BOĞAZİÇİ UNIVERSITY

CE505
APPLIED STOCHSTIC ANALYSIS&MODELLING

TERM PROJECT

BOTTLENECK PROBLEM IN THE BOĞAZİÇİ UNIVERSITY ETHERNET NETWORK

PRESENTED TO PROF. EMRE OTAY

PROJECT TEAM: OLEG MOROZOV
DZAMBOLAT TOULATOV

20.06.2002

Abstract

As modern business applications become more network aware and mature, they are placing increasing burdens on the underlying network architecture. Applications, such as networked digital video, exact high bandwidth and low latency requirements in order to function properly.

One of the most frequent and severe problems in networks are bottlenecks. Although network breakthroughs repeatedly remove bottlenecks and provide new opportunities, a pattern appears as we attempt to scale up in capability and capacity without limit: every old bottleneck broken reveals another.

Understanding the bottlenecks, the corresponding solutions and potential upper bounds is an important factor in increasing overall network efficiency.

In this project data obtained form the of Boğaziçi University Ethernet network is analyzed using different probabilistic and statistical methods, using stochastic analysis identification of bottlenecks performed and different solutions proposed.

Table of Contents

LIST OF TABLES & FIGURES

1 INTRODUCTION

1.1 INTRODUCTION TO NETWORKS

Over a hundred years ago the world was transformed at the hands of the new industrial age. Steel and iron came to replace the agrarian cotton and wool. Occupations changed, as did the pace of life. Manufacturing processes, formerly measured in weeks, now are accomplished in mere days. Capital was pivotal to ensuring the security of businesses. Copious amounts of capital allowed the purchase of the biggest and fastest machines for production, in essence, the rights for means of production. Nevertheless, the steel and iron of the industrial age has been slowly conquered and toppled by a new amorphous entity: The Information Age.

Information is the new Holy Grail. The strategies and weapons of business are now defined in terms of information. Information is power; it will be the capital of the new millennium. How fast and how much information businesses can review, analyze, and assimilate plays the critical role in guaranteeing marketplace success.

Money talks. And it talks more clearly about priorities than anything else. If you look at the statistics, information has become the most important way for companies to invest in their future. Information technologies, computers, telephony, video, etc. have become the preferred business tooling investment for corporations.

In analyzing the ascension of the information age, one must also dwell on its intertwined nature with the computing sciences. After all, the remarkable evolutionary track of computers engendered the information age. The creation of the transistor fifty years ago fueled a computing revolution that no one could have imagined. The machines we control today are orders of magnitude faster than yesteryear's, and if we are to believe Moore's Law of semiconductor capacity and speed, the trend has no foreseeable end in sight. However, Moore's Law has thus far only been applicable to semiconductor densities and speeds. Left behind and in dire need of improvement are the interconnect technology (i.e. I/O subsystem) and the management software (i.e. the operating system). One of the areas where the semiconductor and interconnect discrepancies have come to bear is in the networking performance, especially of PC-based network servers.Computer networking technologies are the glue that binds these elements together. The public Internet allows businesses around the world to share information with each other and their customers. The global computer network known as the World Wide Web provides services that let consumers buy books, clothes, and even cars online, or auction those same items off when no longer wanted.

Networking allows one computer to send information to and receive information from another. We may not always be aware of the numerous times we access information on computer networks. Certainly the Internet is the most conspicuous example of computer networking, linking millions of computers around the world, but smaller networks play a roll in information access on a daily basis. Many public libraries have replaced their card catalogs with computer terminals that allow patrons to search for books far more quickly and easily. Airports have numerous screens displaying information regarding arriving and departing flights. Many retail stores feature specialized computers that handle point-of-sale transactions. In each of these cases, networking allows many different devices in multiple locations to access a shared repository of data.

We can classify network technologies as belonging to one of two basic groups. **Local area network** (LAN) technologies connect many devices that are relatively close to each other, usually in the same building. The library terminals that display book information would connect over a local area network. **Wide area network** (WAN) technologies connect a smaller number of devices that can be many kilometers apart. For example, if two libraries at the opposite ends of a city wanted to share their book catalog information, they would most likely make use of a wide area network technology, which could be a dedicated line leased from the local telephone company, intended solely to carry their data.

In comparison to WANs, LANs are faster and more reliable, but improvements in technology continue to blur the line of demarcation. Fiber optic cables have allowed LAN technologies to connect devices tens of kilometers apart, while at the same time greatly improving the speed and reliability of WANs.

At the below Fig.1,2 we can see Turkey ISP network and the place of Boğaziçi University in this network.

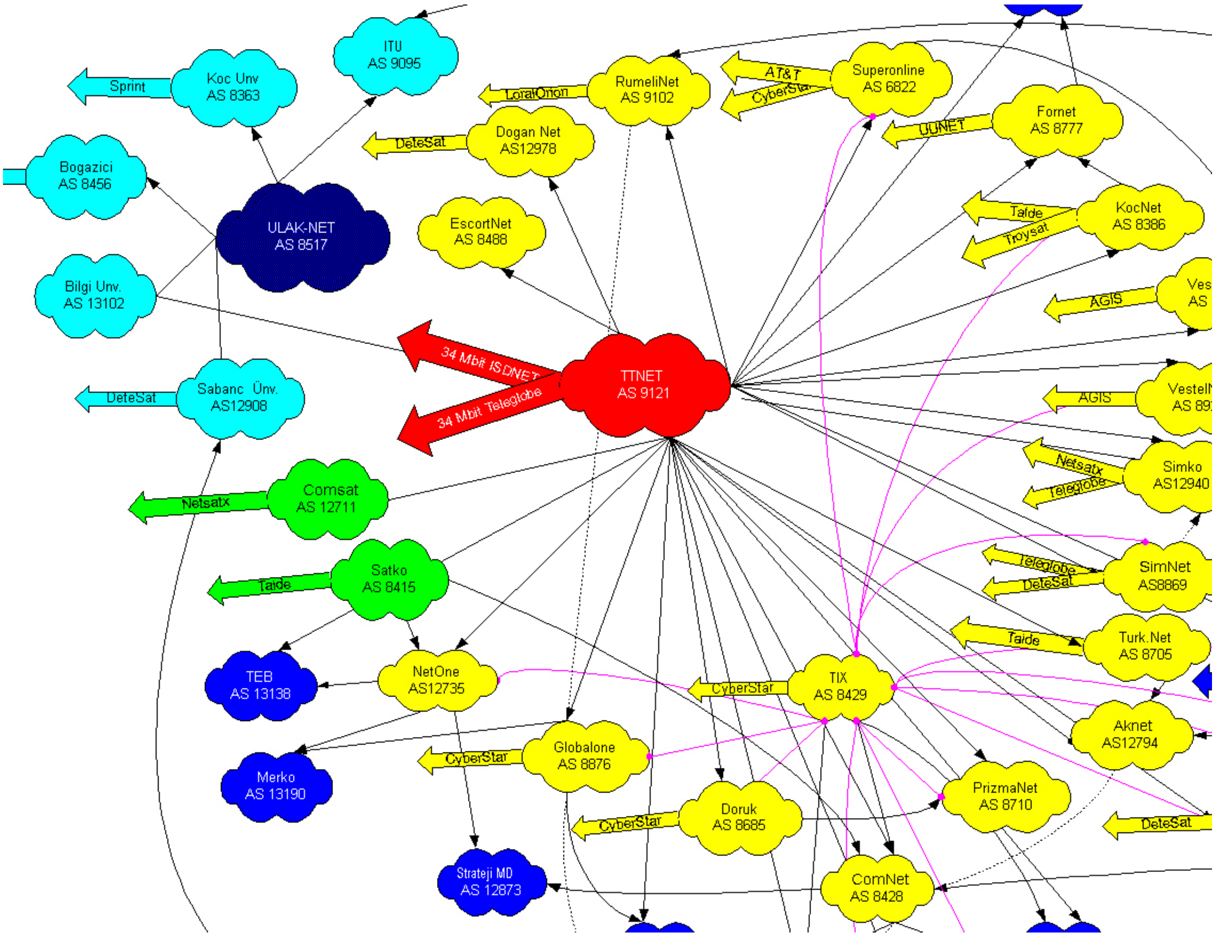

FIG 1. Turkey Internet ISP(Internet Service Provider) Network

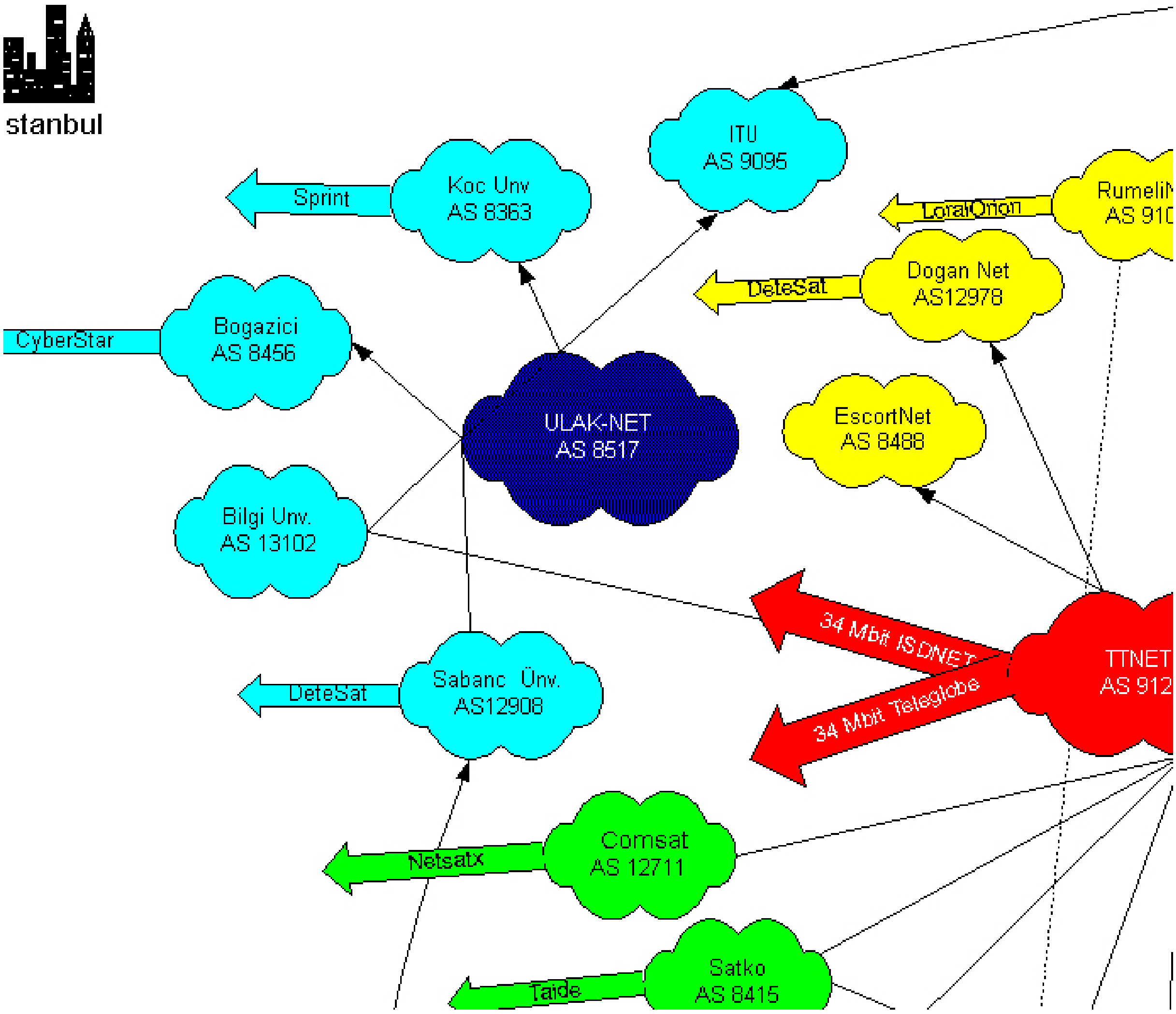

Fig.2 Zoomed part of Fig.1 containing Boğaziçi University

1.2 ETHERNET

In 1973, at Xerox Corporation's Palo Alto Research Center (more commonly known as PARC), researcher **Bob Metcalfe** designed and tested the first Ethernet network. While working on a way to link Xerox's "Alto" computer to a printer , Metcalfe developed the physical method of cabling that connected devices on the Ethernet as well as the standards that governed communication on the cable. Ethernet has since become the most popular and most widely deployed network technology in the world. Many of the issues involved with Ethernet are common to many network technologies, and understanding how Ethernet addressed these issues can provide a foundation that will improve your understanding of networking in general.

