Reputation based Trust in Service-Oriented Network Environments

by

Emmanuel Ayowole Adigun

Submitted in partial fulfilment of the requirements for the degree

Magister Scientia (Computer Science)

in the

Faculty of Engineering, Built Environment and Information Technology

at the

University of Pretoria

December 2010

Abstract

Trust plays an important role in our daily life, both implicitly and explicitly. Our decisions are based on our estimation of how trustworthy a person is or how reliable a service is. Consequently, there has been a rise in trust systems that model human trust in a virtual or computing environment. These trust systems or trust models help to bridge the gap of human feelings and intuition in an unfamiliar environment. Trust models collect information regarding the participants' activities and give a trust rating based on observed activities.

In a network environment, a plethora of network devices are in constant communication as data packets are transported from source to destination. The autonomous nature of network environments and devices make it difficult to monitor the services and devices from a central point. Security mechanisms, such as IPSec, exist in routing protocols to safeguard network packets travelling in a network, however routing devices that act as service providers are not protected by malicious attacks. For example, an attack aimed at the routing architecture of a network involves a routing device advertising itself as another routing device in order to divert network traffic away from its intended destination. This dissertation investigates trust models in network environments as a possible approach to predict and ultimately eliminate attacks on routing devices. To accomplish this, the role of routing devices as service providers and requesters must be stated explicitly. Activities on a routing device must be collected and used to determine the trust level of the routing device.

This dissertation presents the TSONE - Trust in Service-Oriented Network Environment - model. The model incorporates traditional service-oriented architecture (SOA) principles to define a service-oriented network environment. Services in this environment are then defined. Furthermore the characteristics of this environment are adapted from SOA principles. An approach is defined to collect and measure activities on routing devices. This is later used to determine the trust level of the routing device. Finally, a prototype illustrates that incorporation of trust models is a possible option in assessing availability and reliability of routing devices.

Acknowledgements

Writing a dissertation is a feat, however, the true legends are the people in the background that support the author so I'll rant about them for a moment.

- Thank you God for giving me the strength, wisdom, endurance and the patience to do this.

- I'm forever indebted to my parents, Prof Matthew Adigun and Mrs Tayo Adigun, for their love and encouragement. Thanks for setting a good example to follow and for the constant reminder that you are always there when I need anything. A word of thanks to my brother, Tola, for the odd phone calls to ask about my dissertation. It was a kick in the right direction.

- My supervisor, Prof Eloff has been a constant source of encouragement. His experience and knowledge has proven invaluable both in my dissertation and in my personal life. A special word of thanks to Prof Maritjie Eloff, Prof Kourie and Prof Olivier for asking the right questions. Their thoughts and help were highly appreciated.

- Cisco, for providing me with the routing devices used for the prototype implementation. Christo Van Schalkwyk, for assisting with configuring the routing devices.

- The Information and Computer Security Architecture (ICSA) lab colleagues, both past and present have also been a great source of inspiration. Wesley Brandy, Abiodun Modupe, Kweku “SKW” Arthur, Kamil Reddy, Maciej Rossudowski, Michael Kohn, Neil Croft, Johann Fourie and Francois Mouton. Kweku, for a constant reminder of the deadline. Maciej, for allowing me to ”look over” his shoulder. Kamil, for always being there and taking the extra time to read this work over again and thanks to Michael for the breakfasts.

- The administrative support in the ICSA lab deserve a word of thanks for the countless flight, accommodation, car rental and venue bookings. Thanks to Nicolene Landman, Tazneem Moola, Leandi Ligthelm and Sune Oosthuizen

- Telkom and the National Research Foundation (NRF), for the bursary towards my study.

- Finally, I would like to thank Isabel Claassen for the professional way she proof read this dissertation and the other conference articles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Everything else has to be based on it.
Without trust, there is no basis for partnering.
It's the bottom line...

N. Rackman, L. Friedman & R. Ruff 1996

1.1 Overview

The new century has been termed the *Internet century* and some compare the dawn of the Internet to Gutenberg's invention of the printing press in the 15th century [81]. While information dissemination across the world is made easy through the Internet, the existence of the Internet has also been credited for improving the financial trade system by providing anytime-anywhere trading. In some cases, however, the financial crisis in the latter part of 2008 has also been attributed to the Internet's overload of information [50]. Online or virtual organizations, such as eBay [8] and Amazon [1], are part of many electronic commerce (e-commerce) enterprises on the Internet. Different Internet or Web applications have emerged, such as, multimedia-sharing services

([11]), social network sites ([2, 5]) and free content service providers ([3, 7]).

