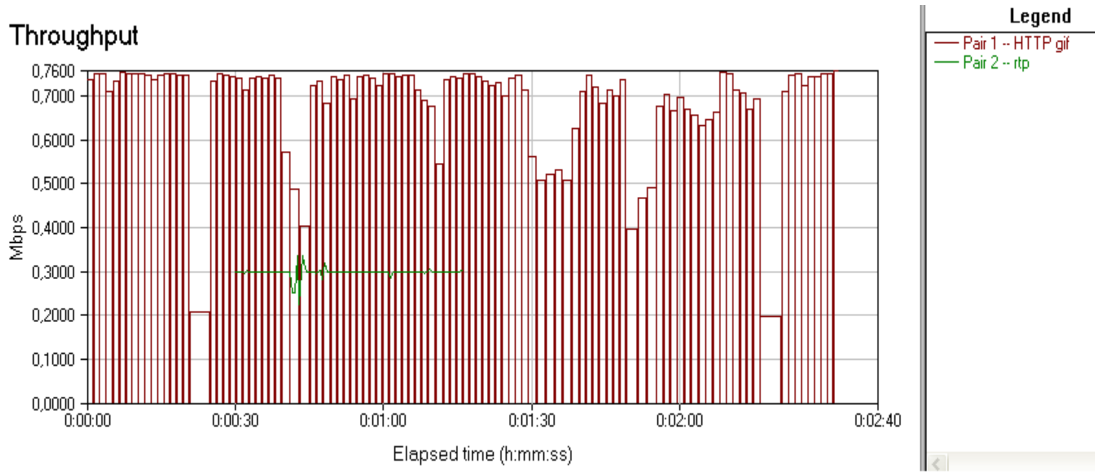

Figure A.22 : 1-1-6-6 channel selection -N2- rtp

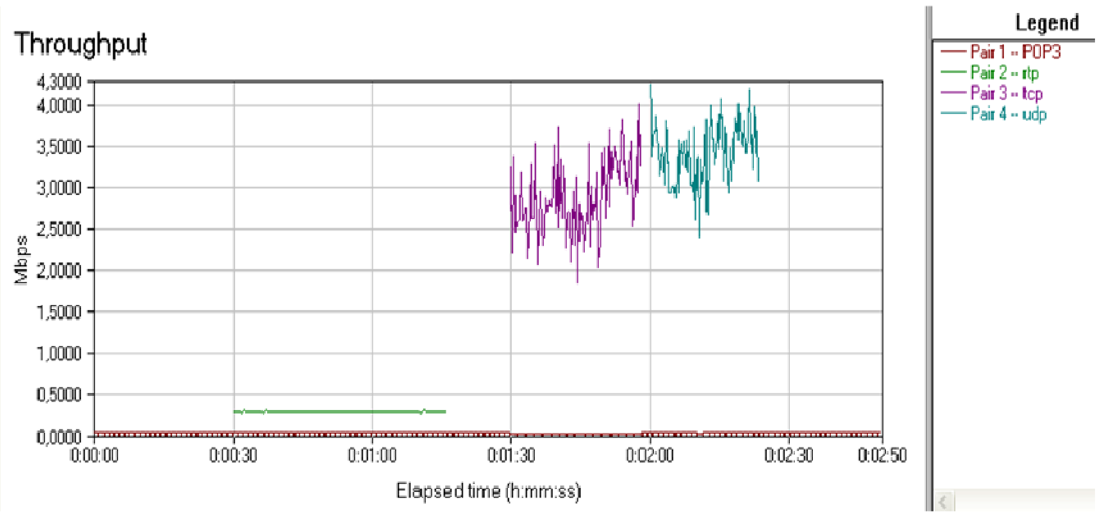

Figure A.23 : 1-1-6-6 channel selection -N3- all

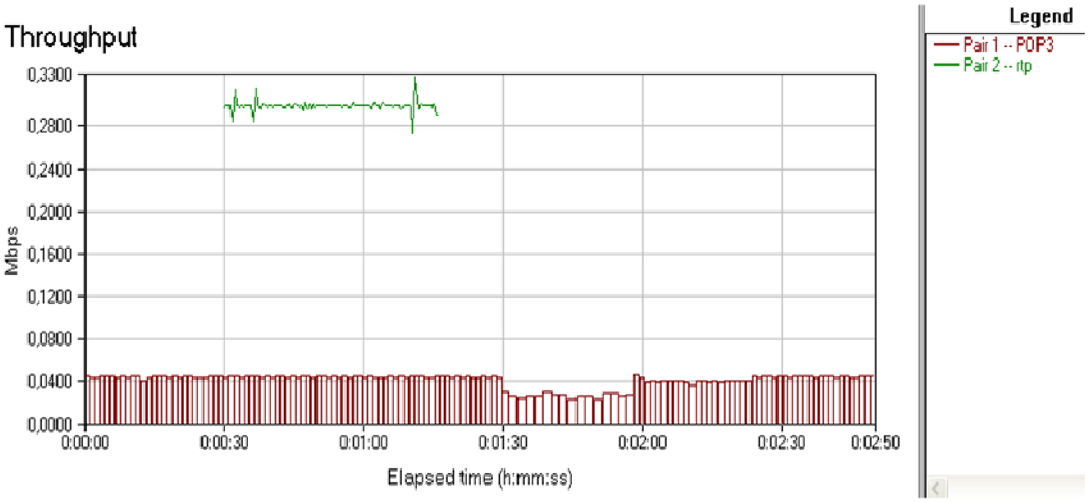

Figure A.24 : 1-1-6-6 channel selection -N3- rtp

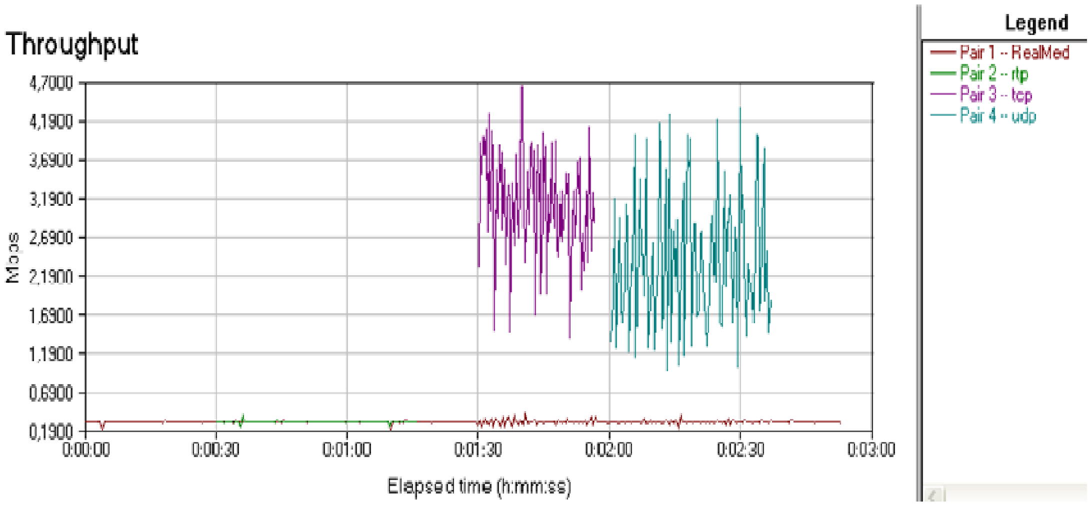

Figure A.25 : 1-1-6-6 channel selection -N4- all