The Ethernet standard has grown to encompass new technologies as computer networking has matured, but the mechanics of operation for every Ethernet network today stem from Metcalfe's original design. The original Ethernet described communication over a **single cable** shared by all devices on the network. Once a device attached to this cable, it had the ability to communicate with any other attached device. This allows the network to expand to accommodate new devices without requiring any modification to those devices already on the network.

Ethernet is a local area technology, with networks traditionally operating within a single building, connecting devices in **close proximity**. At most, Ethernet devices could have only a few hundred meters of cable between them, making it impractical to connect geographically dispersed locations. Modern advancements have increased these distances considerably, allowing Ethernet networks to span tens of kilometers.

Ethernet follows a simple set of rules that govern its basic operation. To better understand these rules, it is important to understand the basics of Ethernet terminology.

- **Medium** - Ethernet devices attach to a common **medium** that provides a path along which the electronic signals will travel. Historically, this medium has been coaxial copper cable, but today it is more commonly a twisted pair or fiber optic cabling.
- **Node** - Devices that attach to that segment are stations or **nodes**.
- **Frame** - The nodes communicate in short messages called **frames**, which are variably sized chunks of information.

Frames are analogous to sentences in human language. In English, we have rules for constructing our sentences: We know that each sentence must contain a subject and a predicate. The **Ethernet protocol** specifies a set of rules for constructing frames. There are explicit minimum and maximum lengths for frames, and a set of required pieces of information that must appear in the frame. Each frame must include, for example, both a **destination address** and a **source address**, which identify the recipient and the sender of the message. The address uniquely identifies the node, just as a name identifies a particular person. No two Ethernet devices should ever have the same address.

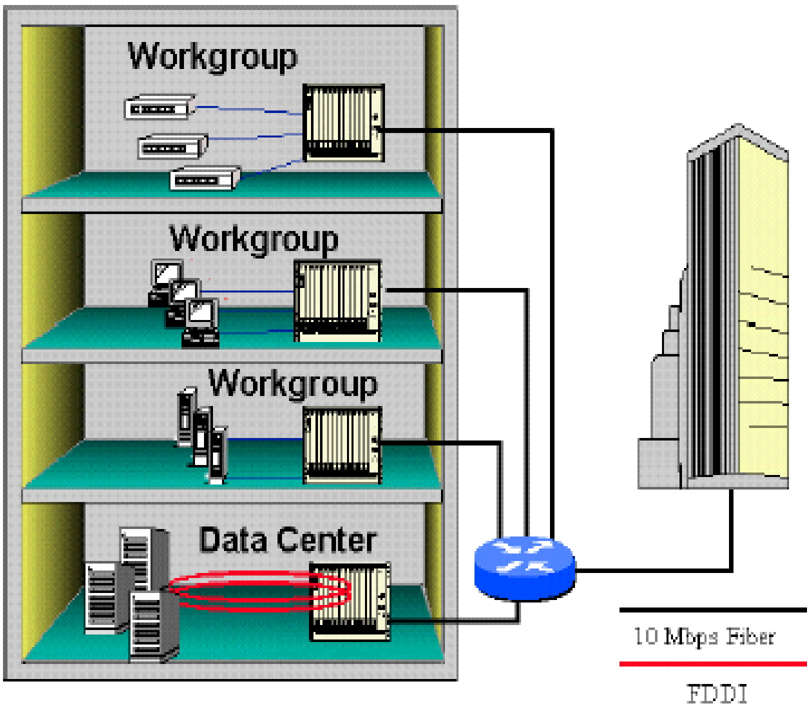

Fig.3 Sample Ethernet Network

1.3 IDENTIFICATION OF PROBLEM

1.3.1 TECHNICAL APPROACH

SYMPTOMS AND EFFECTS OF BOTTLENECKS

Until fairly recently, bottleneck had never been a problem. LANs were historically designed around data rates in the millions of bits per second, far in excess of most computing devices' communications capabilities. However, advances in computing and communications controller technology have changed this scenario. Many devices today can (if given the opportunity) use the full channel capacity of a typical LAN. When many such devices share the channel, it is likely to cause bottlenecks. Bottleneck is a statistical phenomenon; it occurs as a function of traffic patterns. Bottleneck manifests itself in a number of ways.

Increased Network Delay

All LANs have a finite data carrying capacity. When presented with a short-term overload, the LAN distributes that load over time. Thus, when load is light, the average time from submission of a frame for transmission by the host until it is actually sent on the LAN will be short. When there is heavy instantaneous offered load, the average delay (known as *service time*) will increase. This has the effect of making the network appear "slower", since it takes longer to send the same amount of data under congestion conditions than it does when load is light. It is difficult to directly measure service time (unless specially-instrumented driver software is configured for exactly this purpose). There are other, more observable metrics that can be used to infer bottle neck conditions.

Observable Parameters

Many parameters of LAN operation can be measured to assess network performance. Some parameters are automatically measured by standard controllers and host software; others typically require special network monitoring equipment, such as protocol analyzers or remote monitors (RMONs).

User Dissatisfaction.

The ultimate manifestation of bottlenecks is user dissatisfaction. Regardless of all the collected statistics, if users are happy with the behavior of the system, then there really is no problem (at least, not now). If users are dissatisfied with the performance of the system, then all of the statistics indicating that the network is behaving correctly won't relieve their ire. User reaction is the most important metric of network performance. Again, user dissatisfaction with the performance of the network does not immediately indicate a congestion problem. The users' concept of "the network" includes the applications, servers, protocol stacks, internetworking devices, etc., and not just the underlying communications network.

At the below Fig.4 we can see Boğaziçi University Ethernet Network.

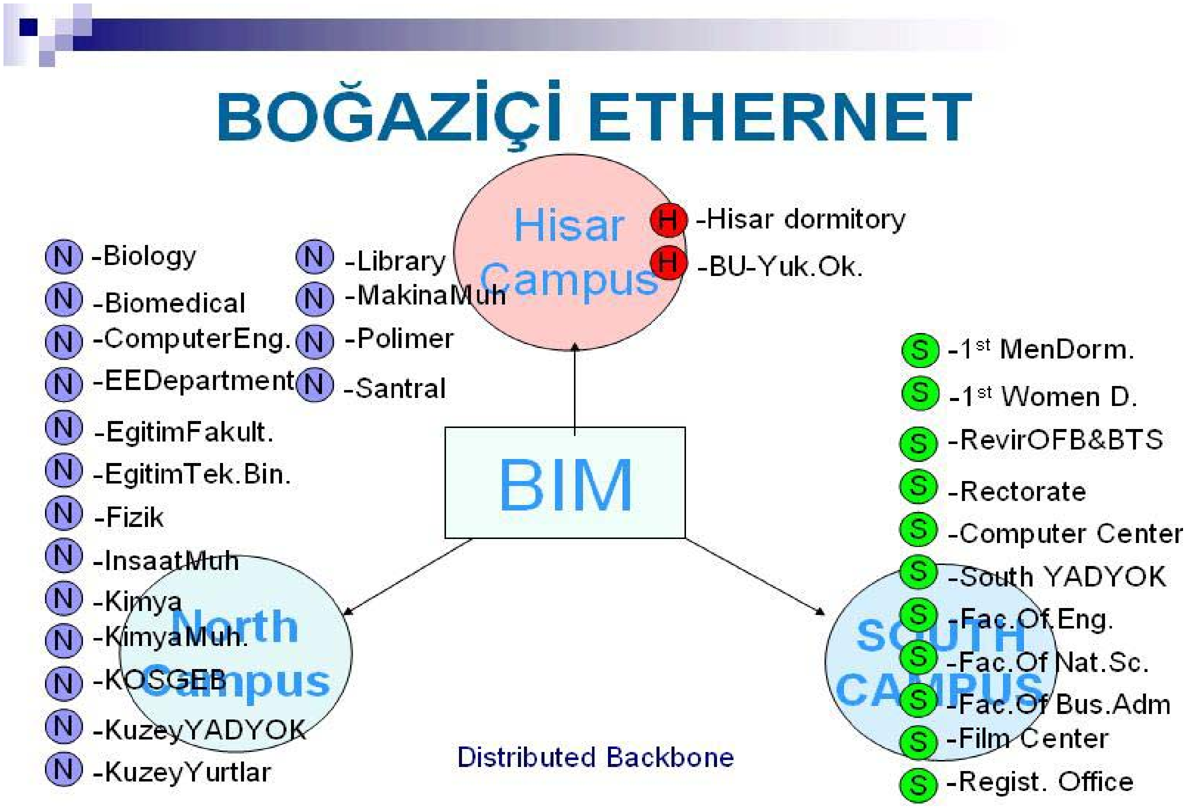

Fig.4 Boğaziçi University Ethernet

1.3.2 MATHEMATICAL APPROACH

Some assumptions are made to make analysis appropriate for our case. Demand may and may not exceed allocation from Server. Sum of Allocations may not exceed Capacitance of Server Bottleneck happens when Demand/Allocation reaches 1. All Demands and Allocations referring to different campuses are identified by variable index i. Approach to the solution to optimization problem is given in **METHODOLOGY** part of this project.

2 DATA ACQUISITION

2.1 MRTG

The Data for the analysis was acquired fusing the MRTG program. The Multi Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network-links. MRTG generates HTML pages containing graphical images which provide a LIVE visual representation of this traffic. MRTG takes the measurements of the internet traffic in the Ethernet links for every 5 minutes.

