ABSTRACT

REDUCTION OF FALSE POSITIVES IN FAULT DETECTION SYSTEM USING A LOW PASS DATA FILTER

by
Ranjit Salunkhe

Network traffic is bursty in nature and exhibits the property of self-similarity, the degree of which is measured by the Hurst parameter. Now, in any network there is always the possibility of the occurrence of fault traffic that can be caused due to faults or malfunction of a network component. Fault Detection Systems that check for traffic anomalies can trigger off an alert on the detection of any traffic behavior that deviates from normal. Such deviation is usually caused when a fault occurs. But in network traffic sudden bursts may occur due to the inherent behavior of a network application. Due to this burst a fault detection system could generate an alert if the burst crosses a preset threshold even though it is not a fault condition. This paper investigates the use of a data filter to reduce such false positives that may be caused due to sudden bursts or spikes.

REDUCTION OF FALSE POSITIVES IN FAULT DETECTION SYSTEM USING A LOW PASS DATA FILTER

by
Ranjit Salunkhe

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree in
Master of Science in Telecommunications

Department of Electrical and Computer Engineering

May 2003

APPROVAL PAGE

REDUCTION OF FALSE POSITIVES IN FAULT DETECTION SYSTEM USING A LOW PASS DATA FILTER

Ranjit Salunkhe

5/21/2003

Dr. Constantine Manikopoulos, Thesis Advisor — Date
Associate Professor, Department Electrical and Computer Engineering, NJIT

5/21/2003

Dr. George Antoniou, Committee Member — Date
Professor, Department of Computer Science, Montclair State University

05/21/03

Dr. Bin He, Committee Member — Date
Senior scientist, XPRT Solutions Inc.

BIOGRAPHICAL SKETCH

Author: Ranjit A. Salunkhe

Degree: Master of Science

Date: May 2003

Date of birth:

Place of birth:

Undergraduate and Graduate Education:

- Master of Science in Telecommunications
  New Jersey Institute of technology, Newark, NJ, 2003

- Bachelor of Engineering in Electronics and Telecommunications
  Maharashtra Institute of Technology, Pune, Maharashtra, India, 1999

Major: Telecommunications

"To my Parents who are a constant source of support and encouragement, and who have taught me a few priceless treasures of sincerity, faith, hope and love."

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Constantine Manikopoulos for serving as thesis advisor and for his guidance and support.

I would like to thank Mr. Jun Li for his constant assistance and direction through the entire course of this research. In addition I wish to express gratitude to my friends Sachin Arora, Gurkirpal Aman Bedi and Kapil Daryani who have helped me in one way or the other through this period and have always provided persistent encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this thesis is to test the possibility of using a low pass data filter to reduce the occurrence of false alarms due to short and sudden bursts in network traffic. Several different network traffic scenarios are simulated and for each scenario certain Management Information Base (MIB) parameters are monitored. For each scenario the value of the Hurst parameter is observed, which is a measure of self-similarity. Observations are made under normal conditions and in the presence of fault traffic.

1.2 Classification of Faults and Fault Management

Network faults can be classified into two categories: hard faults and soft faults. Hard faults are those faults caused when the networks, or some of its elements, are not able to deliver any traffic at all. It may consist of links or nodes failures. Soft faults are those faults caused due to network/service anomaly or performance degradation in various performance parameters such as bandwidth, increase in delay etc.

The Fault Management System can manage the fault in either a reactive or proactive manner. The reactive fault management system waits for a problem and then troubleshoots it, i.e. a problem is solved after it has occurred. The proactive management system monitors certain critical operational thresholds, such as network utilization. In case of a fault the system generates and alert and automatic restore processes are executed if possible to solve the problem or reduce its intensity to minimize downtime.

Proactive detection of network failures and performance degradations is a key to rapid fault recovery and thus robust networking, and has been receiving increasing attention lately. Proactive network performance and fault management is concerned with automatic and adaptive performance monitoring/analysis and fault detection of network and service faults in the midst of networks performance fluctuation and evolutions.

CHAPTER 2

SELF-SIMILARITY AND NOAH EFFECT

2.1 Self-Similarity

In recent years, it has been established that network packet traffic is self-similar in nature. Actual traffic exhibits correlations over a wide range of time scales i.e. it has long range dependence. Self-similarity describes the phenomenon where a certain property of an object is preserved with respect to scaling in space and/or time. In other words, the burstiness of network traffic is independent of time scales. Network traffic that is modeled using Markovian models e.g. the Poisson distribution model does not take into account the self-similar nature of traffic. This leads to inaccurate modeling which when applied to a gigantic network like the Internet, leads to performance problems. The fact that a time series is self-similar manifests itself in a number of different ways, among them-

- "Burstiness" across a large range of time scales.
- Slowly decaying autocorrelations, implying long-range dependence.
- Sample variance decays more slowly than the reciprocal of the sample size.

To explain self-similarity, first some terms must be introduced. The main parameter connected with self-similarity is time scale of the process. If x_i denotes samples of the process (traffic pattern) in the basic time scale, the corresponding process with the time scale m is obtained with the following formula:

$$x_k = 1/m \sum_{i=(k-1)m+1}^{km} x_i$$

It implies that k^{th} sample of scaled process is obtained as a mean value over m subsequent samples of the original process.

Autocorrelation Function (ACF) – Autocorrelation is a statistical measure of the relationship, if any, between a random variable and itself, at different time lags. The autocorrelation coefficient can range between +1 (very high positive correlation) and -1 (very high negative correlation).

Long-Range Dependence- Long-range dependence measures the memory of a process. Intuitively, distant events in time are correlated. This correlation is captured by the autocorrelation function (ACF), which measures how similar a series is with the shifted version of itself. For LRD data the ACF decays very slowly to zero. On the contrary, short-range dependence is characterized by quickly decaying correlations. The strength of long-range dependence is quantified by the Hurst exponent (H). A series exhibits LRD when $1/2 < H < 1$. Furthermore, the closer H is to 1, the stronger the dependence of the process.

2.2 Hurst Estimators

There are many estimators that are used to estimate the value of the Hurst parameter. Below is a list of some estimators –

- Absolute Value method- where an aggregated series $X^{(m)}$ is defined, using different block sizes m. The log-log plot of the aggregation level versus the absolute first moment of the aggregated series $X^{(m)}$ should be a straight line with slope of (H-1), if the data is long-range dependent.

- Variance method- where the log-log plot of the sample variance versus the aggregation level must be a straight line with slope β greater than -1. In this case $H = 1 + \beta/2$.
- R/S method- A log-log plot of the R/S statistic versus the number of points of the aggregated series should be a straight line with the slope being an estimation of the Hurst exponent.
- Whittle estimator- The method is based on the minimization of a likelihood function, which is applied to the periodogram of the time series. It gives an estimation of H and produces the confidence interval. It does not produce a graphical output.

The Variance-Time method is used in the experiment for estimating.

2.3 Noah Effect

The superposition of many strictly alternating, independent and identically distributed ON/OFF sources, each of which exhibits a phenomenon called the Noah effect results in self-similar aggregate traffic. The ON/OFF source is a model where the ON and OFF sources strictly alternate, where the ON periods are independent and identically distributed (i.i.d.) and the OFF periods are i.i.d. and the ON and OFF periods are independent of each other.

These ON/OFF periods have high variability or infinite variance, and this produces aggregate traffic that is self-similar or long-range dependent. This indicates that there is a relation between the parameters describing the high variability, which is the Noah Effect and self-similarity.

Nearly all the heavy-tailed random variables are used in models, which exhibit infinite variance. This has been called the Noah effect due to the infinite variance, which shows high variability. The distribution time of the ON and OFF periods are heavy tailed with parameter $\alpha 1$ and $\alpha 2$. A heavy-tailed distribution has infinite variance and the portion of the tail distribution depends on the α parameter.

