# THE DESIGN OF A LOGICAL TRAFFIC ISOLATION FORENSIC MODEL

**[1]Innocentia Dlamini, [2]Martin Olivier**

Information and Computer Security Architectures Research Group (ICSA)

Department of Computer Science, University of Pretoria

[1]idlamini@csir.co.za, [2]molivier@cs.up.ac.za

ABSTRACT

Currently, network evidence used in a court of law can be lacking and inadequate for prosecution purposes, due to a loss of packets during the network transmission. This packet loss in turn may be caused by the congestion of data transmitted over the network, further delaying the transmitted data. This paper extends the work on a forensic model for traffic isolation based on Differentiated Services (DiffServ). This model intends to solve the packet loss problem that can result to insufficient evidence. It isolates suspicious traffic from the normal flow by placing it on the dedicated route using DiffServ prioritising characteristics. This avoids traffic congestion of the suspicious traffic. The LTI model further includes the preservation station which serves to record all suspicious traffic before it is forwarded to its destination. This paper looks at the analysis and design of the logical traffic isolation model using various UML design artefacts. By incorporating various design algorithms, this paper aims at designing the more flexible and reliable system, with a minimal loss of evidence.

KEY WORDS

Differentiated Services, Preservation station, Network Forensics, Suspicious traffic, Unified Modelling Language.

# THE DESIGN OF A LOGICAL TRAFFIC ISOLATION FORENSIC MODEL

## 1 INTRODUCTION

Traditionally, when network forensic investigations are performed, it is always better for the investigator if the crime is still on progress. Investigators do not have to shut down the communication in order to gather enough evidence. This paper presents the design of the concept on a forensic model for Logical Traffic Isolation (LTI) based on Differentiated Services (DiffServ), proposed by Strauss et al [5]. This model intends to solve the packet loss problem that can result to insufficient evidence. It isolates suspicious traffic from the normal flow by placing it on the dedicated route using DiffServ prioritizing characteristics. This avoids traffic congestion of the suspicious traffic. The LTI model further includes the preservation station which serves to record all suspicious traffic before it is forwarded to its destination.

The LTI model utilises the DiffServ to isolate malicious traffic logically from normal traffic [5]. This could well reduce cost because DiffServ is a standard technique. If a DiffServ infrastructure is already in place where an investigation needs to be performed, evidence collection could be facilitated with minimal changes to the network. The DiffServ approach allows Network Forensic investigators to attach both their marking station (ingress router) in isolating the suspicious traffic and preservation station to a cyber victim's network to investigate the case at hand. The advantage of this approach is that it requires minimal network downtime and most importantly minimal network reconfiguration. This DiffServ-based scheme makes provision for a preservation station to store records of the isolated traffic with a view to later analysis [5].

However, in order to minimise network transmission problems such as transmission delays and high network traffic, the preservation station only stores records related to malicious network traffic. While the proposal seems plausible, it has not been tested yet to prove this system’s viability. In order to have a successful and reliable implementation of this system, this paper uses various design technique in modelling LTI model. A Unified Modelling Language (UML) technique is favoured; it provides abundant diagrams which can explicitly depict most of the processes and the interaction of the components of the LTI model. The rest of the paper is structured as follows: Section 2 discusses the architecture of the LTI model. Section 3 presents a design of the LTI model by system design technique, while Section 4 concludes the paper.

## 2 THE LTI ARCHITECTURAL MODEL

The waterfall model is still the more basic and sill use model when developing the system. The design stage started with carefully revision of the requirements of the system that were defined by Strauss et al [5], and we further added what might be useful. Some of the requirements includes the type of the network set up already in place should be DiffServ network. The system should further solve the problem of inefficient and inadequate evidence by introducing a station for capturing identified packets and also that can be easily plug on the network when intrusion has been detected. This should further allow the system to conducting the investigation while cyber- crime is committed, i.e. the live -network forensics.