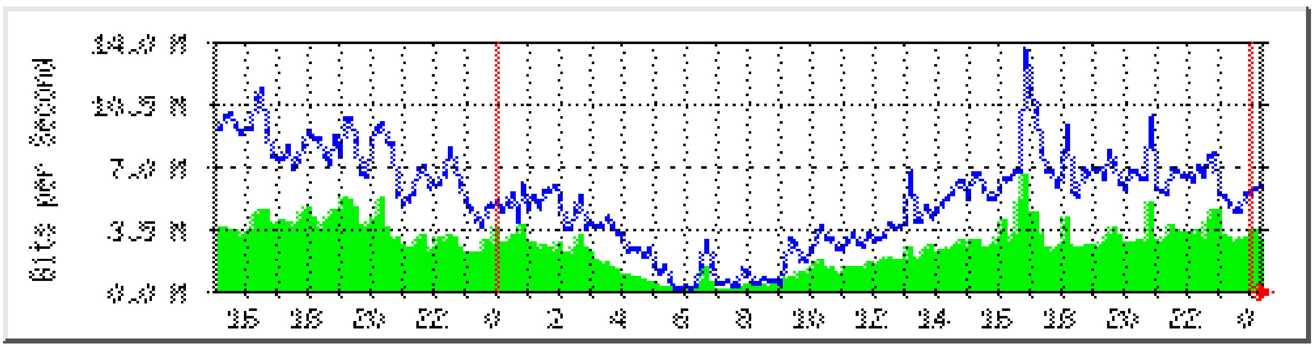

Fig.5 Sample MRTG output

2.2 UNIVERSITY CAFETERIA DATA

Data from Boğaziçi University “Yemekhane” were obtained

Ş SCHOOL CAFETERIA MAY 2002 DAYLY DATA

Date		Coun. 1	Coun.1.Burs	C.1.Pers.	Coun. 2	C.2.Pers.	Counter 3	Sequrity	Staff	Sum	Total
01.05.2002	Lunch	83	36	340	0	0	0	29	228	716	
	Dinner	0	0	0	37	65		27		129	845
02.05.2002	Lunch	0	39	0	0	0	61	31	142	273	
	Dinner	55	46	58	0	0		27		186	459
03.05.2002	Lunch	0	0	0	109	284	0	30	146	569	
	Dinner	0	0	0	79	64		27		170	739
06.05.2002	Lunch	42	248	0	192	344	424	34	225	1509	
	Dinner	106	228	18	158	46		27		583	2092
07.05.2002	Lunch	0	263	0	255	282	349	34	192	1375	
	Dinner	114	329	30	310	77		27		887	2262
08.05.2002	Lunch	0	319	0	265	343	465	30	221	1643	
	Dinner	145	311	26	226	51		27		786	2429
09.05.2002	Lunch	2	307	0	193	256	297	31	167	1253	
	Dinner	159	325	25	301	49	0	27		886	2139
10.05.2002	Lunch	2	379	278	0	0	557	33	177	1426	
	Dinner	339	250	82	0	0		27		698	2124
	Lunch	0	0	0	0	0	0	0	0	0	
11.05.2002	Dinner	163	188	31	0	0	0	0		382	382
13.05.2002	Lunch	0	360	0	227	306	363	34	180	1470	
	Dinner	71	310	17	419	74		27		918	2388
14.05.2002	Lunch	334	0	0	652	325	0	32	197	1540	
	Dinner	42	263	7	314	85	0	27		738	2278
15.05.2002	Lunch	0	346	0	167	233	365	26	161	1298	
	Dinner	50	230	14	255	63		27		639	1937
16.05.2002	Lunch	0	363	0	276	338	521	35	285	1818	
	Dinner	0	0	0	302	71	0	27		400	2218
17.05.2002	Lunch	0	0	0	109	302	530	37	208	1186	
	Dinner	0	0	0	219	74	0	27		320	1506
	Lunch	0	0	0	0	0	0	0	0	0	
18.05.2002	Dinner	0	0	0	121	23		0		144	144
20.05.2002	Lunch	1	370	0	206	269	393	31	205	1475	
	Dinner	115	337	32	255	52	0	27		818	2293
	Lunch									0	
20.04.2002	Dinner	116	162	10	36	0				324	324
24.04.2002	Lunch	2	268	0	237	296	374	44	187	1408	
	Dinner	45	257	11	329	63	0	27		732	2140
25.04.2002	Lunch	0	343	0	183	281	371	32	220	1430	
	Dinner	257	226	68	0	0	0	27		578	2008
26.04.2002	Lunch	18	246	273	0	0	444	33	186	1200	
	Dinner	240	170	68	0	0	0	27		505	1705
	Lunch	0	0	0	0	0	0	0	0	0	0
27.04.2002	Dinner	25	36	10	0	0		0		71	71
29.04.2002	Lunch	0	27	0	59	306	0	30	180	602	
	Dinner	0	0	0	36	52	0	27		115	717
	Lunch									0	
27.10.2001	Dinner				109	4				113	113
30.04.2002	Lunch	0	0	0	95	311	0	30	173	609	
	Dinner	33	52	51	0	0		27		163	772
TOTAL		2559	7634	1449	6731	5389	5514	1129	3680	34085	#REF!

Table 1.Boğaziçi Cafeteria Data

3 DATA PROCESSING AND INTERPRITATION

3.1 EXTRACTION AND CLEARING OF DATA

Data was firstly extracted from the MRTG log file. It is initially stored in the reverse order in a text format. For the proper processing we reversed the data using C++ written program (see code 1 Appendix B).For the some stochastic analysis raw data is used , but for some we had to covert the data using different time averages(hourly,6 hour,12 hour, daily). Initial data was taken with five minute intervals which make 12 intervals for 1 hour.

3.2 CAFETERIA DATA ADJUSTMENTS

The cafeteria data was adjusted according to the daily factors. Data for the weekend was eliminated due to the lack of corresponding data. Data sets that were taken into correlation with the cafeteria data were converted into 1 hour base to obtain meaningful results, because we have user numbers in the cafeteria in two specific time intervals(12:30-2:30) and (17:30-19:30).

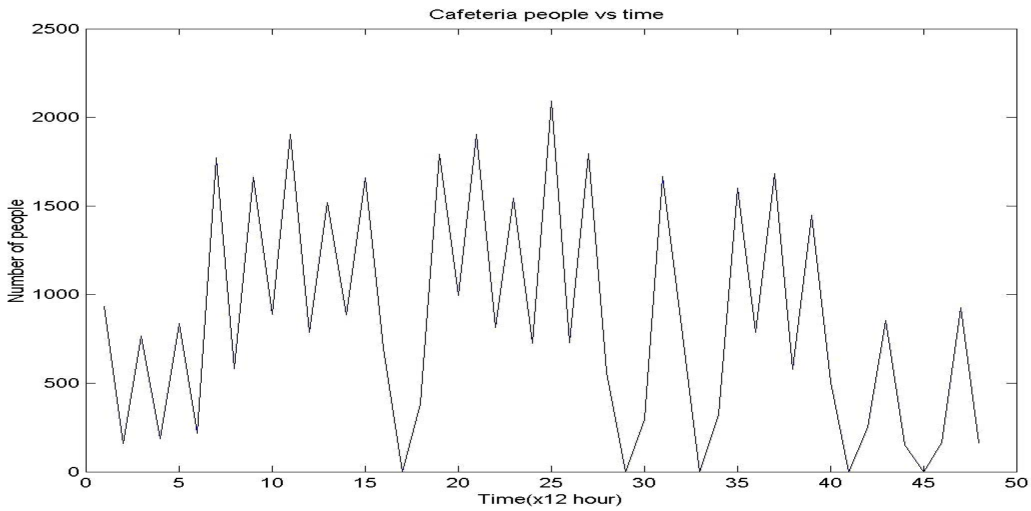

Fig. 6 Cafeteria vs. Time

At the above figure we can easily see the fluctuations corresponding to the lunch and dinner differences.