An example of a heavy-tailed distribution is the Pareto distribution. The general form of Pareto distribution is: $F(x) = P[X \leq x] = 1-(a/x)^{\alpha}$ where $a, \alpha \geq 0$ and $x \geq a$.

If the ON and OFF period length distribution is heavy-tailed, they satisfy the property: $P[X>x] \sim x^{-\alpha}$, as $x \to \infty$, $0<\alpha<2$.

The properties of the ON/OFF source aggregation can be used to develop a simple model to generate a self-similar arrival pattern by aggregating several sources and modeling the ON and OFF length periods of each source with a Pareto distribution.

For $\alpha \leq 2$, the distribution has infinite variance or Noah effect. A common method of estimating value of α is by using the Hill estimator. Let X1, X2, ..., Xn be the observed statistics and write the ordered statistics as $X(1) \leq X(2) \leq .... \leq X(n)$ then the Hill esitmator of alpha is-

$$\alpha = 1/k \left(\sum_{i=0}^{i=k-1} (\log X_{(n-i)} - \log X_{(n-k)} \right)^{-1}$$

In practice α is plotted as a function of k for a range of values of k. In the presence of tail behavior in the data, a typical Hill plot varies considerably for small values of k but becomes more stable as more data points in the tail of the distribution are included.

CHAPTER 3

SNMP AND MANAGEMENT INFORMATION BASE

3.1 Simple Network Management Protocol

The Internet Engineering Task Force (IETF) first defined SNMP (Simple Network Management Protocol) in 1989. Since then, SNMP has become an industry standard for controlling networking devices from a single management application.

SNMP is a set of network management protocols and functions that communicate using the Internet Protocol (IP) stack. SNMP allows network managers to isolate and troubleshoot faults on multi-vendor networks, configure devices on a network, and monitor network performance and status. SNMP normally uses User Datagram Protocol for communication.

Two versions of SNMP exist: SNMP version 1 (SNMPv1) and SNMP version 2 (SNMPv2). Both versions have a number of features in common, but SNMPv2 offers enhancements, such as additional protocol operations. SNMP version 2 (SNMPv2) is an evolution of the initial version, SNMPv1. Originally, SNMPv2 was published as a set of proposed Internet standards in 1993; currently, it is a draft standard. As with SNMPv1, SNMPv2 functions within the specifications of the Structure of Management Information (SMI). The Structure of Management Information defines the rules for describing management information, using a template, known as Abstract Syntax Notation One (ASN.1). SNMPv2 offers a number of improvements to SNMPv1, including additional protocol operations.

3.2 The Network Management Architecture

The Internet network management architecture, of which SNMP is a part, is based on the interaction of many entities. The entities comprising a network management system are listed below-

- Network elements – They are the hardware devices (i.e. computers, routers, and terminal servers) connected to a network. They are sometimes called managed devices.
- Agents – Agents are software modules that reside in network elements. They collect and store management information.
- Managed object - A managed object is a characteristic of something that can be managed. Managed objects differ from variables, which are particular object instances.
- Management information base (MIB) – A MIB is a collection of managed objects residing in a virtual information store. Collections of related managed objects are defined in specific MIB modules
- Syntax notation - A syntax notation is a language used to describe MIB's managed objects in a machine-independent format. Consistent use of a syntax notation allows different types of computers to share information. Network management systems use Abstract Syntax Notation 1 (ASN.1) to define both the packets exchanged by the management protocol and the objects that are to be managed.
- Structure of Management Information (SMI) - It defines the rules for describing management information (the SMI is defined using ASN.1)

- Network management station (NMS)/Manager - Sometimes called consoles, these devices execute management applications that monitor and control network elements. Network management stations are usually workstation-caliber computers with fast CPUs, substantial memory, and abundant disk space. At least one NMS must be present in each managed environment.
- Parties - Newly defined in SNMPv2, a party is a logical SNMPv2 entity that can initiate or receive SNMPv2 communication. Each SNMPv2 party comprises a single, unique party identity, a logical network location, a single authentication protocol, and a single privacy protocol. SNMPv2 messages are communicated between two parties. A SNMPv2 entity can define multiple parties, each with different parameters. For example, different parties can use different authentication and/or privacy protocols.
- Management protocol - A management protocol is used to convey management information between agents and NMS. SNMP is the Internet community's de facto standard management protocol.

Bridges, Hubs, Routers or network servers are examples of managed devices that contain managed objects. These managed objects might be hardware, configuration parameters, performance statistics, as well as complex variables that correspond to TCP/IP data structures, such as ARP cache and IP routing tables and so on, that directly relate to the current operation of the device in question. These objects are arranged in the virtual information database, called Management Information Base (MIB).

Managers invoke an SNMP client on their local computer, and use the client to contact one or more SNMP servers, that run on remote machines. SNMP uses a fetch-

store paradigm. A separate standard for a Management Information Base defines the set of variables that SNMP servers maintain as well as the semantics of each variable.

Usually a typical agent-

- Implements full SNMP protocol.
- Stores and retrieves management data as defined by the MIB.
- Can asynchronously signal an event to the manager.
- Can be a proxy for some non-SNMP manageable network node.

A typical manager-

- Functions as a Network Management Station (NMS).
- Implements full SNMP Protocol.
- Can query agents, get responses from agents, set variables in agents, and acknowledge asynchronous events from agents.

3.3 SNMP Basic Commands & Operations

3.3.1 SNMP Basic Commands

Managed devices are monitored and controlled using four basic SNMP commands: **read, write, trap**, and **traversal** operations.

- The **read** command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.
- The **write** command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.

- The **trap** command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.
- The **traversal** operations are used by the NMS to determine the variables that are supported by a managed device and to sequentially gather information in variable tables, such as a routing table.

3.3.2 SNMPv1 Protocol Operations

SNMP is a simple request/response protocol. The network-management system issues a request, and managed devices return responses. This behavior is implemented by using one of four protocol operations: Get, GetNext, Set, and Trap.

The Get operation is used by the NMS to retrieve the value of one or more object instances from an agent. If the agent responding to the Get operation cannot provide values for all the object instances in a list, it does not provide any values. The GetNext operation is used by the NMS to retrieve the value of the next object instance in a table or a list within an agent. The Set operation is used by the NMS to set the values of object instances within an agent. The Trap operation is used by agents to asynchronously inform the NMS of a significant event.

3.3.3 SNMPv2 Protocol Operations

The Get, GetNext, and Set operations used in SNMPv1 are exactly the same as those used in SNMPv2. However, SNMPv2 adds and enhances some protocol operations. The SNMPv2 Trap operation, for example, serves the same function as that used in SNMPv1,

but it uses a different message format and is designed to replace the SNMPv1 Trap. SNMPv2 also defines two new protocol operations: GetBulk and Inform.

The GetBulk operation is used by the NMS to efficiently retrieve large blocks of data, such as multiple rows in a table. GetBulk fills a response message with as much of the requested data as will fit. The Inform operation allows one NMS to send trap information to another NMS and to then receive a response. In SNMPv2, if the agent responding to GetBulk operations cannot provide values for all the variables in a list, it provides partial results.

3.4 MIB Information Structure

The structure of management information (SMI) defines the structure of the MIB information and the allowable data types. The SMI identifies how resources within the MIB are represented and named. The philosophy behind SMI is to encourage simplicity and extensibility within the MIB.

The SNMP specification includes a template, known as an Abstract Syntax Notation One (ASN.1) OBJECT TYPE macro, which provides the formal model for defining objects and tables of objects in the MIB. The following keywords are used to define a MIB object:

- Syntax - Defines the abstract data structure corresponding to the object type. The SMI purposely restricts the ASN.1 constructs that can be used to promote simplicity.
- Access - Defines whether the object value may only be retrieved but not modified (read-only) or whether it may also be modified (read-write).