For experimentation reasons, the system should have seven nodes; the two nodes on a traffic generator, which acts as users and generates the normal and suspicious traffic randomly; there are three nodes on the DiffServ network, it is the three routers, including ingress, immediate and egress routers; the sixth node is the preservation station for recording the traffic that has been detected suspicious .The last one is the sink server receiving and processing the requests generated. (Both traffic generator and the sink server are additional nodes). The system should be able to isolate the two generated traffics within the DiffServ network, and further record the suspicious packets at the preservation station. The system is designed with the following assumptions made: The network has its intrusion detection system in place; there are various users transmitting data. (Which is represented by Traffic generator for experimentation purposes); the receiver or the destination node is represented by the sink server.

The requirements of the system serve as the foundation of this study as such; these requirements resulted in the following implementation infrastructure of the LTI model in figure 1. It provides a conceptual view of the LTI model based on DiffServ for isolating suspicious traffic. The model consists of two traffic generators on the client side to initiate suspicious and normal traffic; and the DiffServ network with three routers (ingress, interior and egress) for experimental purposes. The preservation station ensures forensic soundness and system reliability, while the sink server receives and responds to all the requests generated by the traffic generator. This nodal setup is however for experimentation purposes only.

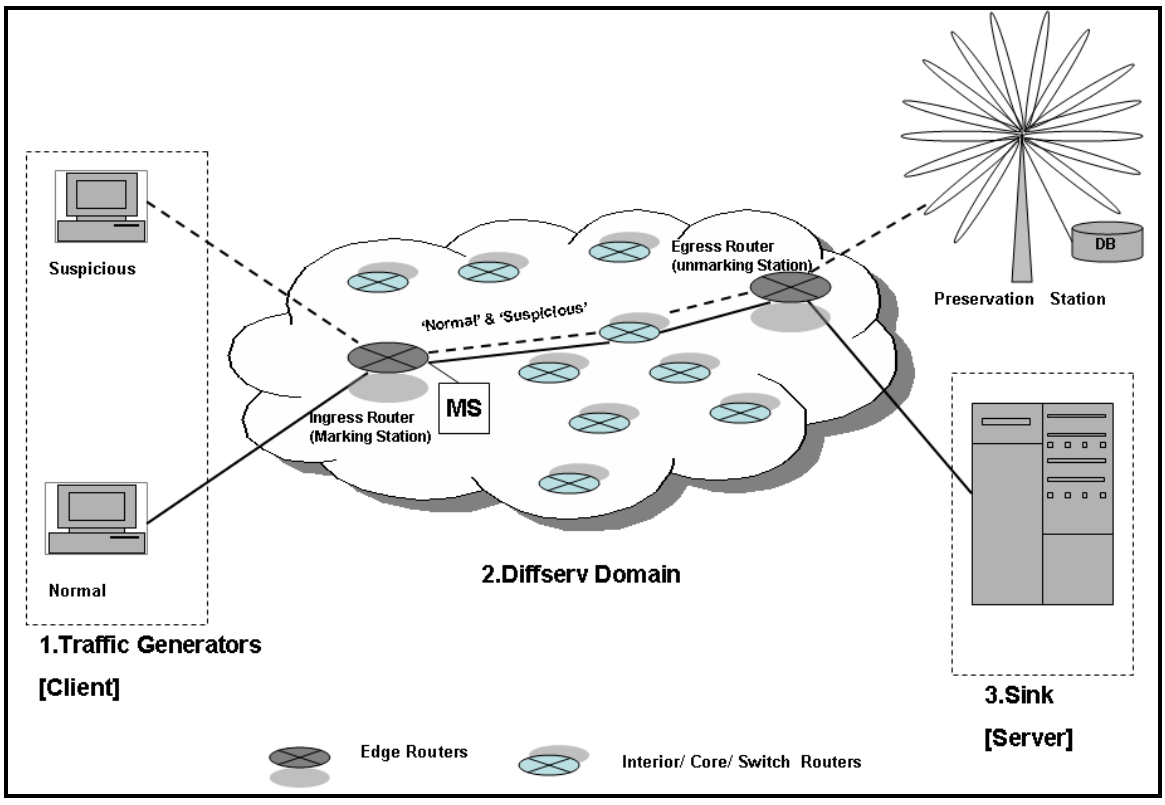


*Figure 1: The Implementation Infrastructure of the LTI model Using DiffServ*