4 STOCHASTIC ANALYSIS

4.1 ANALYSIS IN TIME DOMAIN

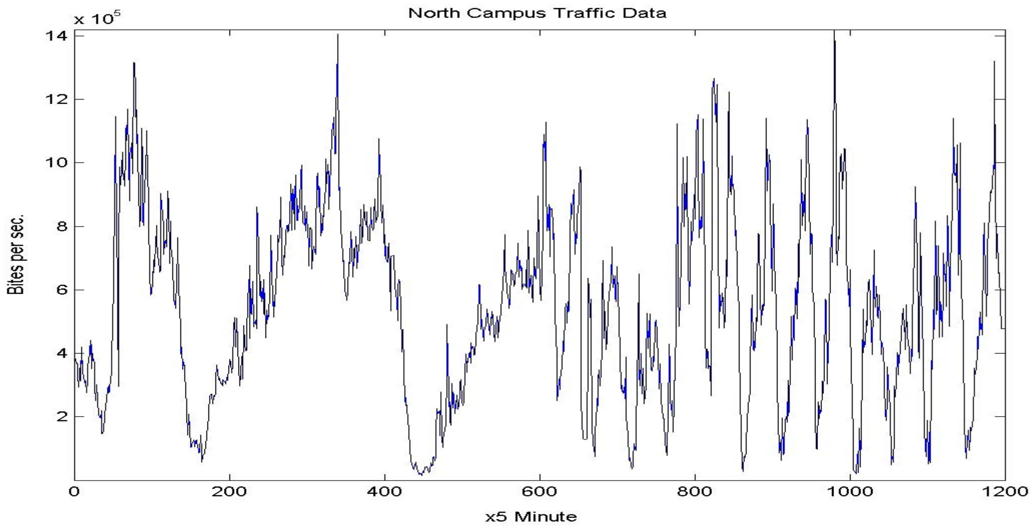

Fig.7 North Campus Internet Traffic Data

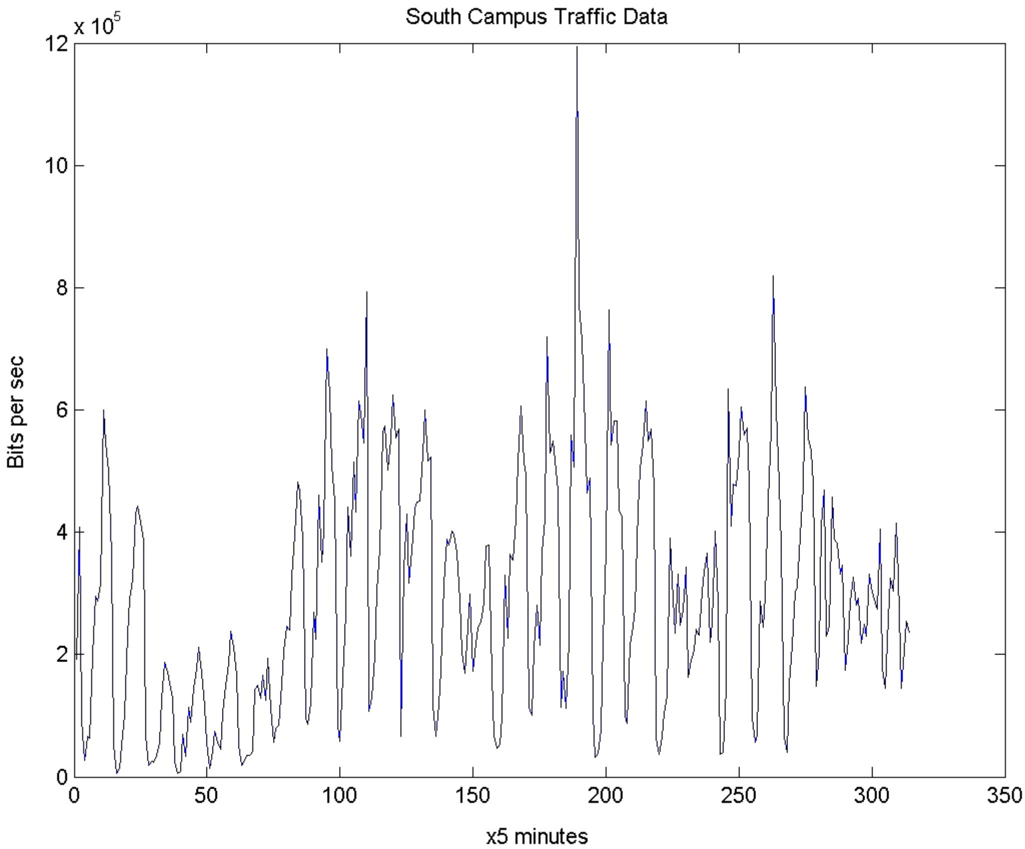

Fig.8 South Campus Internet Traffic Data

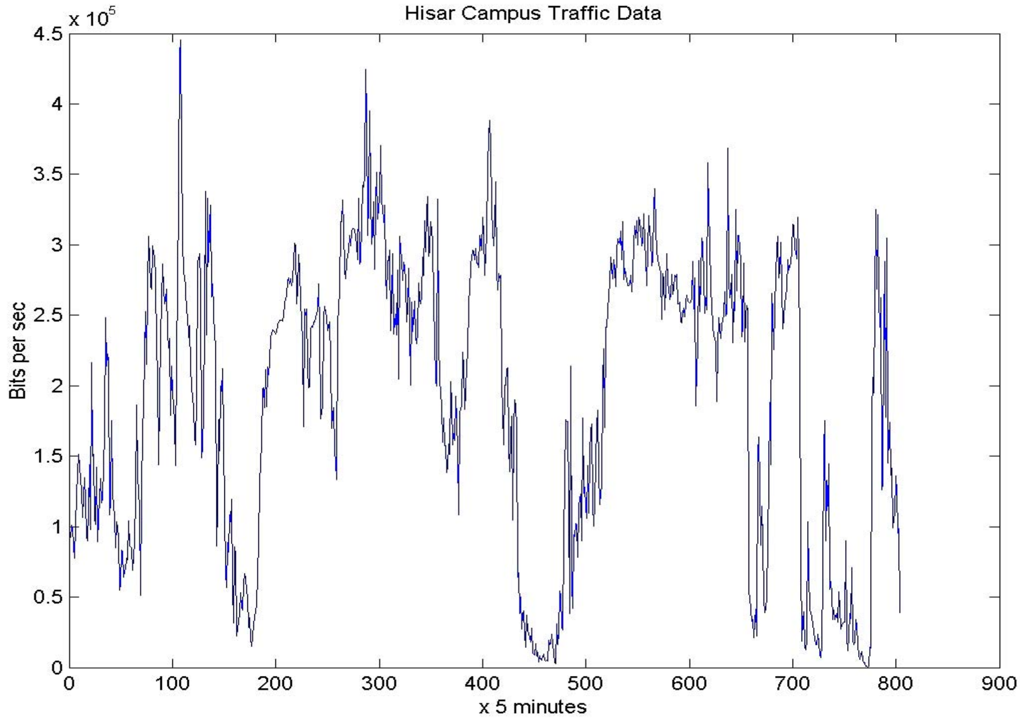

Fig.9 Hisar Campus Internet Traffic Data

Cross-correlations between this campuses data did not give any meaningful results. So, fluctuations in traffic magnitudes are seem to be influenced in each campus due to different reasons and times.

4.2 ANALYSIS IN FREQUENCY DOMAIN

Cross-Correlation between Campuses Internet Traffic Data and Cafeteria didn't gave any reasonable results. It can be concluded that number of people having meal in cafeteria doesn't have direct relationship between the current number of internet users. This may be related to the fact that the duration of dinner is almost 2,5 hour and the effect is smashed. The number of students in the evening is by half smaller, so the fluctuations of the students available for internet usage are not important. Also because of the fact that students enter lessons that are 1 hour long the fact of students having meal is not important. The below figure is the correlation graph between the Hisar campus data and cafeteria. As can be seen no meaningful results can be extracted.