- Description - Contains a textual definition of the object type. The definition provides all semantic definitions necessary for interpretation; it typically contains information of the sort that would be communicated in any ASN.1 commentary annotations associated with the object.

A Management Information Base can be viewed as an abstract tree with an unnamed root. The leaves of the tree are individual data items. Object identifiers (IDs) uniquely identify objects in the tree and are organized hierarchically, with specific numbers assigned by different organizations. As shown in the figure below the object ID structure defines three main branches: Consultative Committee for International Telegraph and Telephone (CCITT), International Organization for Standardization (ISO), and joint ISO/CCITT.

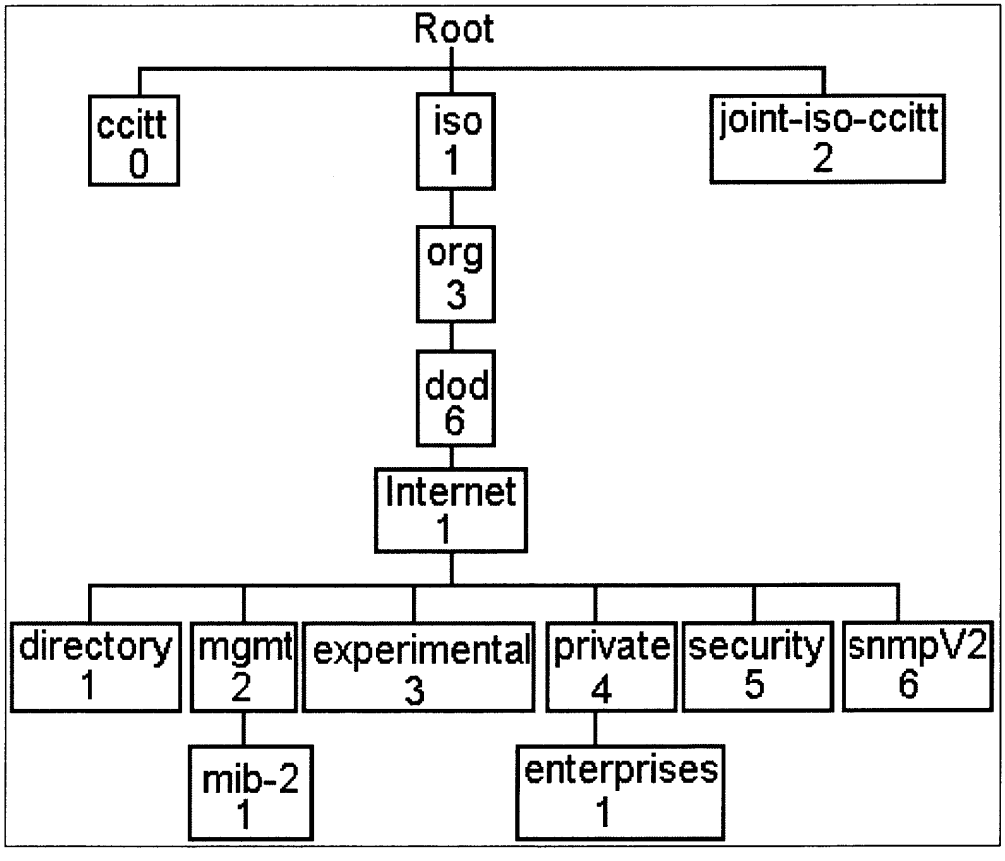

Figure 3.1 The main ISO/CCITT Tree

Most current activities are concerned with the portion of the ISO branch defined by object identifier 1.3.6.1 and dedicated to the Internet community. The current Internet-standard MIB (MIB-II) is defined in RFC 1213 and contains 171 objects grouped by protocol (including TCP, IP, UDP and SNMP), and other categories including system and interfaces. The MIB tree is extensible by virtue of experimental and private branches. Vendors can define their own private branches to include instances of their own products. The private sub-tree enables private organizations to enhance the usefulness of SNMP. The basic MIB structure including MIB-II (object ID = 1.3.6.1.2.1) is shown below.

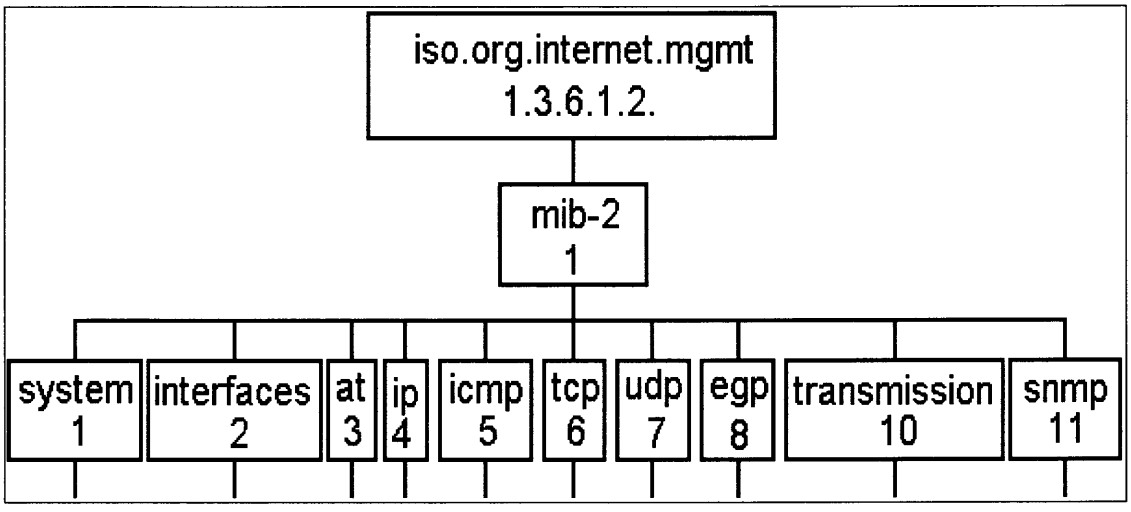

Figure 3.2 MIB node and its object identifiers

MIB contains only essential elements. There is no need to allow individual objects to be optional. The objects are arranged into the following groups:

- SystemOBJECT IDENTIFIER ::= { mib-2 1 }
- Interfaces OBJECT IDENTIFIER ::= { mib-2 2 }
- Address Translation (deprecated) OBJECT IDENTIFIER ::= { mib-2 3 }

- IP OBJECT IDENTIFIER ::= { mib-2 4 }
- ICMP OBJECT IDENTIFIER ::= { mib-2 5 }
- TCP OBJECT IDENTIFIER ::= { mib-2 6 }
- UDP OBJECT IDENTIFIER ::= { mib-2 7 }
- EGP OBJECT IDENTIFIER ::= { mib-2 8 }
- Transmission OBJECT IDENTIFIER ::= { mib-2 10 }
- SNMP OBJECT IDENTIFIER ::= { mib-2 11 }

There are two reasons for defining these groups: to provide a means of assigning object identifiers and to provide a method for implementations of managed agents to know which objects they must implement.

CHAPTER 4

NETWORK MONITORING APPLICATION

WhatsUp Gold (product of Ipswich) is a powerful network-monitoring tool that gives network administrators greater control and understanding of their networks and helps keep mission critical networks up and running. It is an easy-to-use network mapping, monitoring, notification, and performance reporting application that helps network administrators and engineers to catch and fix network problems. It can automatically map the entire network and monitor critical devices and services. When WhatsUp Gold detects a problem, it can immediately notify the administrator via a pager, beeper, email, or other methods. The robust reporting capability, improves the performance of the network over time.