The two clients generate normal and suspicious traffic and forward these packets onto the DiffServ domain. The ingress edge router at the entrance boundary of the DiffServ domain is the first domain recipient and serves as a marking station. This router is responsible for *packet*

*classification* and has *marking*, *shaping* and *dropping* capabilities. The ingress router marks the suspicious traffic by using the packet classifier and forwards them to the nearest core router. The core routers are found within the centre of the DilffServ domain, and they simply forward traffic towards the egress router. The egress router is found at the exit boundary of the DiffServ domain. It unmarks the traffic and decides the destination of each network packet according to its behaviour: compromised traffic is forwarded to the preservation station and then to the sink server, while normal traffic is sent directly to the sink server. In a network-related cyber incident, the investigator searches the preservation station when conducting his/her investigation and captures all recorded suspicious network packets as evidence. The LTI model is further formalized in various UML diagrams. This includes the Use-case diagram, Sequence diagram and the Activity diagram. The following section discusses the LTI model design in details.

## 3 SYSTEM DESIGN TECHNIQUES

The second step on the design of the LTI model is its representation using the UML design technique. Even though the UML is not a panacea, but it does simplifies our work. These diagrams do not include too much detailed information; they simply depict the applicability and the functionality of the LTI model and the involvement of the requirements of the system.

### 3.1 Use Case Diagram

One of the significant uses of the UML use case diagram (UCD) is to associate actor with the use cases (i.e., services or processes) provided by a system. In figure 2, the Network Investigator (the actor) interacts with the system by performing different processes.