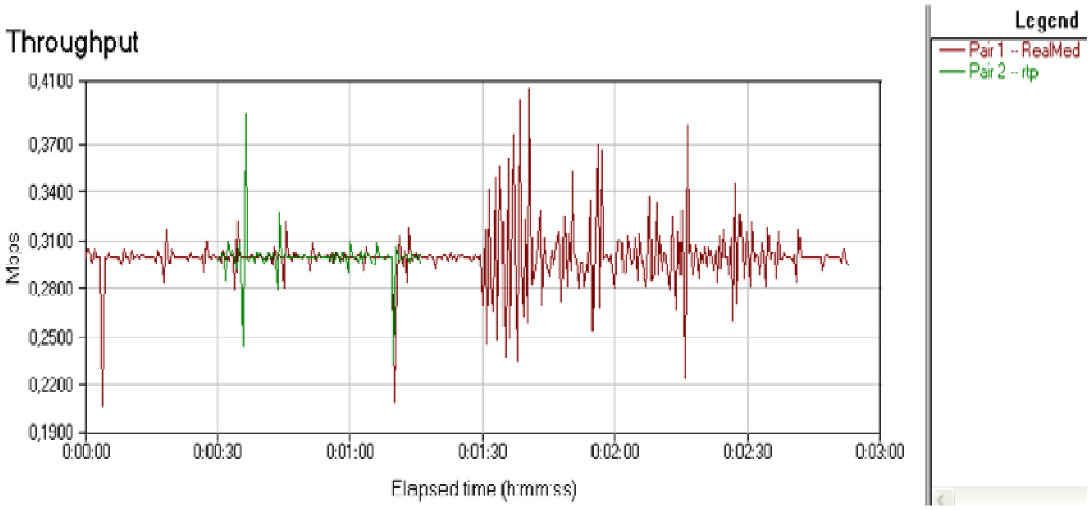

Figure A.26 : 1-1-6-6 channel selection -N4- rtp

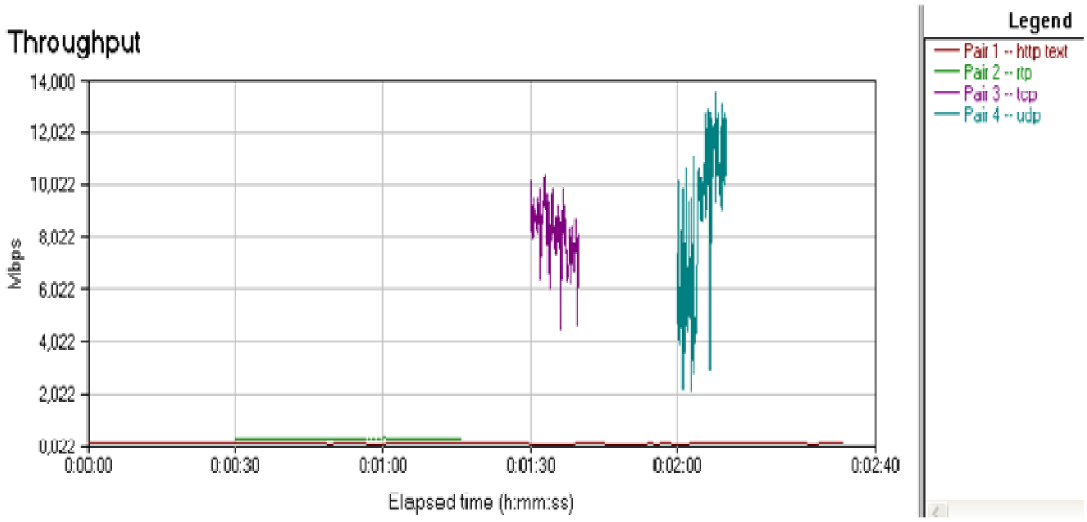

Figure A.27 : 1-1-11-11 channel selection -N1- all

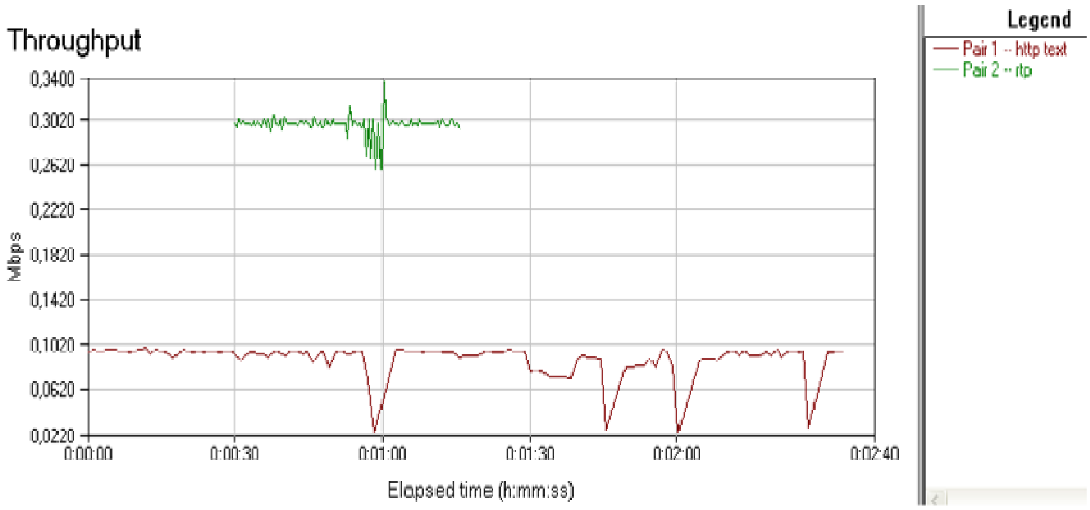

Figure A.28 : 1-1-11-11 channel selection -N1- rtp

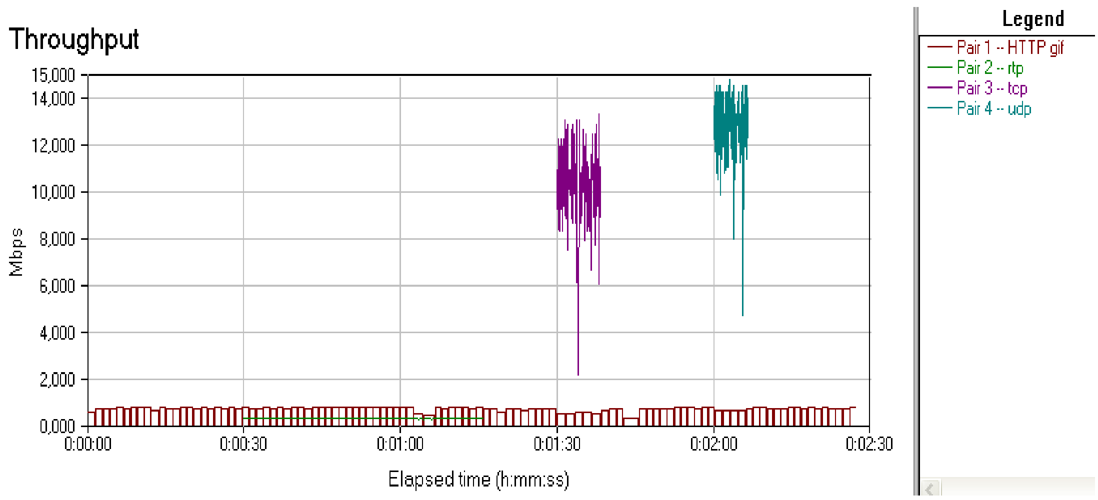

Figure A.29 : 1-1-11-11 channel selection -N2- all