WhatsUp Gold automatically detects the devices and hosts in a network and then creates a network map. It polls all the systems in the network with a polling frequency that can be set by the user. With the auto discovery wizard, it creates an accurate representation of the network based on information contained in the host computer or on the network. An interactive web interface allows administrator to check the status of network devices and perform routine administrative tasks via the Web. Administrators can designate remote map viewing rights to individuals on a per map basis. It can also proactively track network and systems resource. The administrator can monitor usage of CPU, memory, disk space as well as bandwidth utilization of network routers and switches.

The map of the network used is shown below-

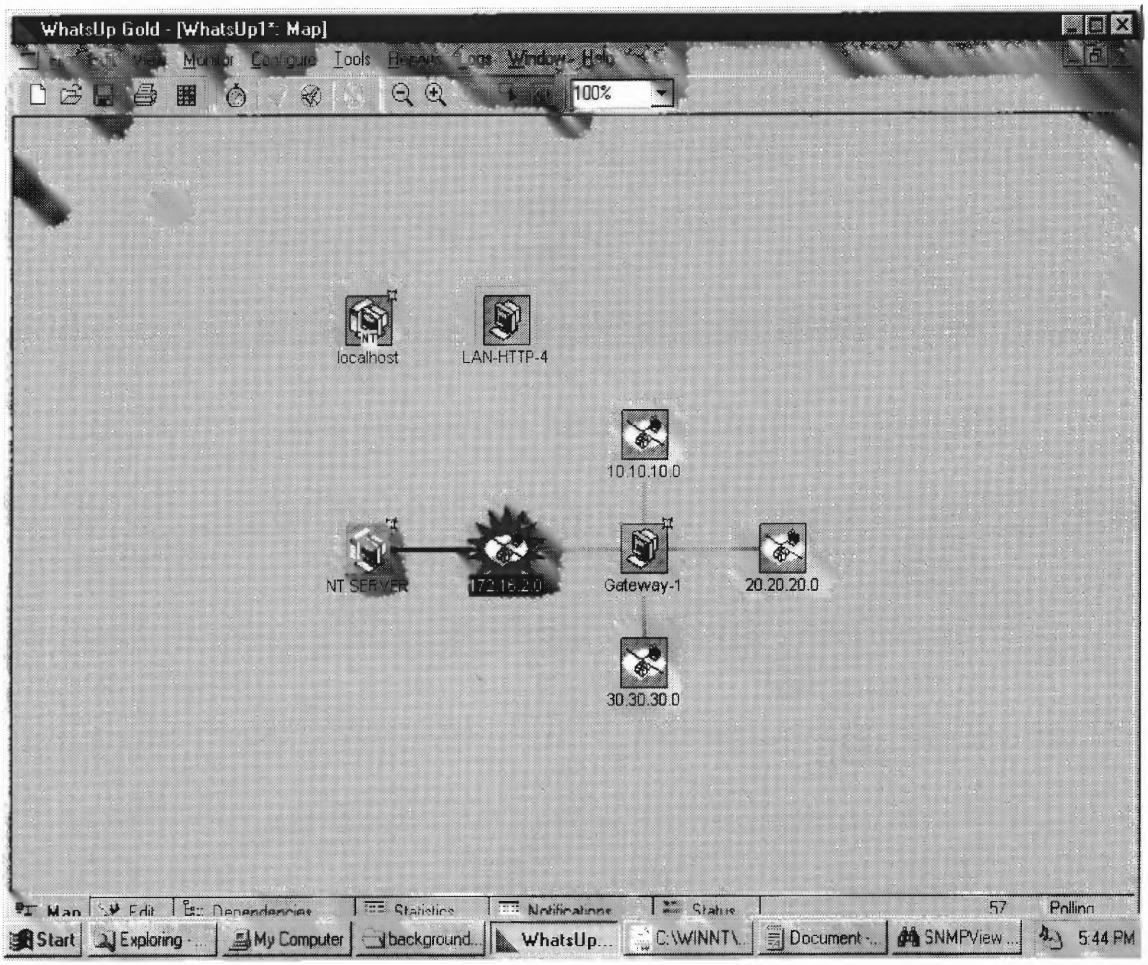

Figure 4.1 Network Map

An icon represents each device in the network. Its color determines the status of the device. Green represents the device is active and responding. Red represents the device is not responding and it has timed out. Grey represents the device is not configured. MIB parameters can be monitored only on those devices that are SNMP enabled.

4.1 Steps involved in collecting MIB data

In order to monitor MIB values on a device the steps carried out are as follows:

Step 1- In order to view the interfaces of the SNMP manageable device, right click on the device in the map and then select SNMP view from the pop-up menu. The SNMP View displays an icon for each interface on the device as shown below.

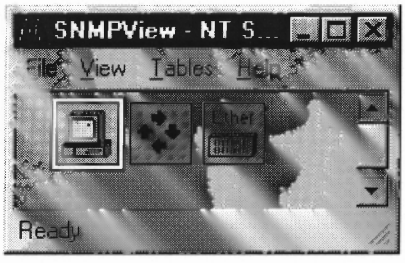

Figure 4.2 SNMP View

The first icon gives the MIB parameters of the system, the second icon gives the loop back interface parameters and the third icon gives the Ethernet interface MIB parameters.

Step 2- To view the SNMP parameters for an interface right click the interface icon, and select "View MIB". The SNMP viewer displays the entire MIB tree as shown in the figure below.

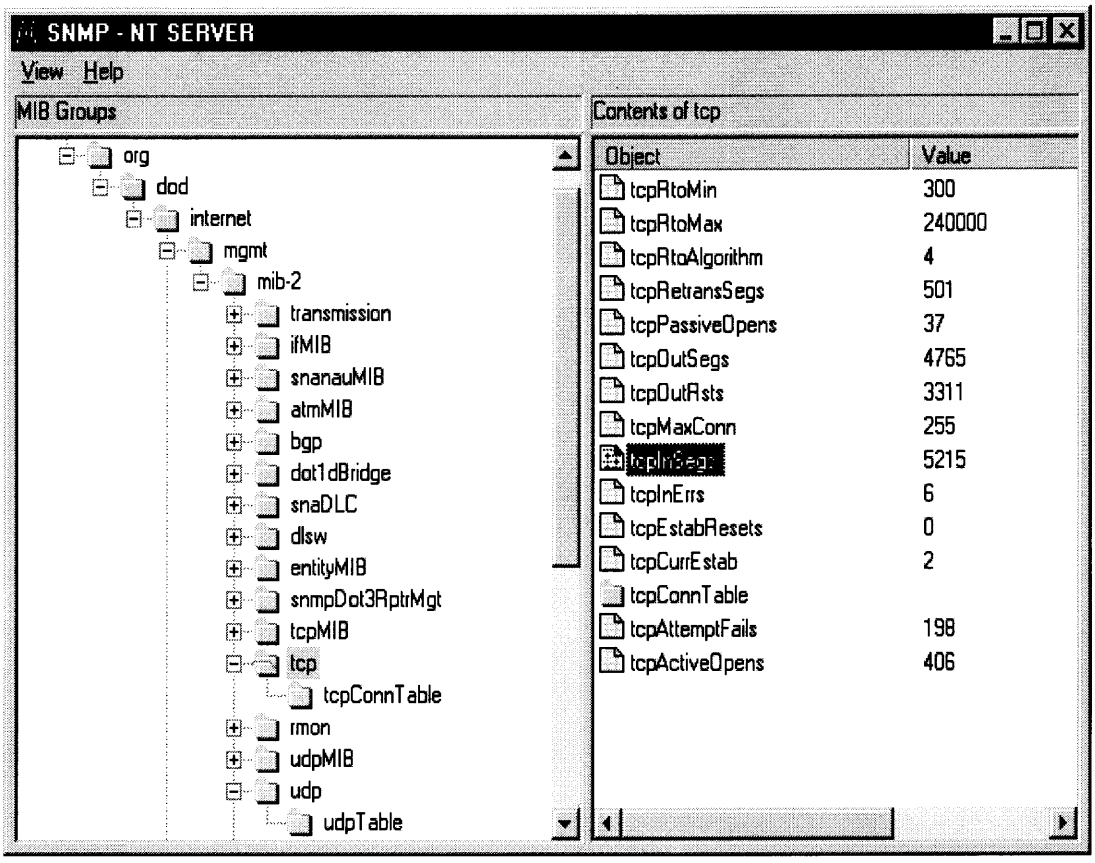

Figure 4.3 MIB tree for selected device

Step 3- To monitor any SNMP object right click on that SNMP object and then select "Monitor". The SNMP graph utility comes up and begins graphing the selected SNMP object.