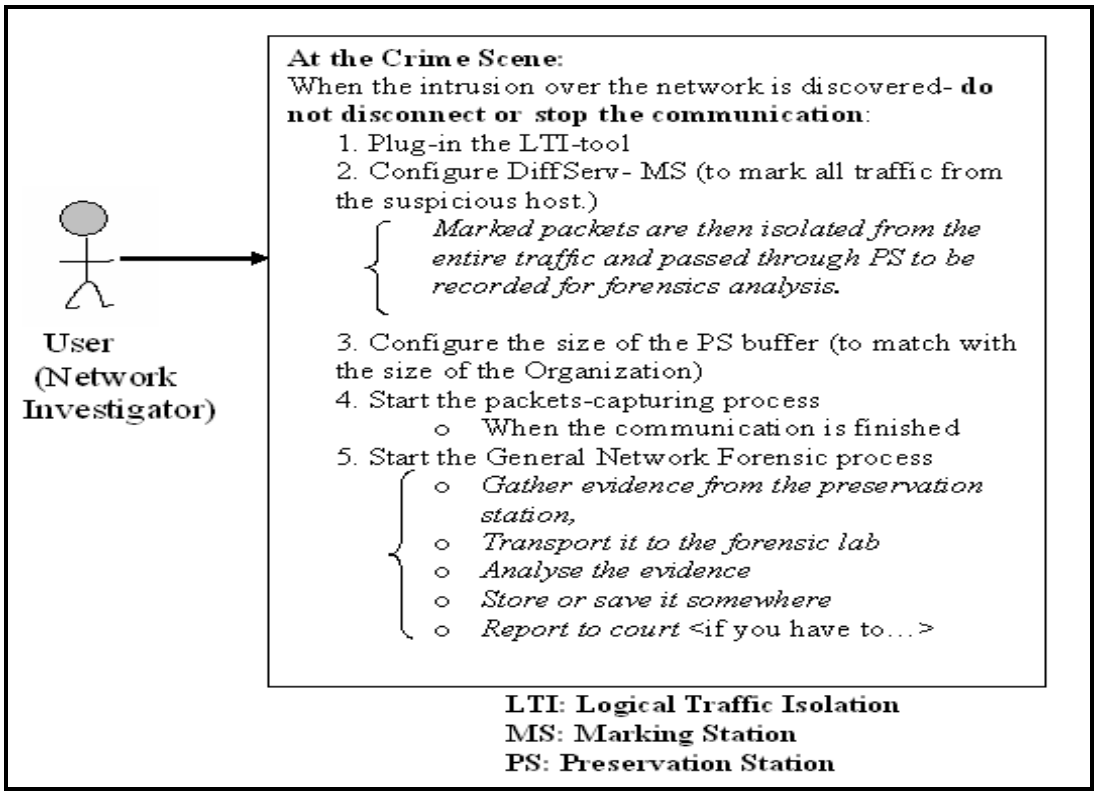


*Figure 2. Use-Case Diagram for the Logical Traffic Isolation system*

These processes are: 1) Plug the LTI tool, 2) Configure DiffServ-MS station, 3) Configure the buffer-size of PS station, 4) Start the packets-capturing process, and 5) Start Network Forensic process of the system. The LTI system involves the processes during the intrusion processes; that is it starts immediately when the intrusion has been detected.

When an investigator arrives at the crime scene, the suspicious communication has to be left on and running in order to capture and record the detected packets. The LTI tool is aimed to be easily plugged on the affected network and minor configuration on it to suite the size of the network at hand can be applicable. These configurations include enabling the marking station to mark the packets from the suspicious host; also the buffer-size of the preservation station has to be considered, it should correspond with the size of the organization. When these configurations are finished, the investigator can start the tool to capture suspicious packets. When the communication is finished, then the normal network forensic processes can be initiated, as in figure 2, number 5.

### 3.2 Sequence Diagram

A sequence diagram is also part of the UML. It is a kind of interaction diagram that shows how processes operate with one another and in what order. Figure 3 depicts five component, where by others are included for the experimentation purposes.