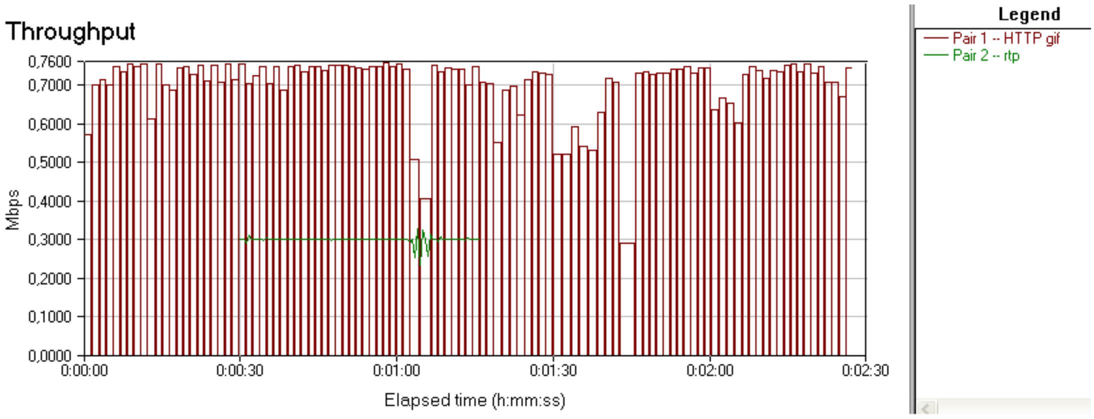

Figure A.30 : 1-1-11-11 channel selection -N2- rtp

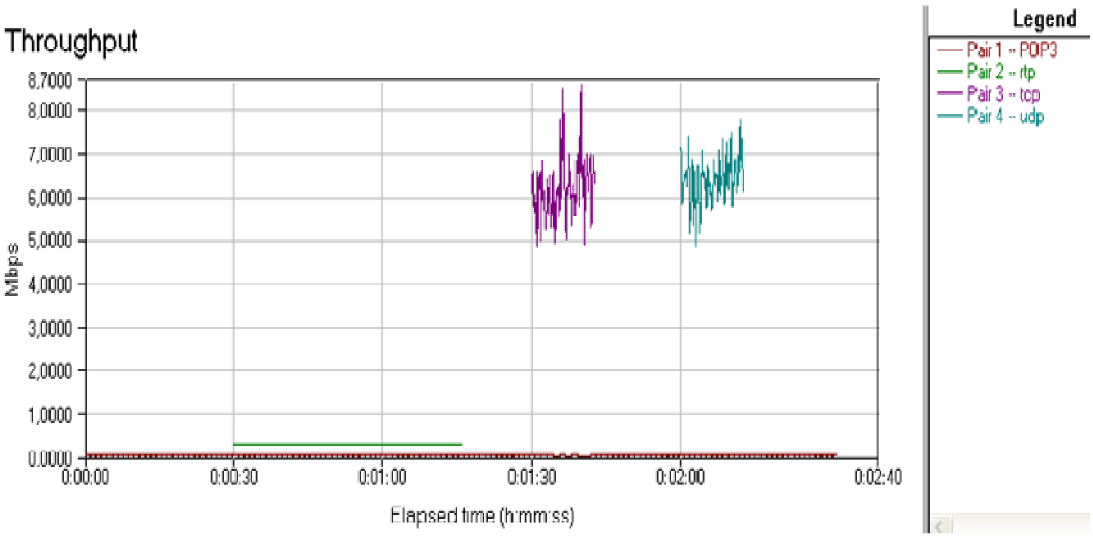

Figure A.31 : 1-1-11-11 channel selection -N3- all

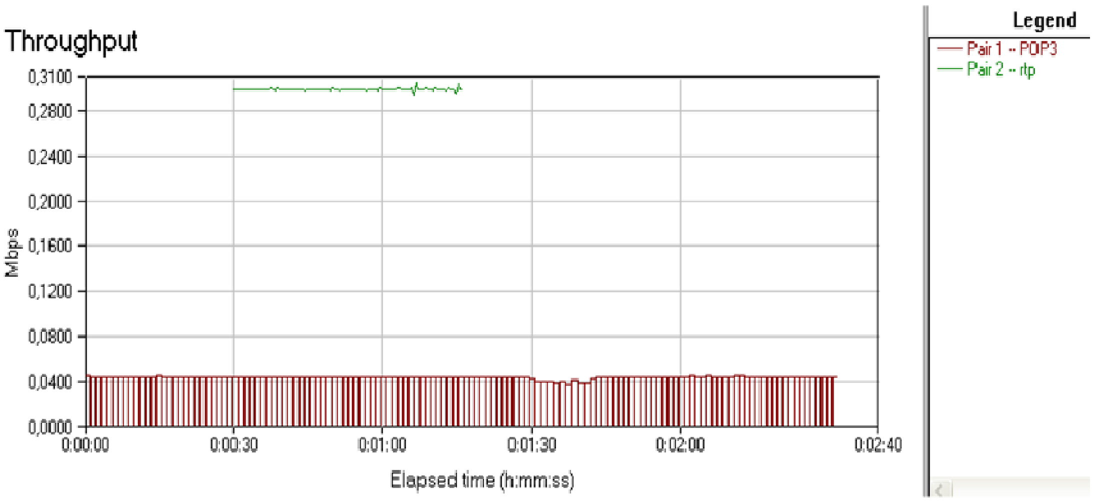

Figure A.32 : 1-1-11-11 channel selection -N3- rtp

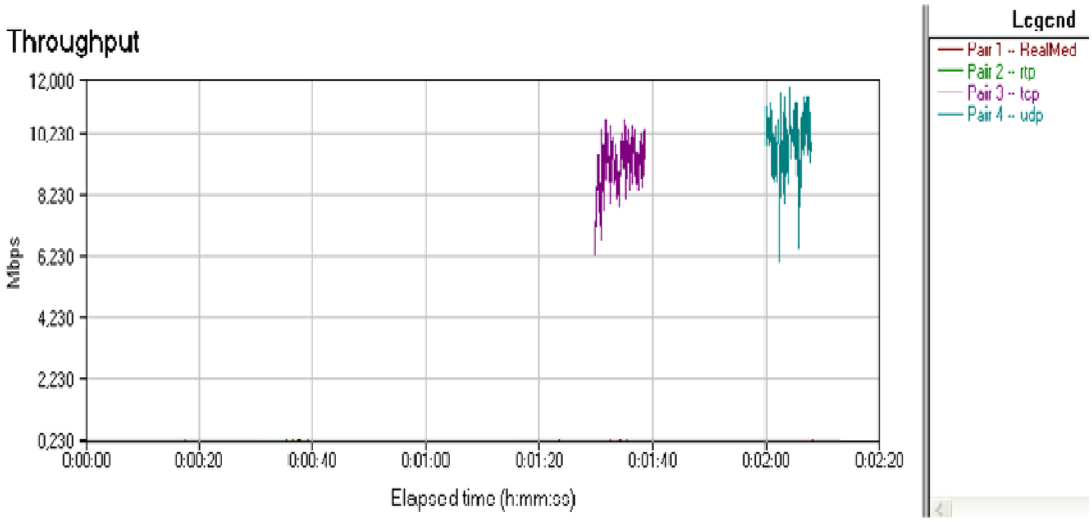

Figure A.33 : 1-1-11-11 channel selection -N4- all