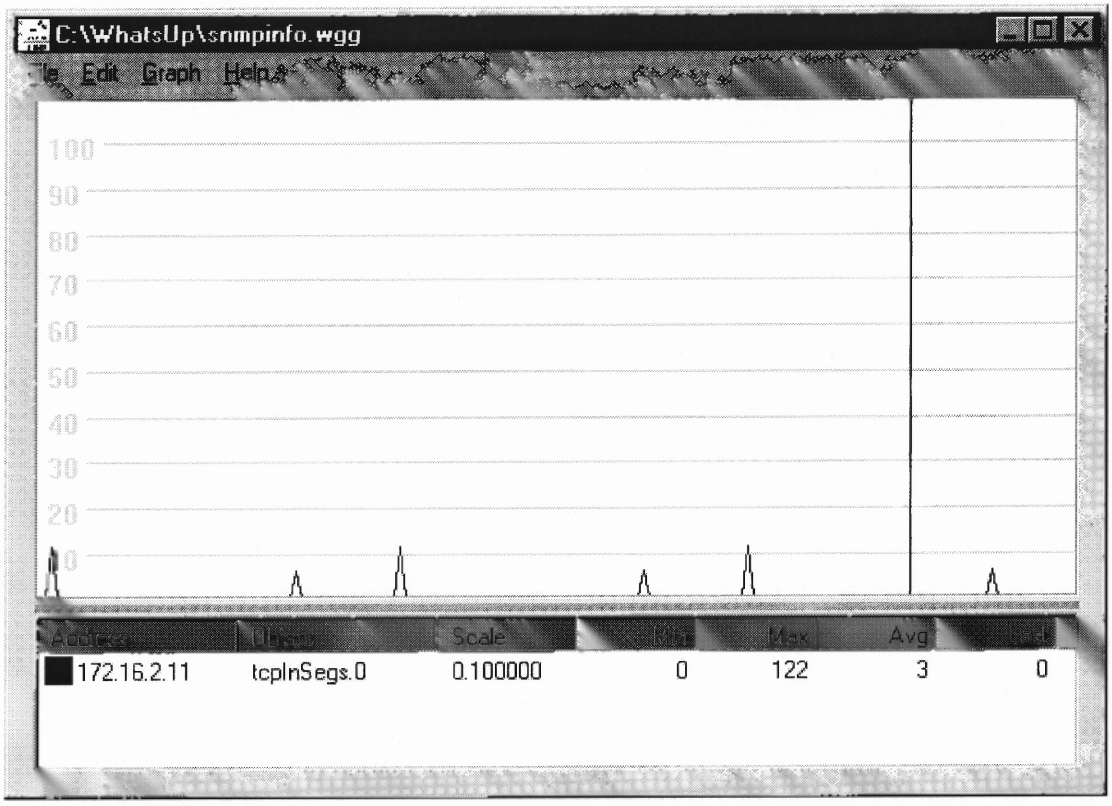

Figure 4.4 Graph utility to monitor MIB parameters

A number of SNMP objects can be graphed simultaneously in the same graph window. This can be achieved by adding SNMP objects to the same graph.

CHAPTER 5

THE EXPERIMENT

The first step is to simulate network traffic with different scenarios and then simulate fault traffic of different intensities.

5.1 Network Traffic and Fault Traffic Simulation

The network in the experiment consists of three subnets: a traffic subnet that is used primarily to generate/simulate network traffic, an attack subnet from where the fault traffic is generated and a victim subnet which consists of a host where the network traffic as well as attack traffic is directed to.

A client-server application was written to simulate network traffic, which is referred to as the background traffic, and a similar application was written to simulate fault traffic. The background traffic server and fault traffic server are run on a particular host in the victim network. Background traffic clients are run on host machines in the traffic subnet and on some host machines in the victim network. The fault traffic client is run from a single host in the attack subnet.

The background traffic clients generate packets of sizes that follow the Pareto distribution. The rate at which the packets are generated follows the Weibull distribution.

The fault traffic model generates network fault traffic in a periodic pulse. The time period in seconds is given as the input. The size of the packet can follow any of the following distribution: Exponential Distribution, Constant Distribution or Pareto

Distribution. The packet rate can follow any of the following distribution: Exponential, Constant, Pareto or Weibull Distribution.

The figure below shows a client generating traffic. The next figure shows background server applications receiving packets from the client applications.