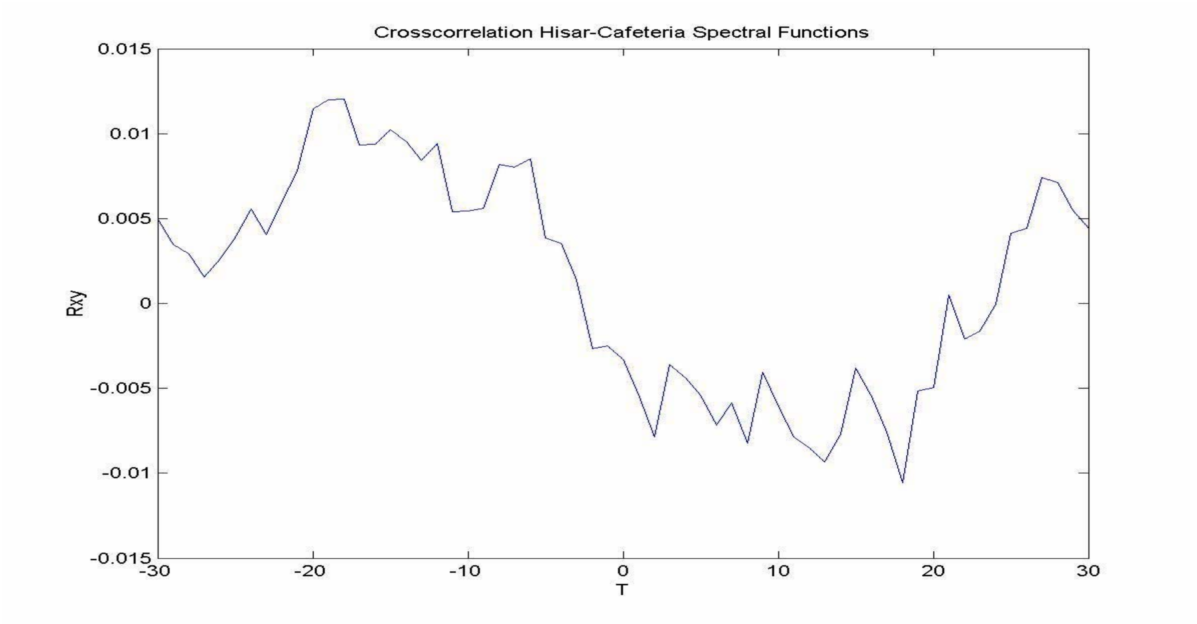

Fig. 10 Cross-Correlation Hisar-Cafeteria

Campuses Spectral Density Functions

Hisar Campus Spectral Density Function

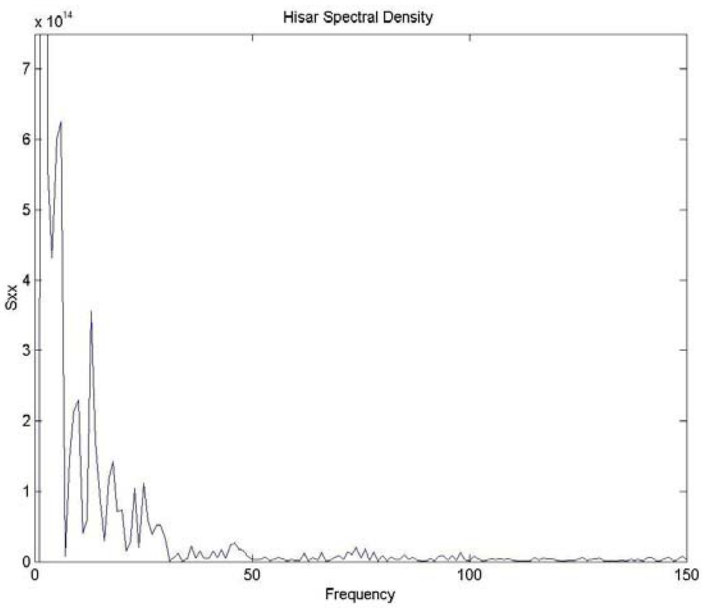

Fig. 11 Hisar Camp. Spectral Density Function

South Campus Spectral Density Function

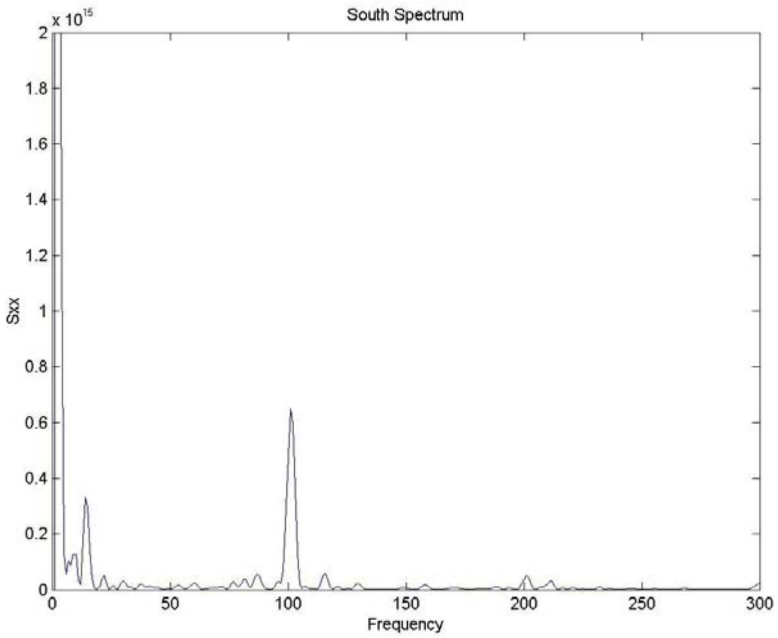

Fig. 12 South Camp. Spectral Density Function

Servers Spectral Density Function

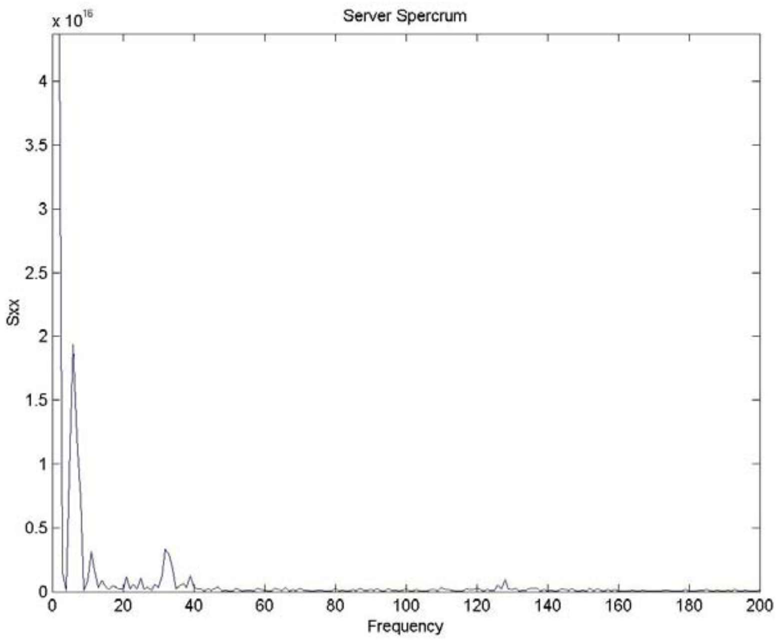

Fig. 13 Main server Spectral Density Function

Cross-Correlation Between Campuses

Hisar-South Campus Cross-Correlation

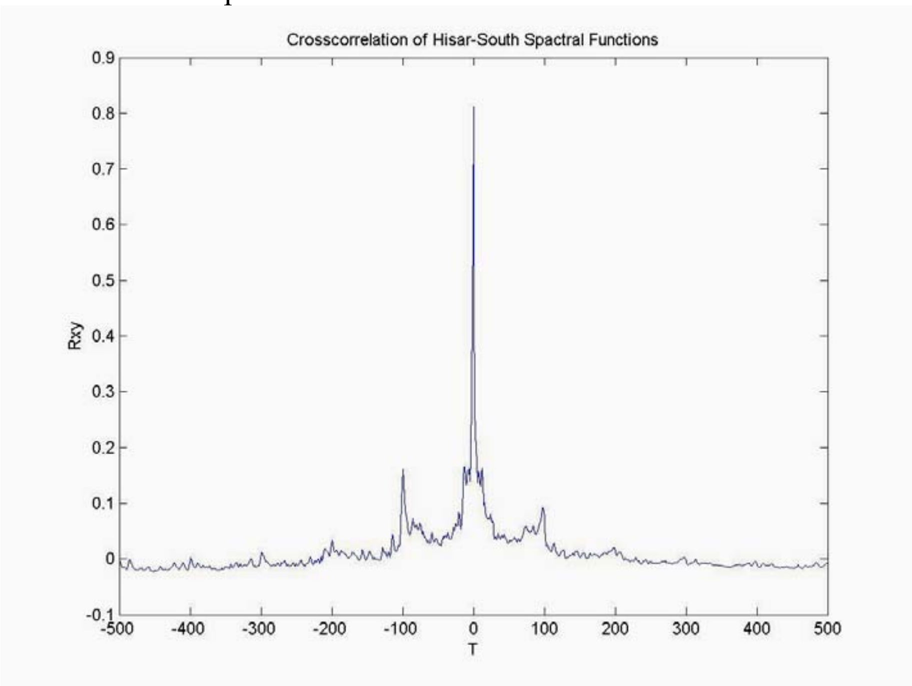

Fig. 14 Hisar-South Cross Correlation

4.3 INTERPRITATION OF RESULTS

As one can see from the cross-correlation function in the Fig.14 that this function has properties of autocorrelation function, such as Rxy(0)≥Rxy(x) and Rxy(-x)≈Rxy(x). The explanation for that is that Hisar and South campuses has almost the same "tempo" or rhythm or pace of life. So Hisar and South campuses belong to one university, have the same time table of lectures and breaks. In the Fig.14 we can distinguish picks at 100 values. They are originated due to South Campus' secondary peaks in spectral graph. Since this campus is the largest one in terms of number internet users , more complex behavior of spectral function is justified.

5 OPTIMIZATION

5.1 METHODOLOGY

Optimization could be performed using several techniques and programs. Since our objective has several constraints, we decided to try several functions with both maximizing and minimizing objectives. But fundament of our analysis is based on following approach:

D(i,j)/A(i,j), where D(i,j) and
A(i,j) corresponds to demand and allocation
Respectively at hour i and campus j

As D(i,j)/A(i,j) approaches 1, possibility of bottleneck increases and actually happens when the ratio exactly equals 1. In following parts more precise description and optimization function described.

In our analysis we tried several approaches on optimization objective identification. Several linear and nonlinear solver programs were considered and some optimization objective functions were proposed. Some of them were to maximize objective functions by changing allocation parameters keeping demand data fixed, while minimizing objective functions with several constraints on demand as well as allocation values were checked. Then linear solver tool in Microsoft Excel was used for a trial. Reasonable results were obtained, but technical aspect was quite cumbersome due to lack of somewhat automatically useful procedure. Since we were supposed to work with a lot of data in our analysis more suitable program was to be introduced and used. So GAMS solver with C++ was implemented.

5.2 SOFTWARE DEVELOPMENT

5.2.1 GAMS SOLVER IDENTIFICATION PARAMETERS

GAMS solver is an optimization program that uses linear and nonlinear solution algorithms for solution of various optimization as mono as well as multi objective functions. Input parameters are defined in text format in ***.gms** file. Output results are obtained after running GAMS solver in ***.LST** file. C++ was used to make running program automatically for every new input parameters and obtained results transform to suitable **xls** format. Source Code can be found in Appendix B.

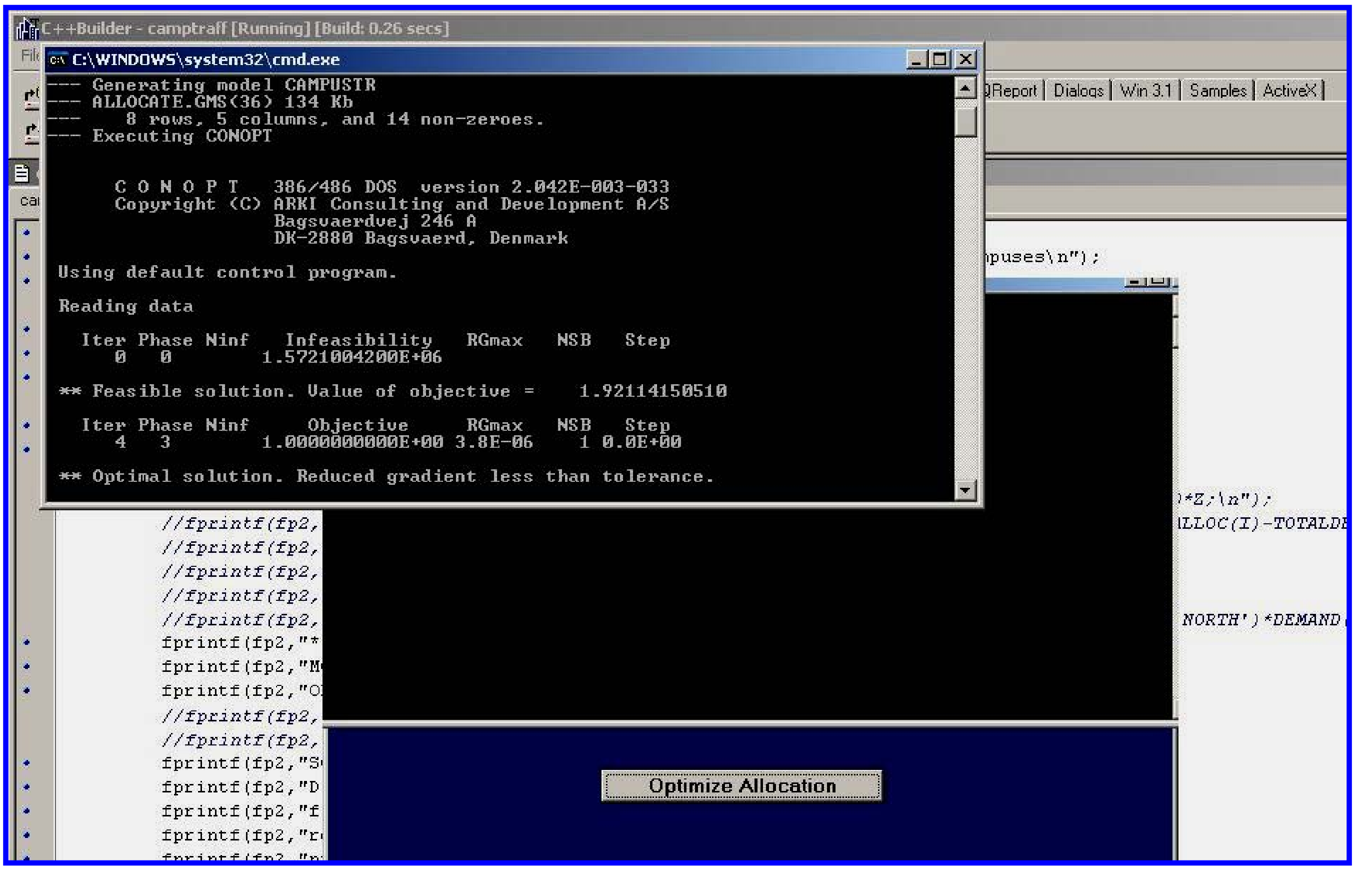

Fig.15. GAMS running.

5.2.2 DETERMINATION OF OPTIMIZATION FUNCTION

As mentioned in METHODOLOGY part, minimization of objective led to more reasonable results, while maximization approach brought about "resonance" results that are meaningless.

Examples of MAXIMIZING Objective Functions:
1) OBJ = SUM(I,(DEMAND(I)/ALLOC(I)-1)*(DEMAND(I)/ALLOC(I)-1))-10*Z, where Z=Limit of DEMAND(I)/ALLOC(I);
2) OBJ = SUM(I,(DEMAND(I)/ALLOC(I)-TOTDEM/MAXCAP)*(DEMAND(I)/ALLOC(I)-TOTDEM/ MAXCAP));
3) OBJ = SUM(I,(ALLOC(I)-DEMAND(I))/DEMAND(I));
Examples of MINIMIZING Objective Functions:
4) OBJ =E= 5*Z+SUM(I,SIGNIF(I)*DEMAND(I)/ALLOC(I)), where Z=Limit of DEMAND(I)/ALLOC(I);
5) OBJ = SUM(I,DEMAND(I)/ALLOC(I));
Finally following Minimization Objective Function was accepted

F(i,j)=$\sum$(Sign.Fact(i,j).)*(D(i,j)/A(i,j), where D(i,j) and A(i,j) corresponds to demand and allocation respectively at hour i and campus j. Sign.Fact(i,j) corresponds to significant(weight) factor for giving priority to definite campuses , labs or dormitory internet users.

Constraint $\sum$A (i,j)=Capacity of Server is introduced for ultimate usage of capacity of server.

5.3 INTERFACE DEVELOPMENT

Our choice of code to be used in interface development stopped on "object orienting programming language" C++, since it's easy compatibility with GAMS solver. C++ Builder Code was used to develop interface, run GAMS program automatically for every new input parameters and transform obtained results into more useful **xls** format , so making recalculation procedure very simple. Source Code can be found in Appendix B.