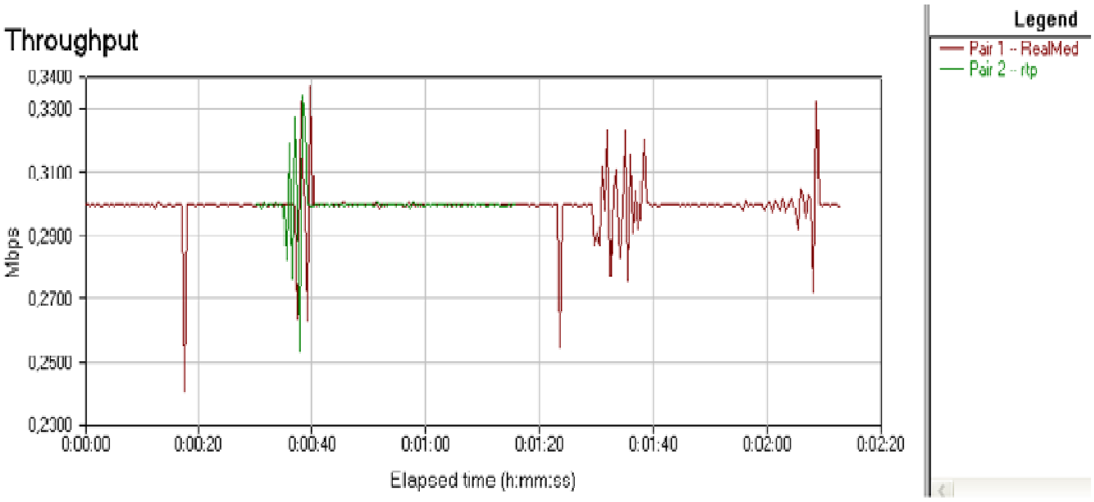

Figure A.34 : 1-1-11-11 channel selection -N4- rtp

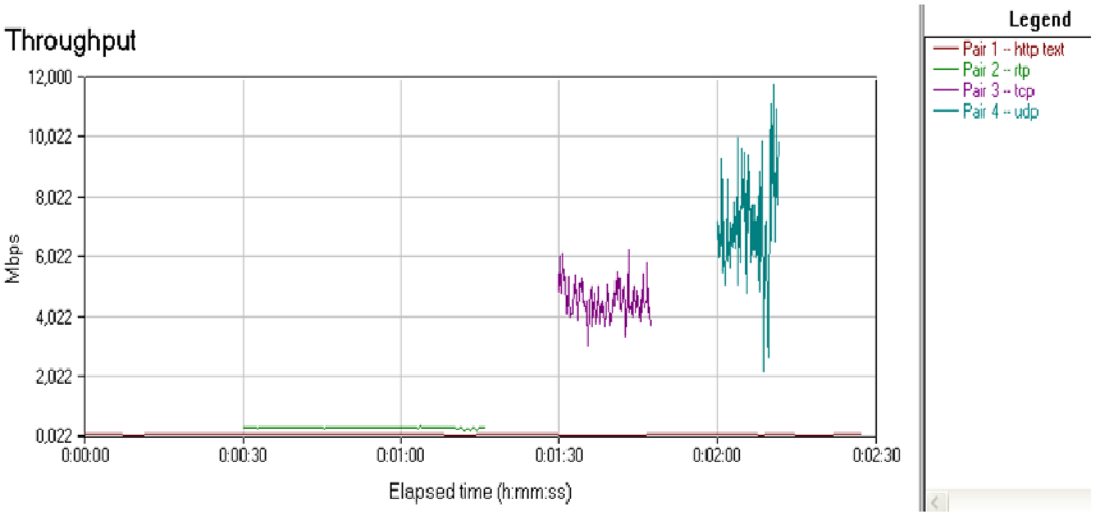

Figure A.35 : 1-6-11-1 channel selection -N1- all

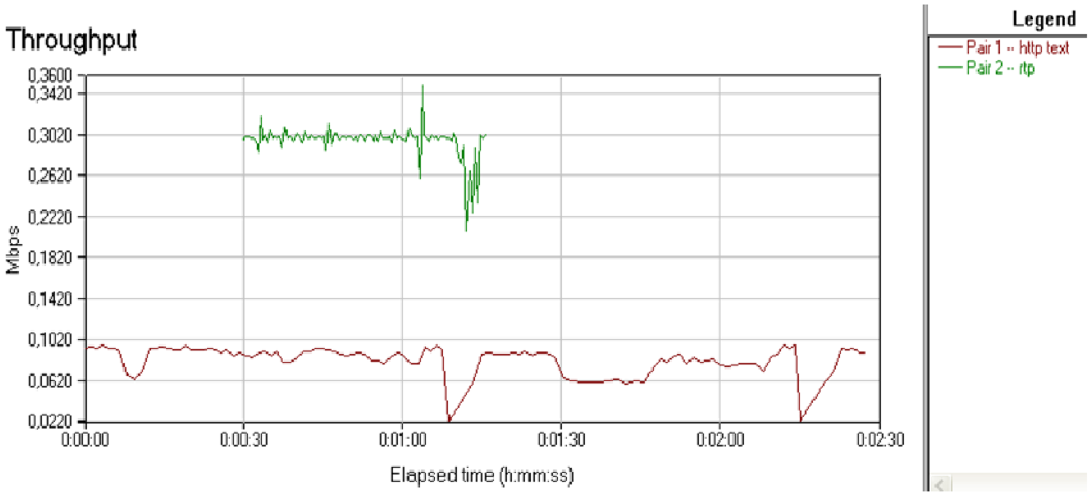

Figure A.36 : 1-6-11-1 channel selection -N1- rtp

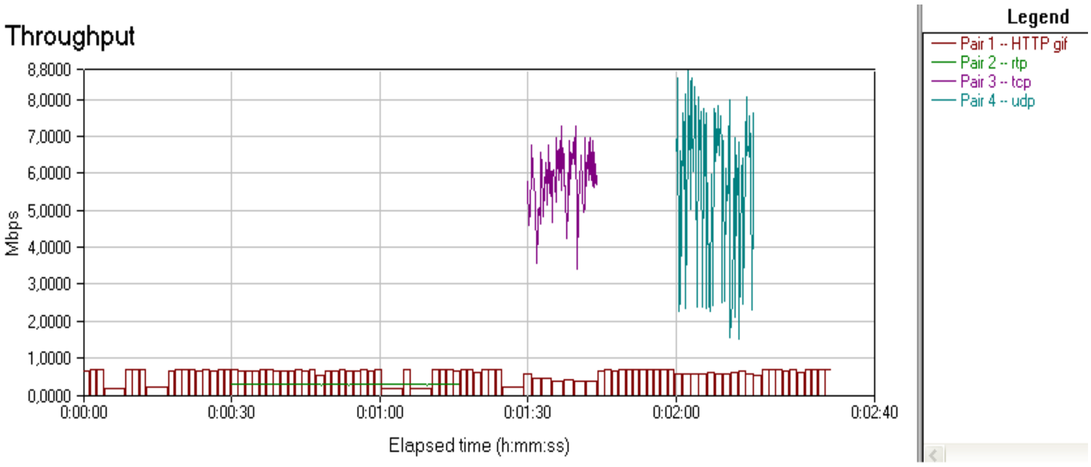

Figure A.37 : 1-6-11-1 channel selection -N2- all