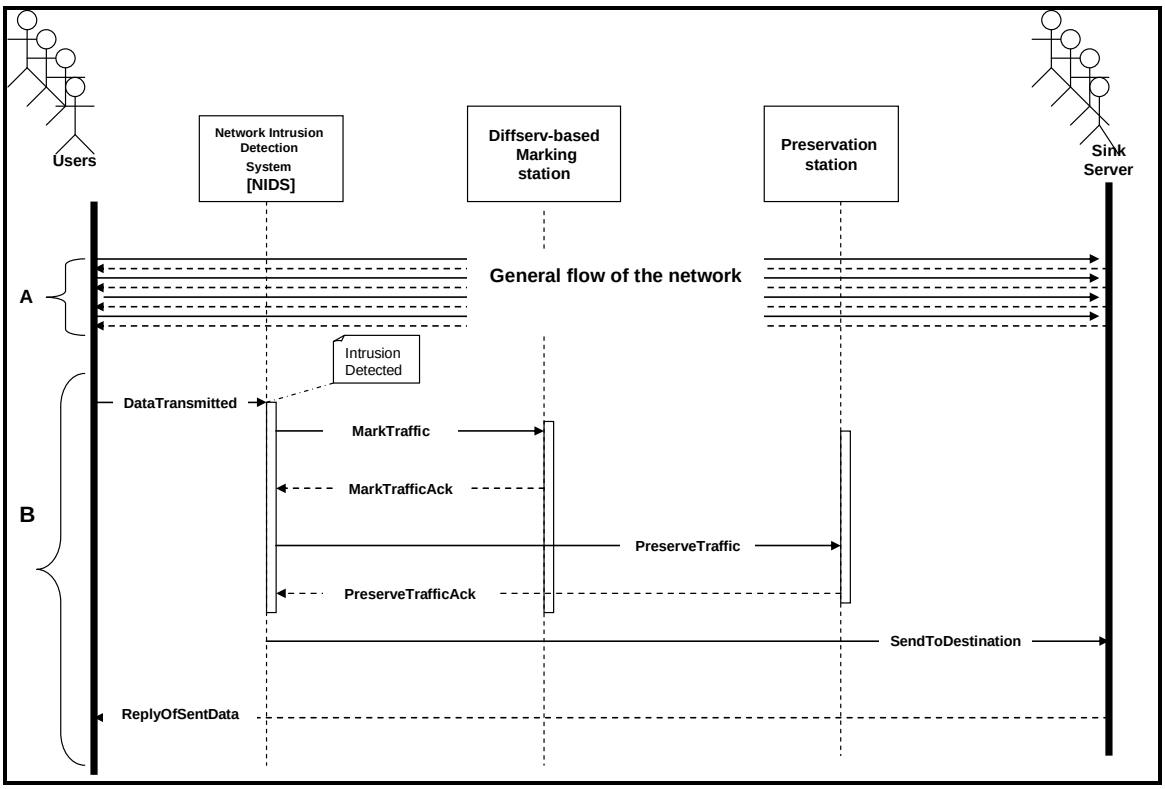


*Figure 3. Sequence Diagram for the Logical Traffic Isolation system*

These components include, the users, NIDS (which can be any detection system, it differs from the organization to organization), DiffServ network, Preservation station and the sink server. As mentioned above, the NIDS and sink server are part of the model. There are two categories of

traffic that can be generates. It is the suspicious and the normal traffic. This is randomly generated, which means that mostly, it can only be the normal traffic and rarely, suspicious traffic can also be generated.

In Figure 3, A represents the normal network traffic flowing from the users to their destinations (which is represented by the sink server for supporting the experimentation of the LTI model). This network passes through all the nodes, except the preservation station. Traffic at B is when there is suspicious traffic that is generated. The NIDS system reports to the DiffServ module to mark this traffic with higher priority and provide proper routing methods to it, as it is the special and significant traffic to the cyber investigators. Suspicious traffic is then routed to the preservation station for the recording, and later on to its destination. The ReplyOfSentData shown on dotted lines depicts the situation whereby the destination user is replying the initiation user's request. The same procedure in B can also be applied for the response of the targeted system on the destination.

### 3.3 Activity Diagram

Activity diagrams provide another ability, to clarify which actor carries out which activity. Consider the Activity diagram in figure 4; activities are broken down further, into different activities. This diagram starts with normal flow of the network, assuming that there is a detection system that is already in place. It is served as a deciding device as it informs the network administrator about any detected incident.