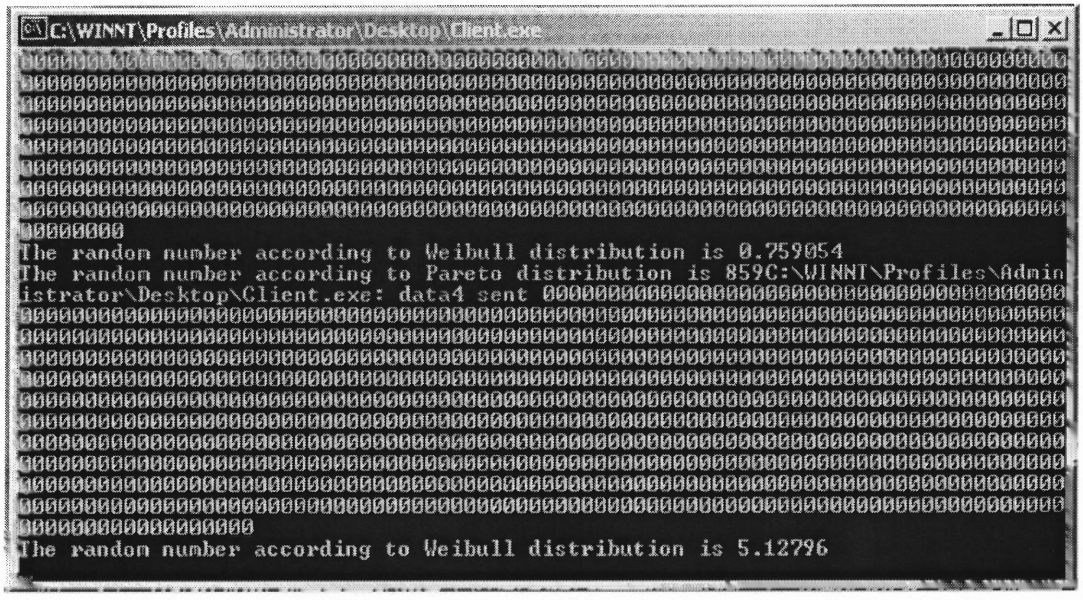

Figure 5.1 Client generating background traffic

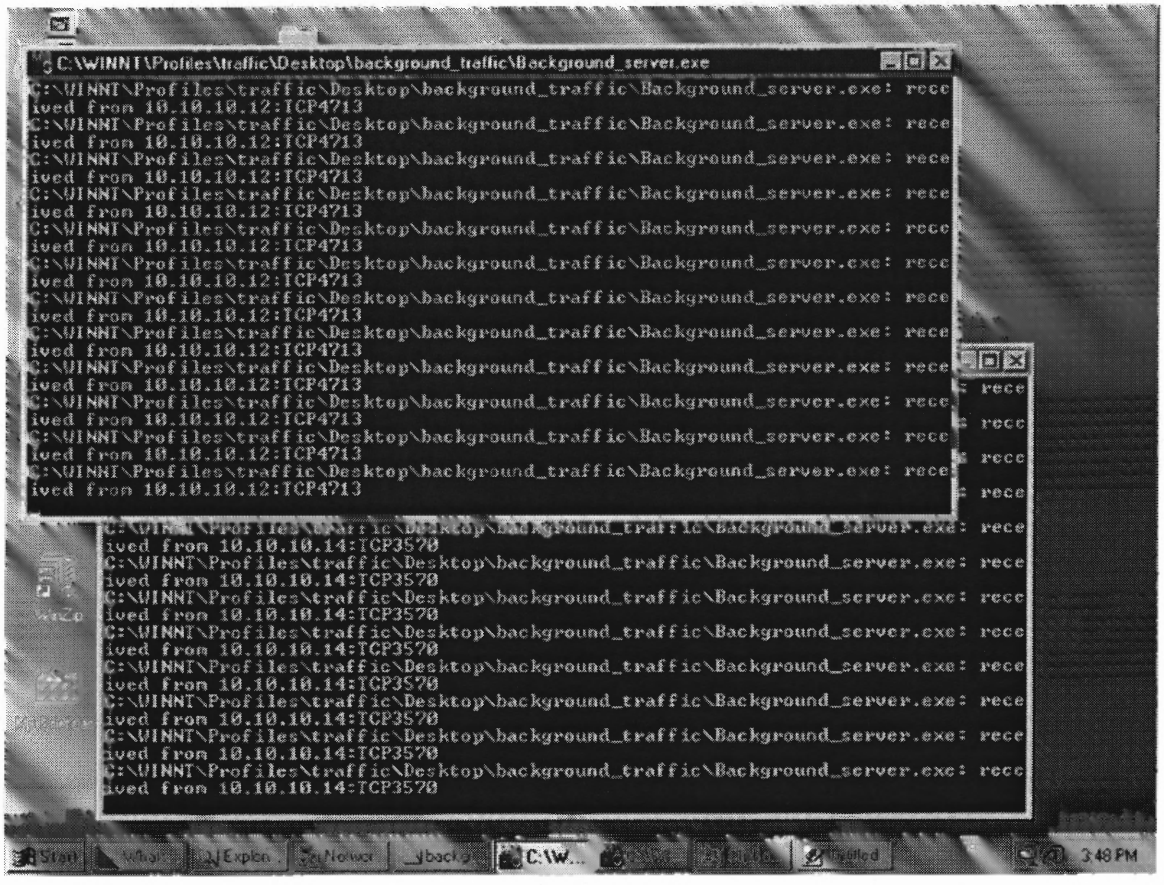

Figure 5.2 Background traffic server windows

The number of clients generating traffic was initially taken as seven. After some scenarios the number of clients was reduced to six and then five.

Once the clients start generating traffic the fault client is set to run after every two thousand seconds. Then using WhatsUp Gold, selected MIB parameters are monitored on the victim machine.

The MIB parameters monitored are-

Interface Group- ifInNUCastPkts, ifInUCastPkts, ifInDiscards, ifInErrors, ifInOctets, ifOutNUCastPkts, ifOutUCastPkts, ifOutDiscards, ifOutOctets

IP Group- ipInDiscards, ipInAddErrors, ipInHdrErrors, ipInReceives, ipOutRequests, ipOutDiscards

UDP Group- udpInDatagrams, udpInNoPorts, udpInErrors, udpOutDatagrams

TCP Group- tcpPassiceOpens, tcpActiveOpens, tcpAttemptFails, tcpEstabResets, tcpInErrs, tcpInSegs, tcpOutSegs, tcpRetransSegs

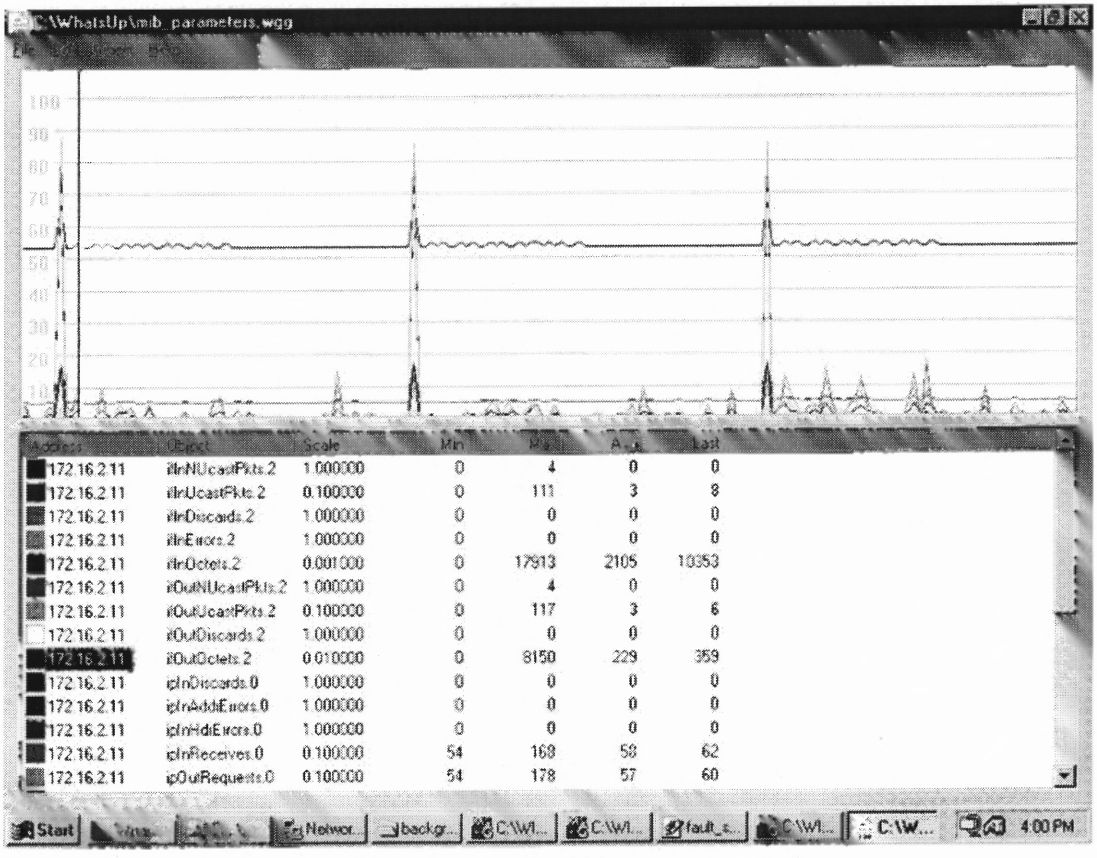

Figure 5.3 Monitoring MIB parameters

The values of the parameters are monitored every second. The difference between the monitored value and the previous monitored value is recorded into a file. This difference gives the number of packets that arrive per second.

5.2 Calculation of Hurst parameter

Several data sets were collected for different fault traffic scenarios. The data sets consisted of the following – data sets with alternating time periods of background traffic and fault traffic, data sets with only background traffic, data sets with only fault traffic and data sets with background traffic followed by some duration of fault traffic. Code was written in Matlab to take the collected data as input and then calculate the Hurst parameter. The mean value of each data set is also observed. The main focus is on the background traffic. Shown below are some example plots.