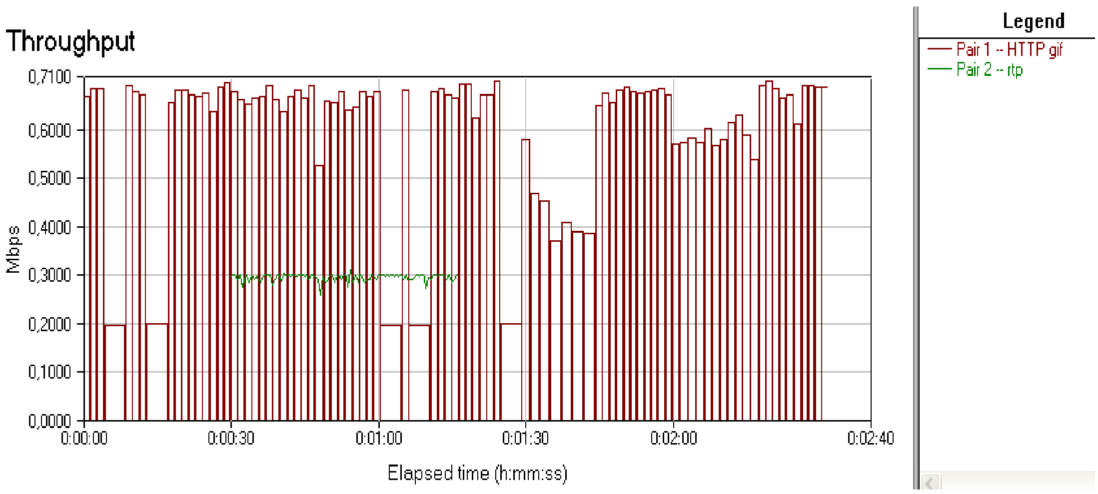

Figure A.38 : 1-6-11-1 channel selection -N2- rtp

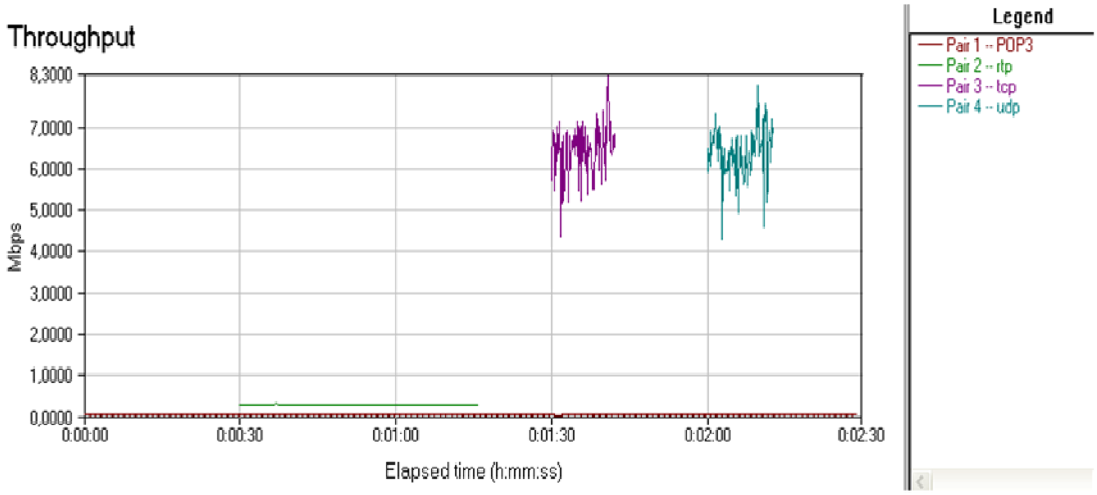

Figure A.39 : 1-6-11-1 channel selection -N3- all

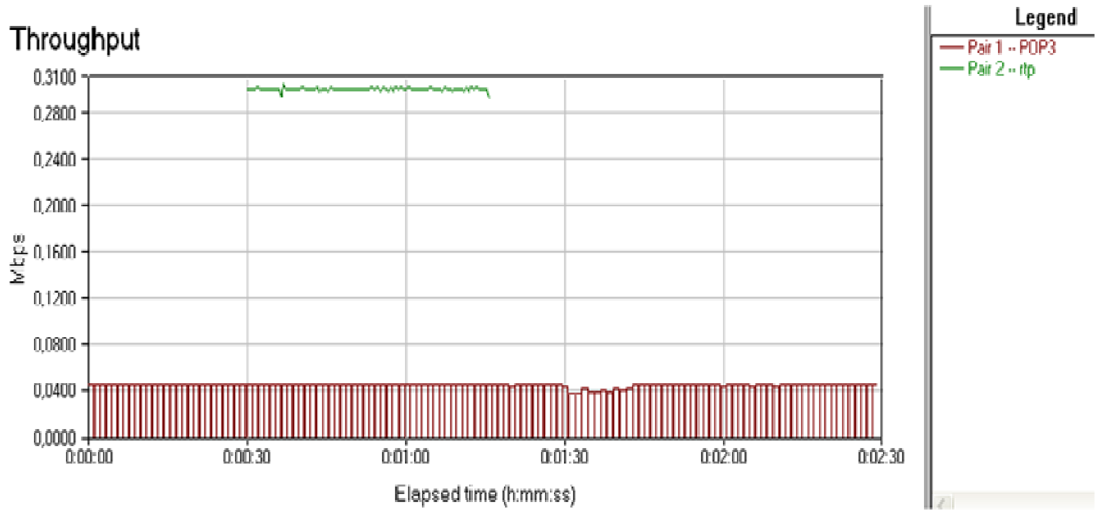

Figure A.40 : 1-6-11-1 channel selection -N3- rtp

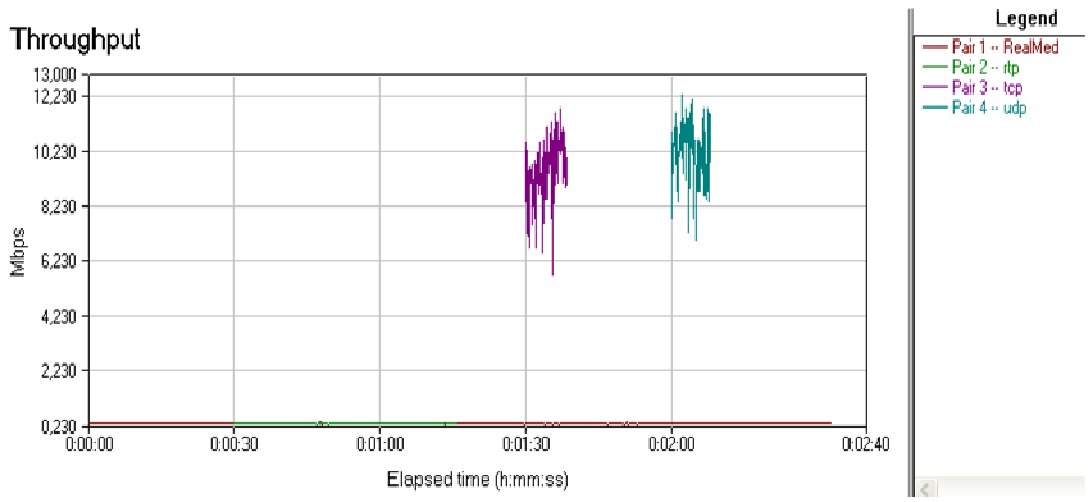

Figure A.41 : 1-6-11-1 channel selection -N4- all