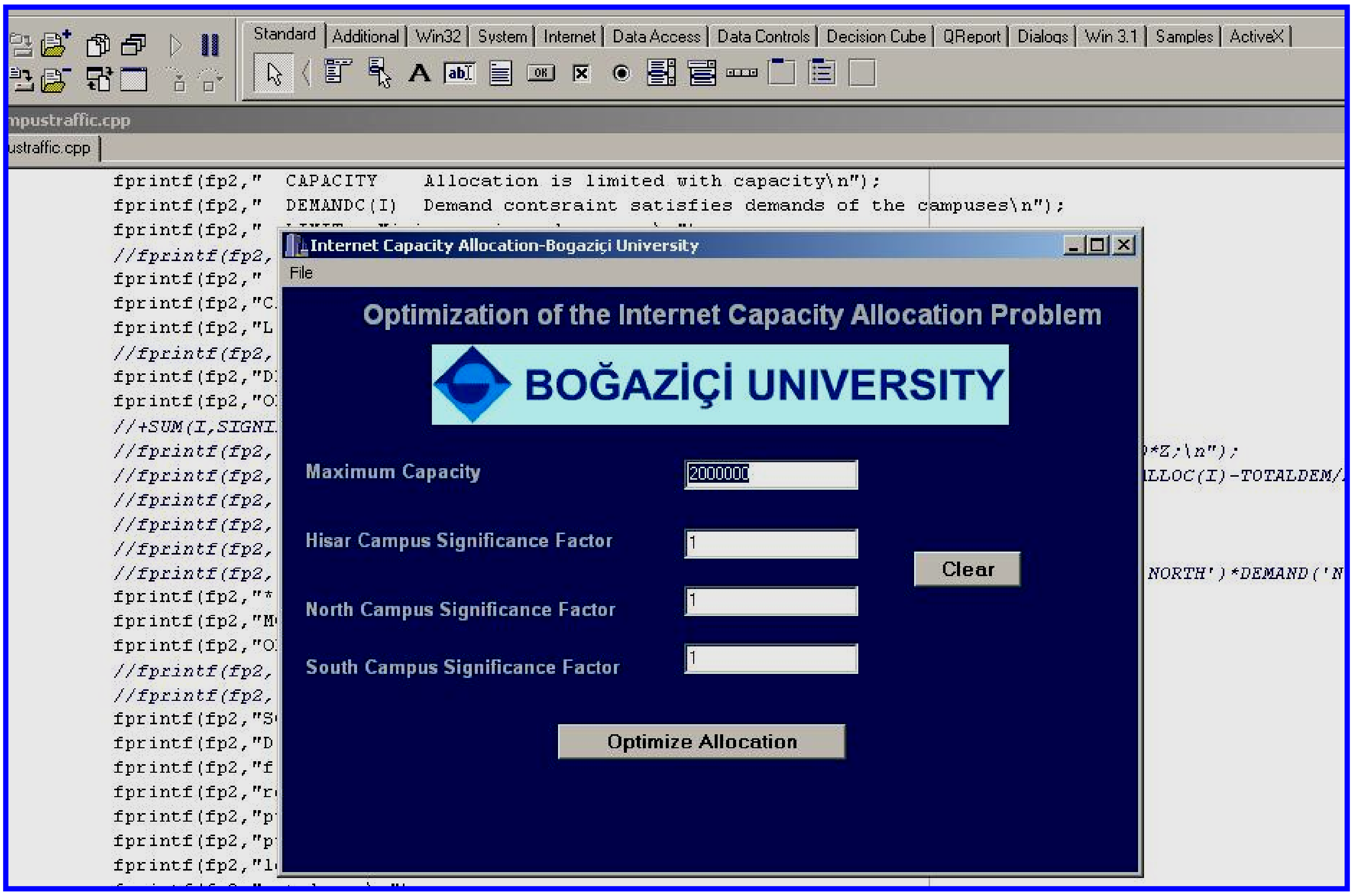

Fig.16 Interface Developed

5.4 CASE STUDY TO DETRMINE EFFICIENCY GAIN

In order to estimate the possible gain after optimization following case is considered. Capacity of university server is 1.500.000 bytes/sec which is lower than the maximum total demand that ever occurred. Assumption is that this capacity is equally allocated between campuses, so that each campus has 500.000 bytes/sec option for usage.
Following Figures shows demand that could not be satisfied for each campus

Fig.17 Demand not satisfied before optimization.

Total area of Demand that is not satisfied for all campuses is equal to 6.275.413 bytes.
After optimization of internet traffic following graph of demand that could not be satisfied is obtained.

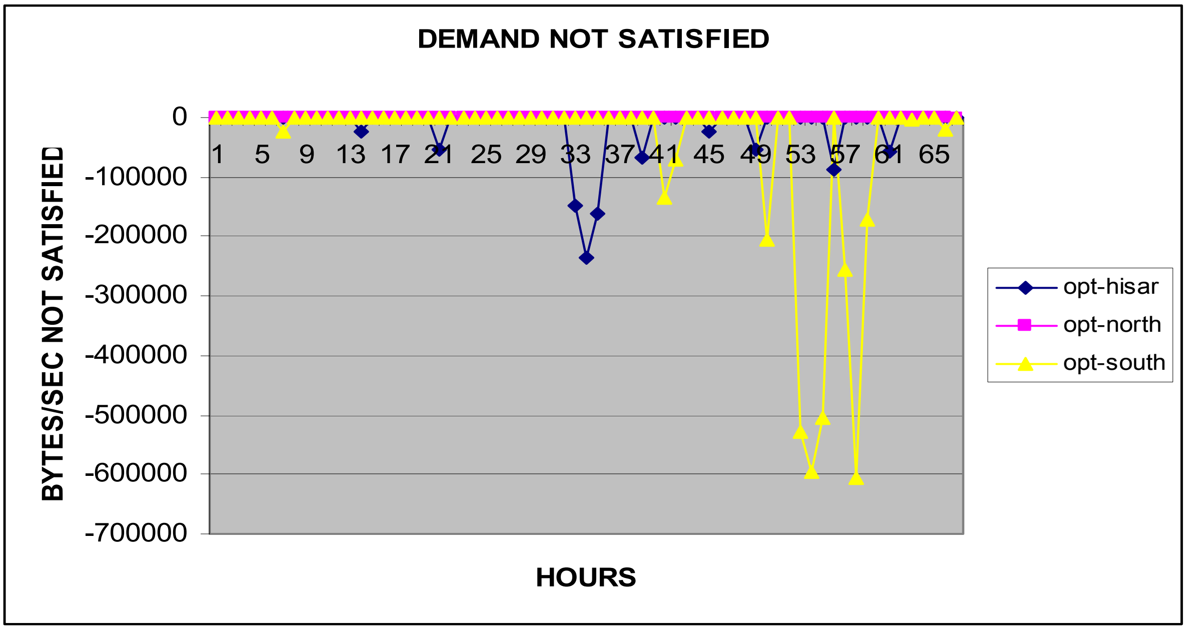

Fig. 18 Demand not satisfied after optimization.

Total area of Demand that is not satisfied after optimization for all campuses is equal to -4.036.739 bytes. So that almost there is 36% reduction in demand that is not satisfied. This is quite a considerable number keeping in mind that capacity of the server kept constant.

6 INTERPRITATION OF RESULTS

Considering our assumption we tried to minimize the overall time when internet traffic exceeds 80% of the server capacity some peaks above this level can be bared by the server without major bottlenecks. This assumption has been made due to the fact that it is impossible measure the actual users demand. Data that have been analyzed is the response of the server to the users demand. Server may have allocated less or more than the amount demanded by the user.

6.1 SUMMARY OF FINDINGS

Internet Traffic Data obtained for North, South and Hisar Campuses of Boğaziçi University were converted to a more appropriate time interval and adjusted to other data (cafeteria's number of students each month). After that analysis in time and frequency domains were performed. Time analysis gave just quantitative parameters such as mean values of internet traffic, fluctuations, overall and campus' maximum and minimum demand magnitudes. Stochastic analysis in frequency domain led to more "interesting" results. After taking cross-correlations between North-South, South-Hisar and North- Hisar campuses we could "catch" almost the same pattern of behavior of fluctuations for campuses that led us to some trivial conclusions, for example each campus used same time table for lecture hours or the very same students studied at these campuses or all campuses belong to on university.
Then optimization using linear and nonlinear approaches for allocation capacity of server for all three campuses and labs plus dormitories within each campus was performed and optimization parameters and procedure were found and established.

6.2 FINAL RESULTS AND PROPOSED SOLUTION

Given the capacity of server we allocated it in the most efficient way so that to eliminate the possibility of bottleneck if capacity of server is more than the maximum demand ever occurred or decrease the severity of bottleneck if capacity of server is lower than the demand values that may occur several times.
Case study was considered so that to estimate the efficiency that maybe reached thanks to the optimization we propose.

TRADITIONAL SOLUTIONS TO BOTTLENECKS

Bottleneck means that there is more offered load than there is capacity of the LAN to carry it. All solutions to Bottleneck problems involve either increasing the LAN's capacity, or reducing the offered load.

Increasing LAN Capacity

If a 10 Mbps LAN is experiencing utilization of 80% averaged over a period of hours (typically an indication of bottleneck), the utilization would be on the order of 5% over the same period if the LAN capacity were increased to 100 Mbps. Thus, we can eliminate bottleneck by increasing the capacity of the LAN (as long as we don't simultaneously increase the offered load).
This solves the problem, but has some undesirable side effects such as :
1) *Increased cost*. Controllers, hubs, and internetworking devices designed for 100 Mbps operation (regardless of the particular technology) cost more than their 10 Mbps equivalents. While the relationship is not linear with data rate (i.e., it isn't 10 times the cost), it is still more expensive to use faster technologies.
2) *Replacement of existing interfaces and equipment*. Not only is the higher speed equipment more expensive, there is usually an existing investment in the lower speed equipment already in place. So the higher cost is not really *instead of*, but *in addition to*, the lower speed equipment costs. Added to that is the expenseof equipment replacement and installation, and disruption of network services during the changeover.Newly-installed networks (particularly high-speed LANs using less-mature products) often go through a period of unstable, problem-prone behavior, until the administrators become familiar with the equipment and its operation.
3) *Replacement of media*. In many cases, a higher-speed LAN will require new cabling to be usable. Many existing cable plants were not designed to support high-speed LAN operation. Cables, connectors, patch panels and other wiring closet components may have to be replaced. For some technologies, copper wire may have to be replaced with fiber to be effective.
SWITCH CONCEPTS
A switch is a link-layer internetworking device that allows simultaneous frame exchange among large numbers of LANs and workstations.Using the switches one can control the ISP(Internet Service Provider) incoming backbone bandwidth in most efficient way. Using the results obtained we switch router can be programmed in order to allocate the necessary traffic between the campuses daily, weekly, monthly. Although the procedure made above can be fully atomized it can although be internalized into the router for continuous analysis according to the incoming data. We propose the switching rate as 1 hour to overcome inconveniences that may be caused to the users.

7 CONCLUSION

Proposed optimization technique, procedure as well as software we used led to 36% reduction in demand that is not satisfied with capacity of the server kept constant. So there is a gain in internet demand satisfaction of internet users in Boğaziçi University and in turn in decrease in troubleshooting while using internet service there.

APPENDIX A: MATLAB SOURCE CODE

1) Function **grafx** to open file with data and to plot it in time domain.

```
function  f=a();
fid=fopen('Backbone.txt','r')
[X,count]=fscanf(fid,'%f')
%X=X-mean(X);
plot(X);
f=X;
```

2) Function **average** to open file with data, calculate hourly averages for adjustments

```
 function  f=a()
fid=fopen('HisarAv.txt','r')
[X,count]=fscanf(fid,'%f')
Y=1:floor(count/12);
a=0;
i=1;
% z=floor(count/2);
for n = 1:count
   a=a+X(n) ;
  if (mod(n,12)==0 ) & (n~=1)
       Y(i)=a/12;
       a=0;
       i=i+1;
   end
end
f=Y;
```

3) Some other built-in functions that are already in library of MATLAB such as:

crosscorr, mean, plot, abs, fft and etc

.