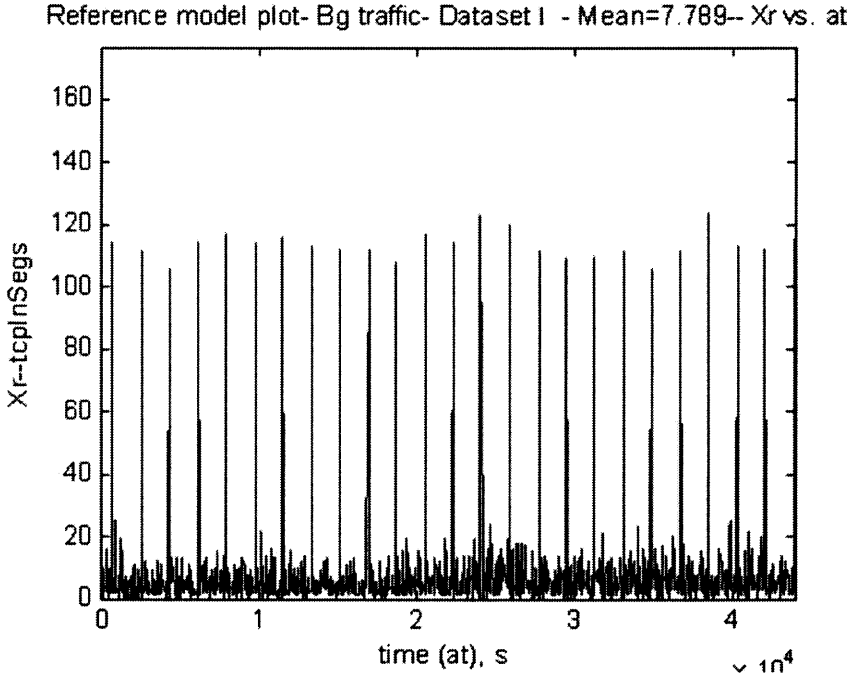

Figure 5.4 Plot showing only background traffic

The figure below shows the variance-time plot being used to calculate the Hurst parameter of the above shown data set.

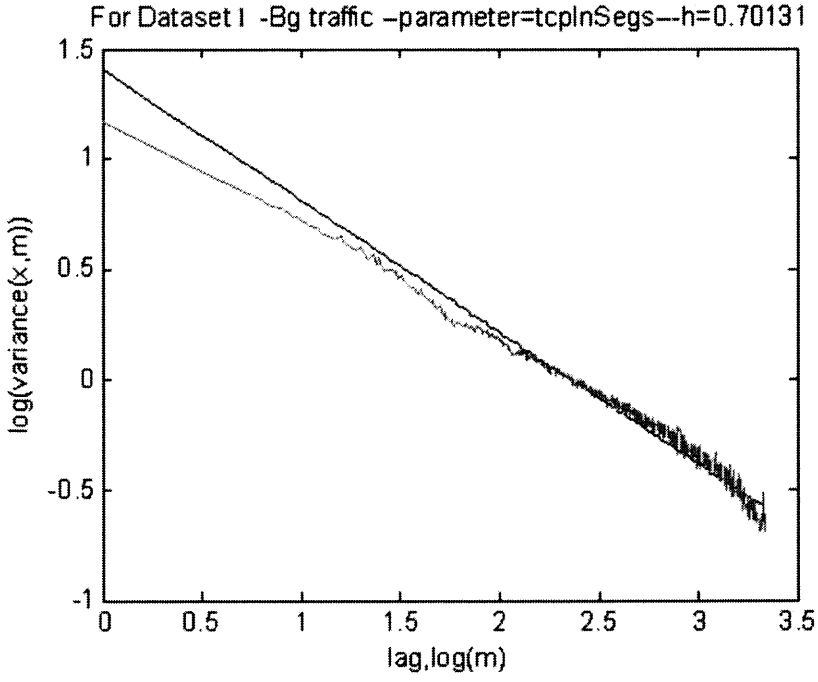

Figure 5.5 Variance-time plot to obtain Hurst parameter

The value of the Hurst parameter was calculated for the different scenarios of background traffic using the variance-time method. In the table below are the results for the data sets, H is the value of Hurst parameter and the mean is given by m.

Table 5.1 Values of the Hurst parameter and the mean of background traffic data sets.

Data set -	**1**		**2**		**3**		**4**		**5**	
	H	m	H	m	H	m	H	m	H	m
Background Traffic	0.70	7.78	0.78	8.06	0.73	7.55	0.69	7.73	0.72	7.58
Data set -	**6**		**7**		**8**		**9**		**10**	
	H	m	H	m	H	m	H	m	H	m
Background Traffic	0.66	7.67	0.75	6.81	0.69	6.55	0.93	5.82	0.92	7.01

5.3 Savitzky-Golay Filter for data smoothing

After observing the data plots, sudden spikes of different length can be noticed in all the data sets. These sudden spikes are normal and can occur in regular traffic. But these spikes may be seen as anomalies either due to an attack or due to a network fault condition by an IDS. The intrusion detection system may trigger an alert on the basis of this anomaly. So in the next step of the experiment, a method was developed to prevent or reduce the occurrence of false alerts. When fault traffic occurs there is an increase in traffic for durations which last several minutes. So it is different from the normal spikes, which are also caused due to sudden increase in traffic, but they usually last only for a few seconds.

The use of a data filter like the Savitzy-Golay filter is now considered. Savitzky-Golay filtering can be thought of as a generalized moving average. The filter coefficients are derived by performing un-weighted linear least-squares fit using a polynomial of a degree less than or equal to the chosen number of 2n + 1 channels. For this reason, a Savitzky-Golay filter is also called a digital smoothing polynomial filter or a least-squares smoothing filter. A higher degree polynomial makes it possible to achieve a high level of smoothing without attenuation of data features. This filter is thus used to reduce the spikes and thus obtain a lower threshold for detection of anomalies by the IDS.

Below are plots of a data set before filtering and after filtering using a window size of 33 and degree of 4.