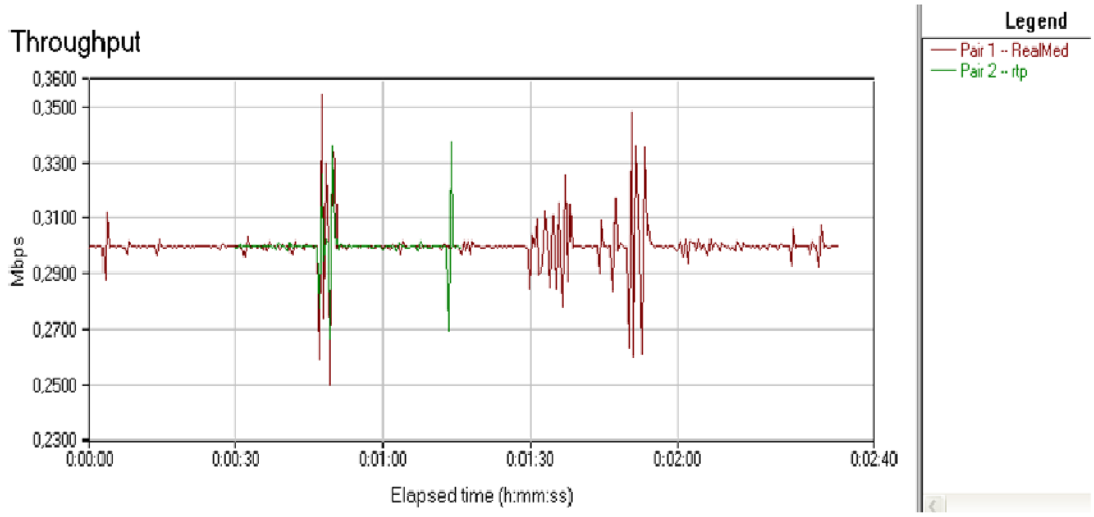

Figure A.42 : 1-6-11-1 channel selection -N4- rtp

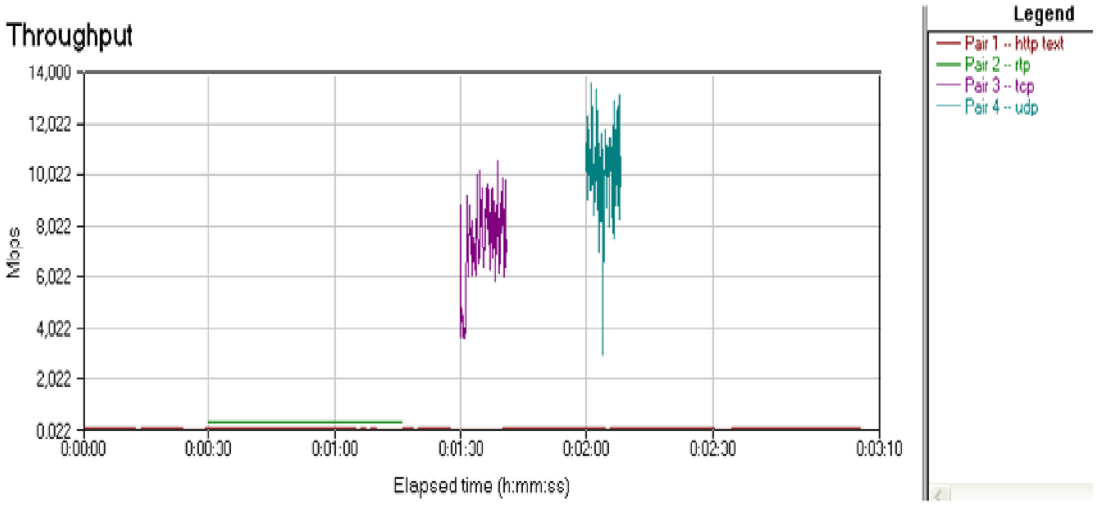

Figure A.43 : 1-7-1-11 channel selection -N1- all

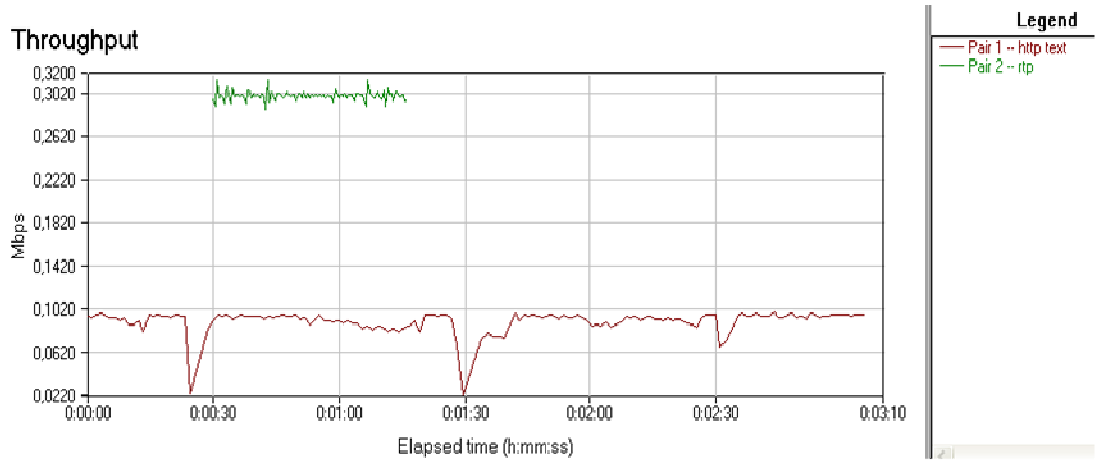

Figure A.44 : 1-7-1-11 channel selection -N1- rtp

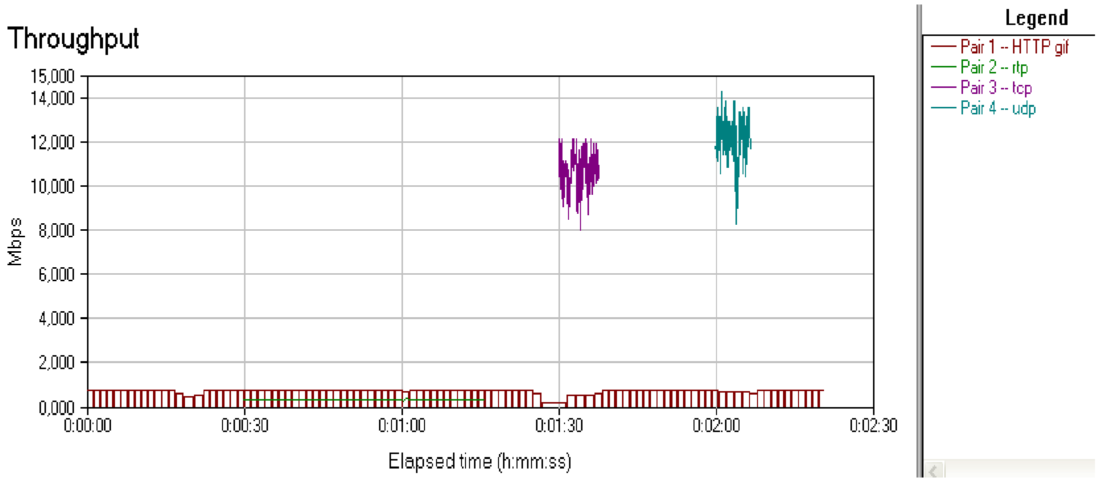

Figure A.45 : 1-7-1-11 channel selection -N2- all

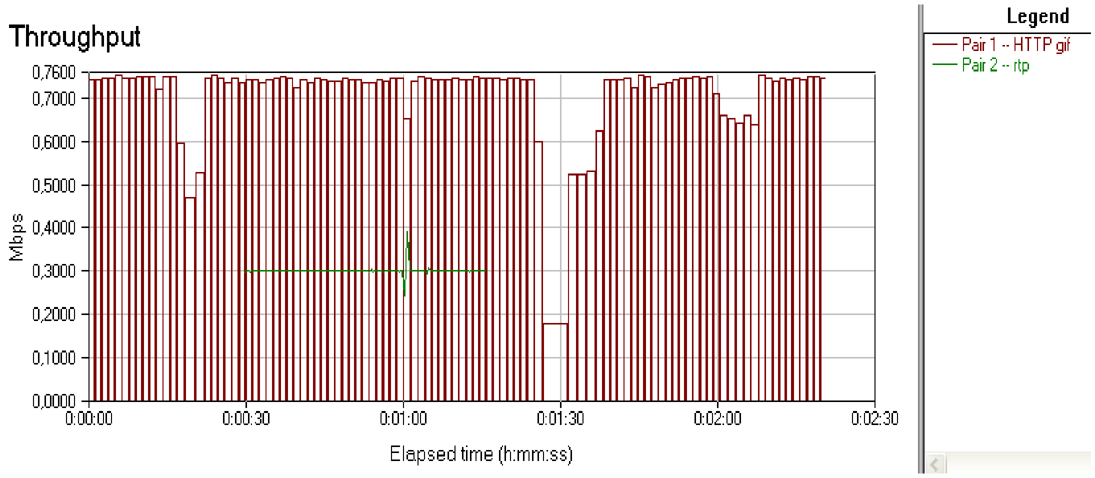