APPENDIX B: GAMS AND C++ SOURCE CODE

1) **Input** parameter **gms** file for **GAMS** solver

```
$TITLE Campus Traffic
$OFFUPPER
SETS  I  /SOUTH,NORTH,HISAR/;
SCALARS
MAXCAP   "Maximum Capacity " /1500000.000000/;
******************
PARAMETERS DEMAND(I), SIGNIF(I),TOTALDEM;
  DEMAND('HISAR')= 114373.000000 ;
  DEMAND('NORTH')= 499831.000000 ;
  DEMAND('SOUTH')= 294893.000000 ;
  SIGNIF('HISAR')= 1.000000 ;
  SIGNIF('NORTH')= 1.000000 ;
  SIGNIF('SOUTH')= 1.000000 ;
  TOTALDEM=SUM(I, DEMAND(I));
******************
 VARIABLES
  ALLOC(I)     Allocation to campus i
  Z        Maximum bottleneck indicator
  OBJ              Objetive Function Value;
POSITIVE VARIABLES ALLOC,Z;
FREE VARIABLE OBJ;
ALLOC.LO(I)=DEMAND(I);
******************
EQUATIONS
CAPACITY    Allocation is limited with capacity
DEMANDC(I)           Demand contsraint satisfies demands of the campuses
LIMIT   Minize maximum d over a
OBJECTIVE   Objective function;
CAPACITY..  SUM( I , ALLOC(I) ) =E= MAXCAP;
LIMIT(I)..       DEMAND(I)/ALLOC(I) =L= Z;
DEMANDC(I)..          ALLOC(I) =G= DEMAND(I);
OBJECTIVE.. OBJ =E= 5*Z+SUM(I,SIGNIF(I)*DEMAND(I)/ALLOC(I));
******************
MODEL CampusTr /ALL/;
OPTION NLP=CONOPT;
SOLVE CampusTr USING NLP MINIMIZING OBJ;
DISPLAY OBJ.L,ALLOC.L;
file results /CampusTraffic.csv/;
results.pc=5;
put results;
put "Objective Function Value ",OBJ.l/;
loop(i,put i.tl, ALLOC.l(i)/);
putclose;
```

2) **Output** parameter **LST** file of **GAMS** solver

```
GAMS 2.50.094  DOS Extended/C                    06/19/02 19:30:42  PAGE      1
Campus Traffic
   3  SETS  I  /SOUTH,NORTH,HISAR/;
   4  SCALARS
   5  MAXCAP    "Maximum Capacity " /1500000.000000/;
   6  ******************
   7  PARAMETERS DEMAND(I), SIGNIF(I),TOTALDEM;
   8    DEMAND('HISAR')= 114373.000000 ;
   9    DEMAND('NORTH')= 499831.000000 ;
  10    DEMAND('SOUTH')= 294893.000000 ;
  11    SIGNIF('HISAR')= 1.000000 ;
  12    SIGNIF('NORTH')= 1.000000 ;
  13    SIGNIF('SOUTH')= 1.000000 ;
  14    TOTALDEM=SUM(I, DEMAND(I));
  15  ******************
  16   VARIABLES
  17    ALLOC(I)      Allocation to campus i
  18    Z     Maximum bottleneck indicator
  19    OBJ           Objetive Function Value;
  20  POSITIVE VARIABLES ALLOC,Z;
  21  FREE VARIABLE OBJ;
  22  ALLOC.LO(I)=DEMAND(I);
  23  ******************
  24  EQUATIONS
  25    CAPACITY      Allocation is limited with capacity
  26    DEMANDC(I)    Demand contsraint satisfies demands of the campuses
  27    LIMIT   Minize maximum d over a
  28    OBJECTIVE     Objective function;
  29  CAPACITY..      SUM( I , ALLOC(I) ) =E= MAXCAP;
  30  LIMIT(I)..      DEMAND(I)/ALLOC(I) =L= Z;
  31  DEMANDC(I)..    ALLOC(I) =G= DEMAND(I);
  32  OBJECTIVE..     OBJ =E= 5*Z+SUM(I,SIGNIF(I)*DEMAND(I)/ALLOC(I));
  33  ******************
  34  MODEL CampusTr /ALL/;
  35  OPTION NLP=CONOPT;
  36  SOLVE CampusTr USING NLP MINIMIZING OBJ;
  37  DISPLAY OBJ.L,ALLOC.L;
  38  file results /CampusTraffic.csv/;
  39  results.pc=5;
  40  put results;
  41  put "Objective Function Value ",OBJ.l/;
  42  loop(i,put i.tl, ALLOC.l(i)/);
  43  putclose;
COMPILATION TIME     =        0.010 SECONDS      0.1 Mb      WAT-50-094
GAMS 2.50.094  DOS Extended/C                    06/19/02 19:30:42  PAGE      2
Campus Traffic
Equation Listing    SOLVE CAMPUSTR USING NLP FROM LINE 36
---- CAPACITY    =E=  Allocation is limited with capacity
CAPACITY..  ALLOC(SOUTH) + ALLOC(NORTH) + ALLOC(HISAR) =E= 1500000 ;
```

```
     (LHS = 909097, INFES = 590903 ***)
---- DEMANDC     =G=  Demand contsraint satisfies demands of the campuses
DEMANDC(SOUTH)..  ALLOC(SOUTH) =G= 294893 ; (LHS = 294893)
DEMANDC(NORTH)..  ALLOC(NORTH) =G= 499831 ; (LHS = 499831)
DEMANDC(HISAR)..  ALLOC(HISAR) =G= 114373 ; (LHS = 114373)
---- LIMIT       =L=  Minize maximum d over a
LIMIT(SOUTH)..  - (3.391060E-6)*ALLOC(SOUTH) - Z =L= 0 ; (LHS = 1, INFES = 1 ***)
LIMIT(NORTH)..  - (2.000676E-6)*ALLOC(NORTH) - Z =L= 0 ; (LHS = 1, INFES = 1
***)
LIMIT(HISAR)..  - (8.743322E-6)*ALLOC(HISAR) - Z =L= 0 ; (LHS = 1, INFES = 1 ***)
---- OBJECTIVE   =E=  Objective function
OBJECTIVE..  (3.3910605E-6)*ALLOC(SOUTH) + (2.0006762E-6)*ALLOC(NORTH)
     + (8.7433223E-6)*ALLOC(HISAR) - 5*Z + OBJ =E= 0 ; (LHS = -3, INFES = 3 ***)
GAMS 2.50.094  DOS Extended/C                06/19/02 19:30:42  PAGE      3
Campus Traffic
Column Listing      SOLVE CAMPUSTR USING NLP FROM LINE 36
---- ALLOC      Allocation to campus i
ALLOC(SOUTH)
             (.LO, .L, .UP = 294893, 294893, +INF)
        1       CAPACITY
        1       DEMANDC(SOUTH)
 (-3.391060E-6) LIMIT(SOUTH)
 (3.3910605E-6) OBJECTIVE
ALLOC(NORTH)
             (.LO, .L, .UP = 499831, 499831, +INF)
        1       CAPACITY
        1       DEMANDC(NORTH)
 (-2.000676E-6) LIMIT(NORTH)
 (2.0006762E-6) OBJECTIVE
ALLOC(HISAR)
             (.LO, .L, .UP = 114373, 114373, +INF)
        1       CAPACITY
        1       DEMANDC(HISAR)
 (-8.743322E-6) LIMIT(HISAR)
 (8.7433223E-6) OBJECTIVE
---- Z          Maximum bottleneck indicator
Z
             (.LO, .L, .UP = 0, 0, +INF)
       -1       LIMIT(SOUTH)
       -1       LIMIT(NORTH)
       -1       LIMIT(HISAR)
       -5       OBJECTIVE
---- OBJ        Objetive Function Value
OBJ
             (.LO, .L, .UP = -INF, 0, +INF)
     1    OBJECTIVE
GAMS 2.50.094  DOS Extended/C                06/19/02 19:30:42  PAGE      4
Campus Traffic
Model Statistics    SOLVE CAMPUSTR USING NLP FROM LINE 36
MODEL STATISTICS
```

```
BLOCKS OF EQUATIONS       4     SINGLE EQUATIONS        8
BLOCKS OF VARIABLES       3     SINGLE VARIABLES        5
NON ZERO ELEMENTS        17     NON LINEAR N-Z          6
DERIVATIVE POOL           6     CONSTANT POOL          11
CODE LENGTH              78
GENERATION TIME     =       0.030 SECONDS    0.1 Mb     WAT-50-094
EXECUTION TIME      =       0.030 SECONDS    0.1 Mb     WAT-50-094
GAMS 2.50.094  DOS Extended/C                  06/19/02 19:30:42  PAGE      5
Campus Traffic
          S O L V E      S U M M A R Y
   MODEL   CAMPUSTR           OBJECTIVE  OBJ
   TYPE    NLP                DIRECTION  MINIMIZE
   SOLVER  CONOPT             FROM LINE  36
**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE                7.1613
 RESOURCE USAGE, LIMIT          0.055     1000.000
 ITERATION COUNT, LIMIT         5        10000
 EVALUATION ERRORS              0            0
    C O N O P T   386/486 DOS  version 2.042E-003-033
    Copyright (C) ARKI Consulting and Development A/S
                  Bagsvaerdvej 246 A
                  DK-2880 Bagsvaerd, Denmark
 Using default control program.
 ** Optimal solution. Reduced gradient less than tolerance.
 CONOPT time Total                       0.055 seconds
   of which: Function evaluations        0.000 =  0.0%
             Derivative evaluations      0.008 = 14.3%
 Work length =    0.03 Mbytes
    Estimate =    0.03 Mbytes
    Max used =    0.01 Mbytes
                 LOWER     LEVEL     UPPER    MARGINAL
---- EQU CAPACITY   1.5000E+6 1.5000E+6 1.5000E+6      EPS
  CAPACITY    Allocation is limited with capacity
---- EQU DEMANDC     Demand contsraint satisfies demands of the campuses
         LOWER     LEVEL     UPPER    MARGINAL
SOUTH 2.9489E+5 2.9518E+5     +INF       .
NORTH 4.9983E+5 4.9983E+5     +INF       EPS
HISAR 1.1437E+5 7.0499E+5     +INF       .
GAMS 2.50.094  DOS Extended/C                  06/19/02 19:30:42  PAGE      6
Campus Traffic
---- EQU LIMIT       Minize maximum d over a
         LOWER     LEVEL     UPPER    MARGINAL
SOUTH     -INF  -9.775E-4      .        .
NORTH     -INF       .         .     -5.000
HISAR    -INF     -0.838       .        .
                 LOWER     LEVEL     UPPER    MARGINAL
---- EQU OBJECTIVE       .         .         .       1.000
  OBJECTIVE   Objective function
---- VAR ALLOC       Allocation to campus i
```

```
       LOWER     LEVEL     UPPER    MARGINAL
SOUTH 2.9489E+5 2.9518E+5     +INF      .
NORTH 4.9983E+5 4.9983E+5     +INF      .
HISAR 1.1437E+5 7.0499E+5     +INF  3.1543E-6
                LOWER     LEVEL     UPPER    MARGINAL
---- VAR Z             .        1.000     +INF      .
---- VAR OBJ          -INF      7.161     +INF      .
  Z          Maximum bottleneck indicator
  OBJ        Objective Function Value
**** REPORT SUMMARY :        0     NONOPT
                    0 INFEASIBLE
                    0  UNBOUNDED
                    0     ERRORS
GAMS 2.50.094  DOS Extended/C                06/19/02 19:30:42  PAGE      7
Campus Traffic
E x e c u t i o n
----     37 VARIABLE  OBJ.L              =        7.161 Objetive Function
                                             Value
----     37 VARIABLE  ALLOC.L       Allocation to campus i
SOUTH 295181.527,    NORTH 499831.000,    HISAR 704987.473
**** REPORT FILE SUMMARY
RESULTS    C:\GAMS\GAMS\CAMPUSTRAFFIC.CSV
EXECUTION TIME       =        0.020 SECONDS    0.1 Mb      WAT-50-094
USER: Nijaz Bajgoric (PC system #1)                G981007:1547AS-WAT
    Bogazici University                              DC1736
**** FILE SUMMARY
INPUT      C:\GAMS\GAMS\ALLOCATE.GMS
OUTPUT     C:\GAMS\GAMS\ALLOCATE.LST
```

4) C++ CODE FOR GAMS INTERFACE

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <fstream.h>
#include <conio.h>

#define DELIMITER ","
#include "campustraffic.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{    Image1->Picture->LoadFromFile("c:\\gams\\gams\\bountxt.bmp");
```