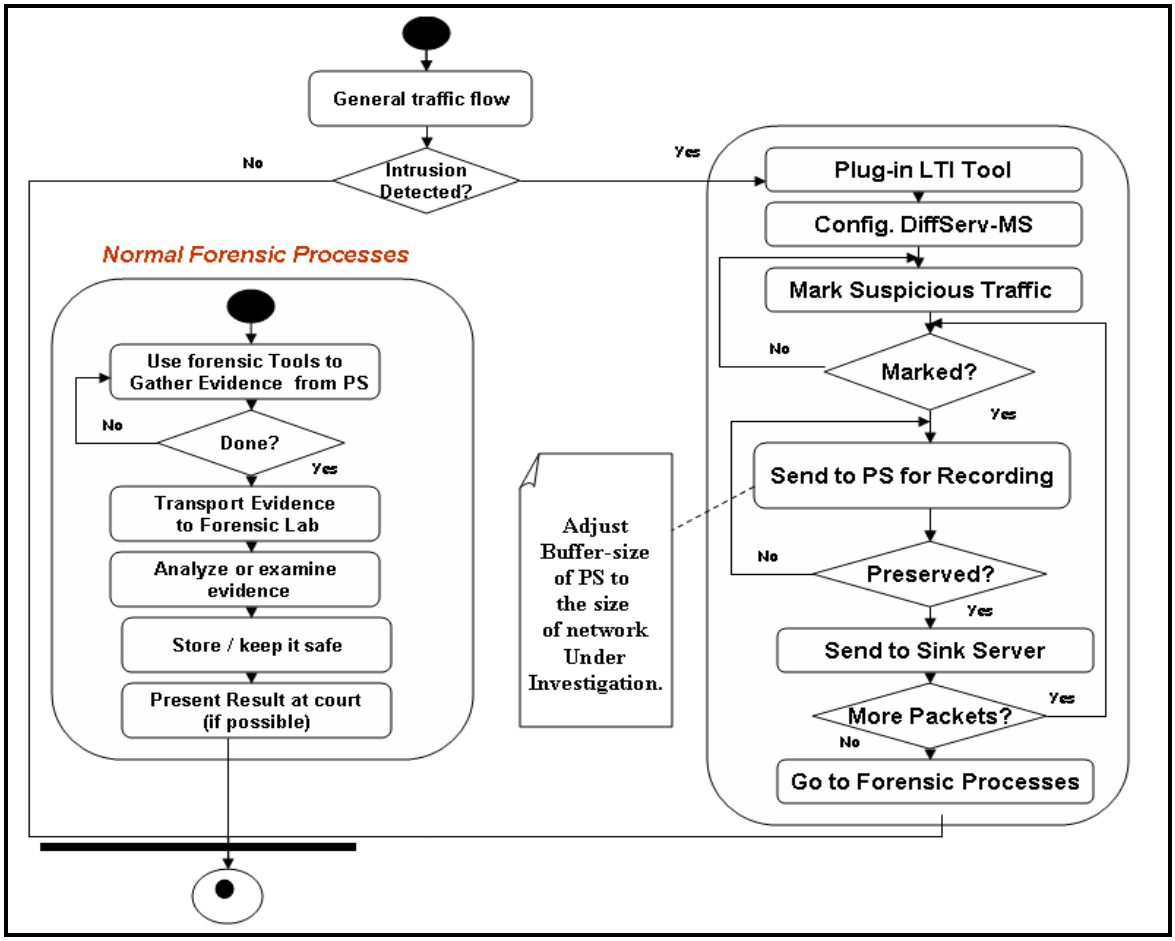

*Figure 4. Activity Diagram for the Logical Traffic Isolation system*

The network administrator easily plugs-in the LTI tool, then Configure the marking station to mark the packets of the detected traffic. System checks whether the packets have been marked, if they are not marked, they sent back to be marked otherwise they are forwarded to the preservation station to be recorded. The system checks again whether packets have been recorded and send them back to be recorded if they have not, or else they are sent to their destination (we used sink server in our model as a supporting node).

The system further checks whether there are more packets that are still detected and go back to the marking station to start the process again, if this is a case. Otherwise, Network Forensic investigation process has to take place. Its activities are also included in a form of activity diagram. This process is initiated by using the approved Network Forensic tool to collect the evidence preserved from the preservation station. This step is carried out till it is done; otherwise, the evidence is gathered more. The next activity that is performed is the transportation of forensic evidence to the forensic lab; followed by the analysis of evidence gathered. The analyses of the evidence are store in a safe place, which is safe enough waiting for the court date. This evidence is further presented as a report to the court of law if this is necessary for the cyber investigator to do so, and that is the final activity.

### 3.4 Class Diagram

A class diagram is formulated from the components mentioned above. Figure 5 depicts the class diagram of the LTI system. Some Object-Oriented design principles [8] that were considered during the modelling of this class diagram are as follows:

*"... strive for loosely coupled design between objects that interact (p. 53) …open-close principles (p. 86) … favour composition over inheritance (p. 75)"* [8].