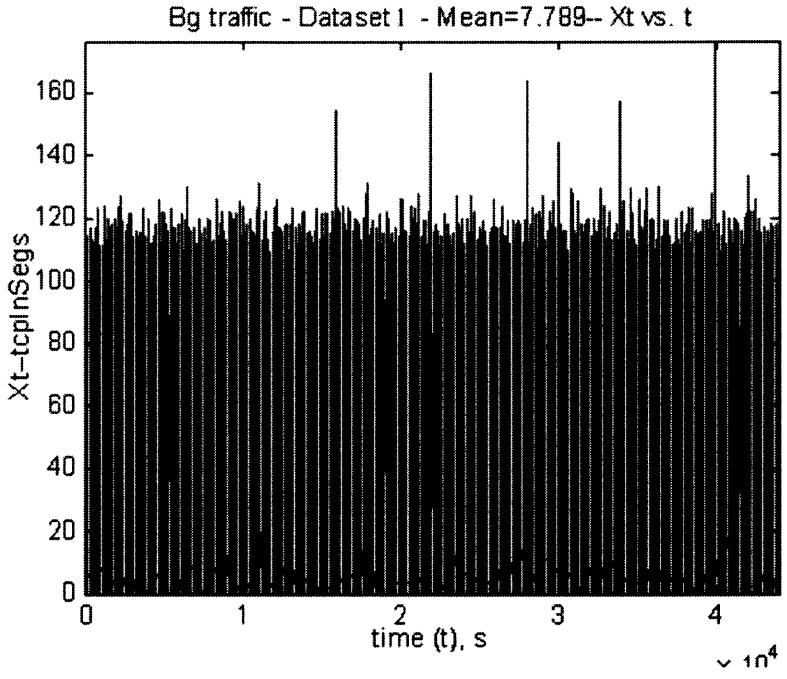

Figure 5.6 Background traffic plot before filtering

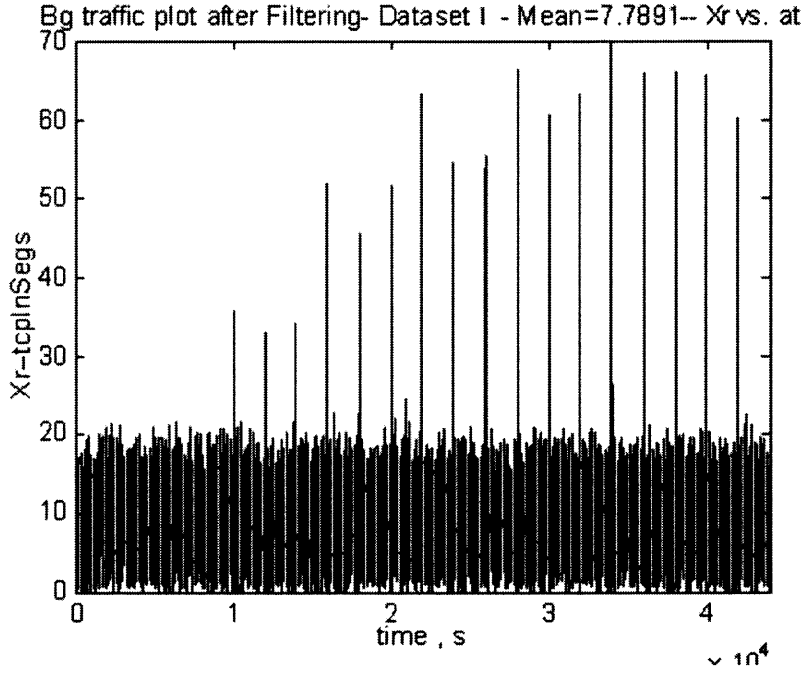

Figure 5.7 Background traffic plot after filtering

From figures 5.8 and 5.9 it can be observed that the spikes in the plot before filter touch up to 180, but after filtering the maximum level is 70. It can also be seen that the mean value of the data does not vary. And as shown in the table below we can see that the Hurst parameter also remains the same.

Table 5.2 Values of the Hurst parameter and the mean of data sets for background traffic before filter and after filter.

Data set -	**1**		**2**		**3**		**4**		**5**	
	H	m	H	m	H	m	H	m	H	m
Original Background Traffic	0.70	7.78	0.78	8.06	0.73	7.55	0.69	7.73	0.72	7.58
Background Traffic after filtering	0.71	7.78	0.79	8.06	0.73	7.55	0.69	7.73	0.73	7.58
Data set -	**6**		**7**		**8**		**9**		**10**	
	H	m	H	m	H	m	H	m	H	m
Original Background Traffic	0.66	7.67	0.75	6.81	0.69	6.55	0.93	5.82	0.92	7.01
Background Traffic after filtering	0.66	7.67	0.75	6.81	0.69	6.55	0.93	5.82	0.92	7.01

Shown below are two plots for a data set containing background as well as fault traffic, one showing the plot before filter and the next one showing the plot after it is filtered.

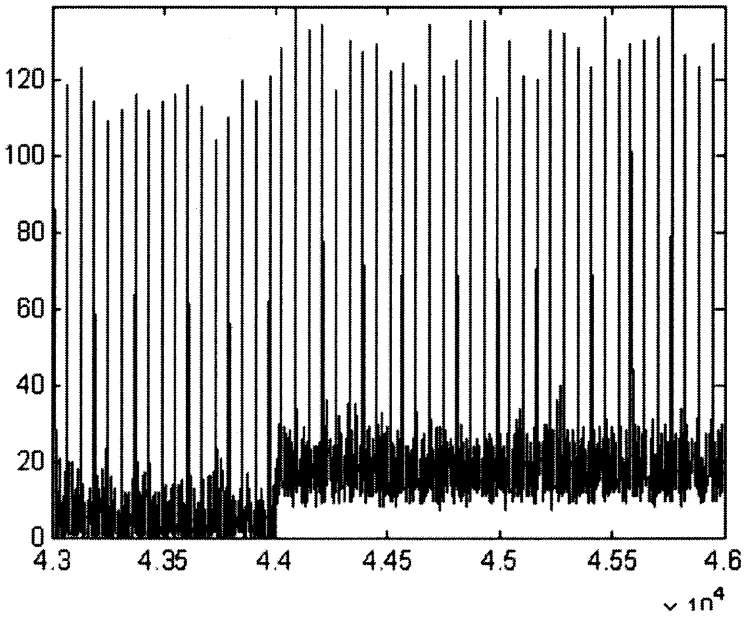

Figure 5.8 Data containing background and fault traffic: before filter

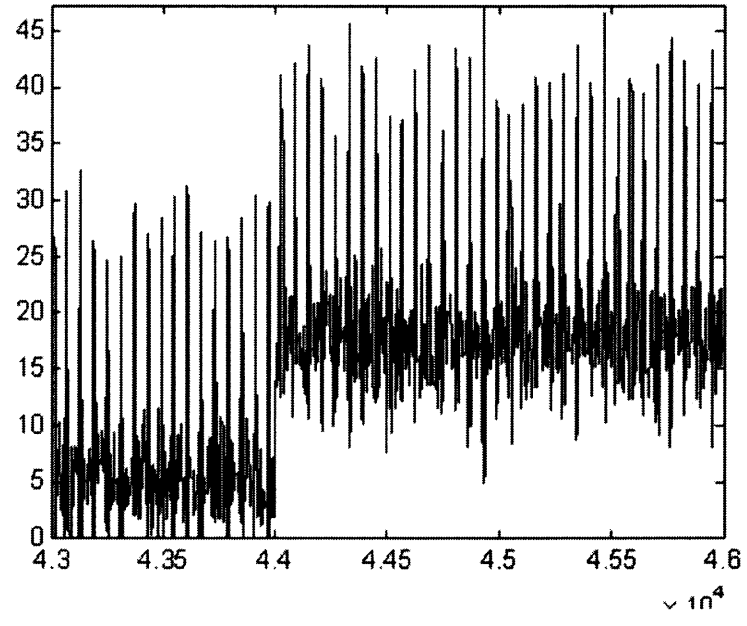

Figure 5.9 Data containing background and fault traffic: after filter

It can be noticed, from the observations made, that the data after filtering has the same mean and value of the Hurst parameter. Thus this indicates that the correlation structure of the data has not changed and also after observing the plots we notice that the peak values of the sudden short bursts are reduced. If such sudden bursts occur in network traffic due to a requirement by an application then it would be normal behavior for that application. But if a fault detection system has been set to generate an alert on a particular threshold value that is lower than the value of the sudden burst then it will generate an alert. Now instead if the filter is used then the value of the threshold is lowered and also the number of false positives is reduced.

CHAPTER 6

CONCLUSION

It can be seen that with the use of the Savitzky-Golay filter the high value of the spikes could be lowered. Due to this the threshold level for fault detection used by a fault management system can be reduced to a smaller value. Also after observing the plots it can be noticed that the number of false positives can be reduced by a considerable amount and fault traffic may be identified more distinctly.

REFERENCES

1. W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson "On the Self-Similar-Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, Feb. 1994.

2. http://www.cs.ucr.edu/~tkarag/papers/SELFIS-Tutorial.pdf (7 May 2003)

3. http://www.cisco.com/warp/public/535/3.html (7 May 2003)

4. http://www.ietf.org/rfc/rfc1213.txt (7 May 2003)

5. W. Willinger, M.S. Taqqu, R. Sherman, D. V. Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," April 15, 1997.

6. http://www.et.put.poznan.pl/snmp/intro/indexint.html (25 April 2003)