Figure A.46 : 1-7-1-11 channel selection -N2- rtp

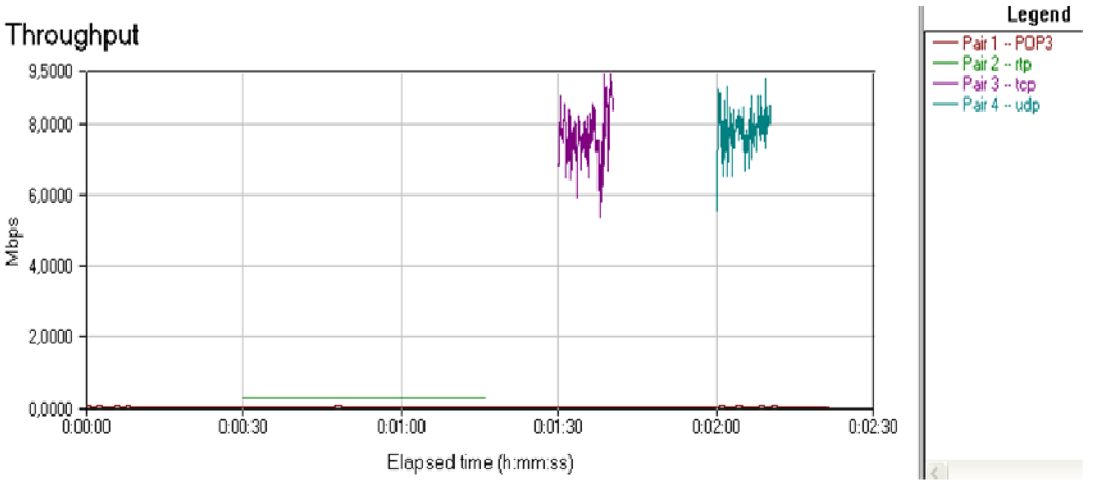

Figure A.47 : 1-7-1-11 channel selection -N3- all

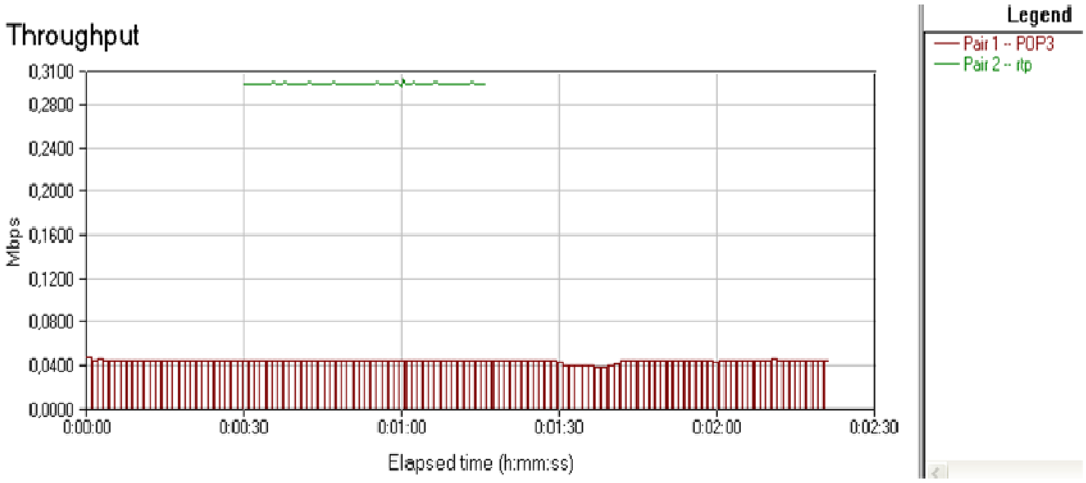

Figure A.48 : 1-7-1-11 channel selection -N3- rtp

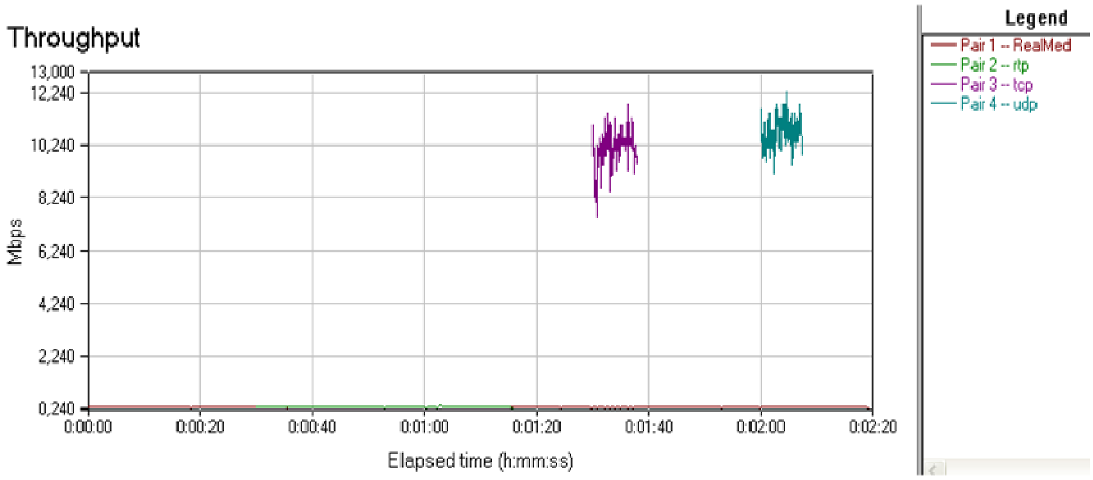

Figure A.49 : 1-7-1-11 channel selection -N4- all

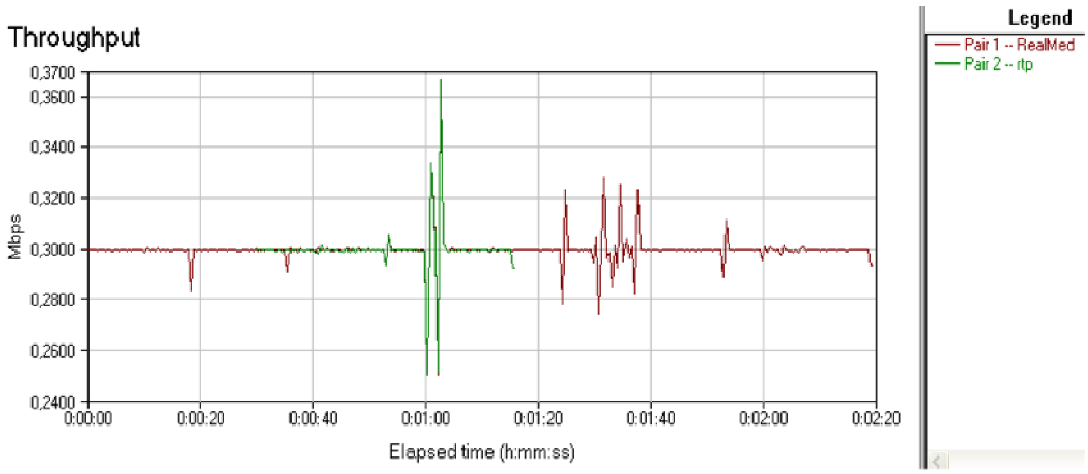

Figure A.50 : 1-7-1-11 channel selection -N4- rtp

CURRICULUM VITAE

Candidate's full name : Gökhan YILDIRIM

Place and date of birth : İstanbul, 1982

Permanent Address : İstanbul Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Maslak – İSTANBUL

Universities and Colleges attended :

B.Sc. : İstanbul Technical University – Computer Engineering (2000 – 2006)