The relationship between the subject and the observers in the observer pattern complies with the design principle for favouring composition over inheritance; while the communication between the subject and the observers is kept loosely coupled. The open-close principle is implemented by the decorator pattern through allowing the behaviour of the traffic generated to be extended without any modification to the entire code. The traffic generator and the sink server objects use the DiffServ object for communication. This reduces the number of messages sent between the objects in the system. DiffServ therefore acts as a mediator.

There are three design patterns are used in modelling the LTI architecture, namely: the Decorator, Observer and the Mediator patterns. The decorator pattern [8] is used to randomly wrap the behaviour of the traffic generated. The observer pattern [7] [8] is interchangeably used in most of the components of the LTI model, including the traffic generator, DiffServ, preservation station and sink server. The mediator pattern [9] [10] is used to coordinate the traffic generator with the preservation station or the sink server components; the following subsections explain how the different design patterns are applied and class diagrams are provided to show the relationships and interactions between the objects of the system.

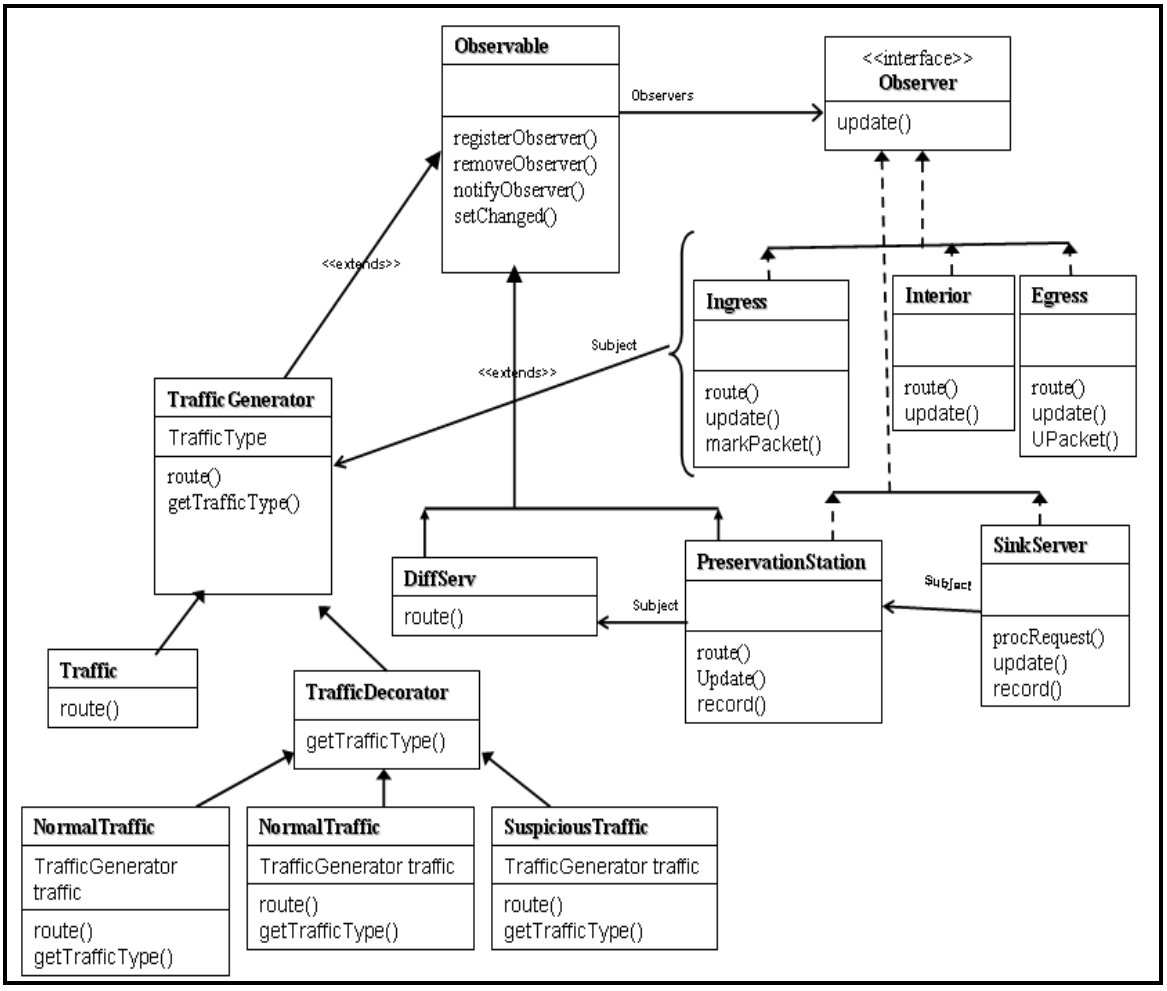


*Figure 5: The Class Diagram of the LTI model Using DiffServ*

## 4 CONCLUSION

This paper represents the LTI model using the UML design technique. From the specified requirements of the LTI system, various design diagrams are used for its representing, including use case diagram, sequence diagram and activity diagram. These diagrams clearly specify in details the role of the each component of the system and the interactions of the investigator and the systems, including the precautions that must be kept in mind when handling the piece of evidence. The use of these diagrams simplifies the made of LTI model so that its implementation can be easily achieved. The implementation of the LTI model is currently on progress, and the performance evaluation and tests will be carried out in future