```
}
//---------------------------------------------------------------------------


void __fastcall TForm1::Initiallize_Parameters()
{  MaxCap=atof(Edit1->Text.c_str());
  SignHisar=atof(Edit2->Text.c_str());
  SignNorth=atof(Edit3->Text.c_str());
  SignSouth=atof(Edit4->Text.c_str());
}
//---------------------------------------------------------------------------
  void __fastcall TForm1::WriteGAMSCode(float demH,float demN,float demS)
  {     FILE *fp2;   //GMS file for the code
        fp2=fopen("allocate.gms","w");
        fprintf(fp2,"$TITLE Campus Traffic\n");
        fprintf(fp2,"$OFFUPPER \n");
        fprintf(fp2,"SETS  I  /SOUTH,NORTH,HISAR/; \n");
        fprintf(fp2,"SCALARS \n");
        fprintf(fp2,"MAXCAP   \"Maximum Capacity \" /%f/; \n",MaxCap);
        fprintf(fp2,"****************** \n");
        fprintf(fp2,"PARAMETERS DEMAND(I), SIGNIF(I),TOTALDEM; \n");
        fprintf(fp2,"  DEMAND('HISAR')= %f ;\n",demH);
        fprintf(fp2,"  DEMAND('NORTH')= %f ;\n",demN);
        fprintf(fp2,"  DEMAND('SOUTH')= %f ; \n",demS);
        fprintf(fp2,"  SIGNIF('HISAR')= %f ;\n",SignHisar);
        fprintf(fp2,"  SIGNIF('NORTH')= %f ;\n",SignNorth);
        fprintf(fp2,"  SIGNIF('SOUTH')= %f ; \n",SignSouth);
        fprintf(fp2,"  TOTALDEM=SUM(I, DEMAND(I)); \n",SignSouth);
        fprintf(fp2,"****************** \n");
        fprintf(fp2," VARIABLES \n");
        fprintf(fp2,"  ALLOC(I)      Allocation to campus i \n");
        fprintf(fp2,"  Z         Maximum bottleneck indicator\n");

        fprintf(fp2,"  OBJ              Objetive Function Value; \n");
        fprintf(fp2,"POSITIVE VARIABLES ALLOC,Z;\n");
        fprintf(fp2,"FREE VARIABLE OBJ; \n");
        fprintf(fp2,"ALLOC.LO(I)=DEMAND(I); \n");
        //fprintf(fp2,"ALLOC.L(I)=DEMAND(I); \n");
        fprintf(fp2,"****************** \n");
        fprintf(fp2,"EQUATIONS \n");
        fprintf(fp2,"  CAPACITY   Allocation is limited with capacity\n");
        fprintf(fp2,"  DEMANDC(I)          Demand contsraint satisfies demands of the
campuses\n");
        fprintf(fp2,"  LIMIT   Minize maximum d over a \n");
        //fprintf(fp2,"  RATIO  Limit of d over a \n");
        fprintf(fp2,"  OBJECTIVE  Objective function; \n");
        fprintf(fp2,"CAPACITY..   SUM( I , ALLOC(I) ) =E= MAXCAP; \n");
        fprintf(fp2,"LIMIT(I)..        DEMAND(I)/ALLOC(I) =L= Z; \n");
        //fprintf(fp2,"RATIO(I)..     DEMAND(I) =G= ALLOC(I)*0.6; \n");
```

```
        fprintf(fp2,"DEMANDC(I)..           ALLOC(I) =G= DEMAND(I);\n");
        //fprintf(fp2,"OBJECTIVE..          OBJ =E= Z;\n");
        //SUM(I,SIGNIF(I)*DEMAND(I)/ALLOC(I));\n");
        //fprintf(fp2,"OBJECTIVE..          OBJ =E= SUM(I,(DEMAND(I)/ALLOC(I)-
1)*(DEMAND(I)/ALLOC(I)-1))-10*Z;\n");
        //fprintf(fp2,"OBJECTIVE..          OBJ =E= SUM(I,(DEMAND(I)/ALLOC(I)-
TOTALDEM/MAXCAP)*(DEMAND(I)/ALLOC(I)-TOTALDEM/MAXCAP));\n");
        fprintf(fp2,"OBJECTIVE..  OBJ =E=
5*Z+SUM(I,SIGNIF(I)*DEMAND(I)/ALLOC(I));\n");
        //fprintf(fp2,"OBJECTIVE..          OBJ =E= SUM(I,(ALLOC(I)-
DEMAND(I))/DEMAND(I));\n");
        //fprintf(fp2,"OBJECTIVE..          OBJ =E= SUM(I,ALLOC(I)-DEMAND(I));\n");
        //fprintf(fp2,"OBJECTIVE..          OBJ =E=
SIGNIF('HISAR')*DEMAND('HISAR')/ALLOC('HISAR')+SIGNIF('NORTH')*DEMAN
D('NORTH')/ALLOC('NORTH')+SIGNIF('SOUTH')*DEMAND('SOUTH')/ALLOC('SO
UTH');\n");
        fprintf(fp2,"****************** \n");
        fprintf(fp2,"MODEL CampusTr /ALL/; \n");
        fprintf(fp2,"OPTION NLP=CONOPT;\n");
        //fprintf(fp2,"SOLVE PHIZER USING LP MAXIMIZING OBJ;\n");
        //fprintf(fp2,"SOLVE CampusTr USING NLP MAXIMIZING OBJ;\n");
        fprintf(fp2,"SOLVE CampusTr USING NLP MINIMIZING OBJ;\n");
        fprintf(fp2,"DISPLAY OBJ.L,ALLOC.L;\n");
        fprintf(fp2,"file results /CampusTraffic.csv/;\n");
        fprintf(fp2,"results.pc=5;\n");
        fprintf(fp2,"put results;\n");
        fprintf(fp2,"put \"Objective Function Value \",OBJ.l/;\n");
        fprintf(fp2,"loop(i,put i.tl, ALLOC.l(i)/);\n");
        fprintf(fp2,"putclose;\n");
        fclose(fp2);

  }

 //---------------------------------------------------------------------
  void __fastcall TForm1::AppendToResultFile(int count)
  {  FILE *fp3;   //Results file : Results.csv
     FILE *fp4; //CampusTraffic.csv
     fp4=fopen("CampusTraffic.csv","r");
     char str1[501];
     float value;


     fp3=fopen("Results.txt","a+");
     if(count==1)
       fprintf(fp3,"ObsNo,Obj,AHisar,ANorth,ASouth\n");
     fprintf(fp3,"%d,",count);

     for(int k=0;k<4;k++)
      { fgets(str1,500,fp4);
        strtok(str1,DELIMITER);
```

```
      value=atof(strtok(NULL,DELIMITER));
    if (k==3)

    fprintf(fp3,"%f \n",value);
    else    fprintf(fp3,"%f,",value);
    }
          fclose(fp4);
            fclose(fp3);

  }
  //-----------------------------------------
  void __fastcall TForm1::CopyResultsFile()
  { FILE *fp1;//Results.txt
   char str1[501];
   float value;
   ofstream results("Results.csv");

results<<"ObsNo"<<","<<"Obj"<<","<<"AHisar"<<","<<"ANorth"<<","<<"ASouth"<<"\
n";

   fp1=fopen("Results.txt","r");
   fgets(str1,500,fp1);

    int c=0;

    while(!feof(fp1))
  {
   fgets(str1,500,fp1);
   str1[strlen(str1)-1]='\0';
   results<<strtok(str1,DELIMITER)<<",";
   for(int i=0;i<4;i++)
    { value=atof(strtok(NULL,DELIMITER));

      if(i<3)
      results<<value<<",";
      else
      results<<value<<"\n";
    }
    c++;

    if(c==NoObs)
      break;

  }
   fclose(fp1);

  }
  //---------------------------------------------

void __fastcall TForm1::Button1Click(TObject *Sender)
```

```
{
  char str1[501];
  //char str2[501];
  int  count=0;
  float demandH, demandN, demandS;

  FILE *fp1;  //DATA file
  //FILE *fp3;   //Results file
  remove("Results.txt");

  Initiallize_Parameters();
  fp1=fopen("Data.csv","r");
  fgets(str1,500,fp1);
   NoObs=0;
  //first line of Excel file,no data stored
   while( !feof(fp1))
   { fgets(str1,500,fp1);
     NoObs++;
   }
   NoObs--;
   ShowMessage(NoObs);
   rewind(fp1);
   fgets(str1,500,fp1);
   while(count<=NoObs)
   {
    count++;

   fgets(str1,500,fp1);

     demandH=atof(strtok(str1,DELIMITER));
     //ShowMessage(demandH);
     demandN=atof(strtok(NULL,DELIMITER));
     //ShowMessage(demandN);
     demandS=atof(strtok(NULL,DELIMITER));
     //ShowMessage(demandS);
     WriteGAMSCode(demandH,demandN,demandS);
     system("cd c\gams\gams");
     system("gams Allocate");
     AppendToResultFile(count);

     //remove("allocate.gms");
     //remove("allocate.lst");
      if(count==NoObs)
     break;

   }
   fclose(fp1);
   CopyResultsFile();
```

```
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Exit1Click(TObject *Sender)
{
      Form1->Close();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
{
  Edit1->Clear();
  Edit2->Clear();
  Edit3->Clear();
  Edit4->Clear();
}
//---------------------------------------------------------------------------
```

5) C++ CODE FOR DATA REVERSIONS AND TRANSFORMATIONS

```
// Data Reversal
#include<string>
#include <fstream.h>
#include<iostream.h>
#include<stdio.h>
#include<cstring>
#include <iostream.h>
#include <stdlib.h>
void main()
{
fstream examplefile;
examplefile.open ("InputFile.txt", ios::in | ios::out);

char buffer[15];
  float array[10000];
  int i=0;
        while(!examplefile.eof())
        {
          examplefile.getline (buffer,100);
          array[i]=atof(buffer);
           i++;
         }
int z;
z=(i-1)/12;
float* num;
num=new  float[z];
cout<<(i-1)<<endl<<z;
int index=0;
float a=0;
for(int k=0;k<(i-1);k++)
  {
    a=a+array[k];
    if(((k+1)%12)==0)
       {
```

```
        num[index]=float(float(a)/12);
        index++;
        a=0;
      }
  }
   examplefile.close();
   examplefile.open ("OutputFile.txt", ios::trunc | ios::out);
    int j=0;
    while(j<z)
    {
       examplefile<<num[j]<<endl;
       j++;;
    }
    examplefile.close();


}
```

Averaging Data According Specified Base

```
#include<string>
#include <fstream.h>
#include<iostream.h>
#include<stdio.h>
#include<cstring>
#include <iostream.h>
#include <stdlib.h>
void main()
{
fstream examplefile;
examplefile.open ("SouthAv.txt", ios::in | ios::out);
int base=0;
cout<<"Ente the time base";
cin>>base;
char buffer[15];
  float array[10000];
  int i=0;
      while(!examplefile.eof())
      {
        examplefile.getline (buffer,100);
        array[i]=atof(buffer);
         i++;
       }
int z;
z=(i-1)/base;
float* num;
num=new  float[z];
cout<<(i-1)<<endl<<z;
int index=0;
float a=0;
for(int k=0;k<(i-1);k++)
  {
```

```
  a=a+array[k];
  if(((k+1)%base)==0)
     {
       num[index]=float(float(a)/base);
       index++;
       a=0;
     }
}
 examplefile.close();
 examplefile.open ("SouthAvOut.txt", ios::trunc | ios::out);
  int j=0;
  while(j<z)
  {
     examplefile<<num[j]<<endl;
     j++;;
  }  examplefile.close(); }
```