## 5 REFERENCES

[1] Corey, V., Peterman, C., Shearin S., Greenberg, M.S. & Van Bokkelen, J. 2002, Network Forensics Analysis, Internet Computing, Volume 6, pp. 60- 66, IEEE.
[2] Solomon, M.G., Barrett, D. & Broom, N. 2005, The Need for Computer Forensics, in L. Newman and W.G. Kruse (Eds), Computer Forensics Jump Start, pp. 01-20, SYBEX inc.

[3] Kohn, M., Eloff J. & Olivier, M.S. 2006, Framework for a Digital Forensic Investigation, in H.S. Venter, J.H.P. Eloff, L. Labuschagne and M.M. Eloff (Eds), Proceedings of the ISSA 2006 from Insight to Foresight Conference, Sandton, South Africa (published electronically).
[4] Zantyko, K. 2007, Commentary: Defining Digital Forensics, Forensic Magazine, 20, Vicon Publishing, Feb-March 2007 issue, [Online] Available at: http://www.forensicmag.com/articles.asp? pid=130, as on 12 April 2008.
[5] Strauss, T., Olivier, M.S. & Kourie, D.G. 2006, Differentiated Services for Logical Traffic Isolation, in M.S. Olivier and S. Shenoi (Eds), Advances in Digital Forensics II, pp. 229-237, Springer.
[6] [GoF] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. 1996, Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.
[7] Shalloway, A. & Trott, J. 2001, Design Patterns Explained: A New Perspective on Object-Oriented Design, Addison-Wesley.
[8] Freeman, E. & Sierra, K. 2004, Head First Design Patterns, Volume 1, O'Reilly Media, Sebastopol (CA), USA.
[9] Bains, K. & Lau, E. 2002, Mediator Design Pattern. Available at: http://sern.ucalgary.ca/courses/SENG/443/W02/assignments/Mediator/, University of Calgary.
[10] Black, S. 2004, Mediator Design Pattern. http://stevenblack.com/PTN-Mediator.ASP. Steven Black Consulting.
[11] Schmidt, D.C. 1997, Acceptor and Connector: Design Patterns for Initializing Communication Services. The 1st European Pattern Languages of Programming Conference (Washington University technical report #WUCS-97-07).