The networks constituting the Internet consists of various interconnected components such as hosts, nodes, routing devices and end systems [37, 57]. The interconnection of all of these components form a worldwide network that consists of a multitude of computing devices without which the Internet is not possible. This allows Internet users to transcend geographical boundaries and communicate with friends, colleagues and business associates all over the world.

The reliability and availability of all the above-mentioned components of the Internet (such as routing devices) cannot be guaranteed, as their services can be compromised and perhaps used maliciously, resulting in the non-availability of certain vital components that constitute the Internet. Reliability and availability are only two aspects of a much more complex problem domain: the protection of information traversing the Internet and the protection of the components constituting the Internet.

Currently, most protection mechanisms on the Internet focus on the top (user application) layer, that is, the application layer. On the lower levels (and more specifically on the network routing level) there is a shortage of protection mechanisms. Most protection mechanisms on the network routing level are based on encryption [20, 51, 86]. Thus, routing devices are vulnerable to several security threats such as routing table poisoning [28] and traffic redirection [75]. These threats are explained in more detail below.

- Routing table poisoning is characterised by the malicious modification or poisoning of routing tables. Routing tables contain all the information required for forwarding messages, that is, establishing communication links with other components of the Internet.

- While traffic redirection is as a result of an attacker overwhelming a victim router with traffic from neighbouring routers, this is also known as a denial-of-service attack on the victim router and results in the inability of a victim router to provide routing services as it should. An example of this type of attack is evidenced in an incident that diverted traffic away from YouTube's [11] network [72, 27, 67].

"Routers" and "Routing devices" will be used interchangeably in this dissertation. To maintain consistent access to the Internet via routing devices, trustworthiness in terms of the availability of routing devices is essential. The availability of these devices is important for information accessibility, service provision and prevention of a distributed denial of services (DDoS) attack. Such an attack has the potential of completely disabling the whole Internet. Thus, the trustworthiness of a routing device indicates its reliability and availability in a network environment. The knowledge of a router's activity, and its availability, could indicate how trustworthy it is and can also affect how a network administrator configures a routing device as a component in the wider Internet context. *Trustworthiness* is a quality or characteristic of an entity that is worthy of *trust*, worthy of confidence and reliability. The underlying concept of trust, regardless of its different application environments, is the complete confidence and reliance on an entity. Trust is discussed in the following paragraph.

Trust is a widely discussed concept especially in the social sciences. Researchers such as Golembiewski [40], Kramer [56], Gambetta [39] and McKnight [66] are noteworthy references. Trust is seen as a catalyst for cooperative endeavours [39]. Gambetta [39] posits, with the preceding statement, that trust is not a precondition for cooperation but that trust develops as a result of cooperation. For example, two individuals cannot trust each other until they interact the end result of their interaction either brings about trust or disappointment (distrust).

In recent years the concept of trust has opened up a myriad of possibilities where its application is essential. These include but are not limited to service provision (e.g. online banking, electronic ticket purchasing), file sharing (peer-to-peer communication), virtual markets (eBay, Amazon), mobile commerce, online bartering and social networking websites. For example, in a peer-to-peer file-sharing environment, trust is a necessity to identify peers that provide poor quality services [89]. In a peer-to-peer environment trust is an essential tool to monitor badly behaved peers. It is the concept of identifying and monitoring badly behaved peers that led the author of this research project to further investigate how trust can be employed to improve the trustworthiness of routing devices.

Virtual auction sites such as eBay [8], use reputation systems to establish trust among their users. eBay collects buyers' feedback in its reputation system after a transaction with corresponding sellers. A reputation score is assigned based on the type of feedback given by buyers, and this score can be seen on the seller's profile for future transactions by prospective buyers. eBay's feedback system indicate a move away from face-to-face interactions to interactions in a virtual environment. New areas of research focus on investigating how trust can be represented in different environments and the different type(s) of trust best suited to an environment. Since trust is a social concept, interdisciplinary efforts such as linking trust from the social sciences to an online electronic commerce environment are becoming prevalent.

So far this section has examined the pervasiveness of the Internet and its need for trust in its components. A brief introduction of the concept of trust has also been given above. Ultimately, the focus of this study is to investigate trust on the network layer. The network layer in the context of this research project is similar to the concept as described in the documentation of the International Standards Organization Open Systems Interconnection (ISO/OSI) reference model. This model consists of seven layers that specify how network communication occurs in ISO/OSI compliant networks. Each layer performs functions on the messages that are transmitted from a sending host to another receiving host in a network environment. These functions include: establishing a connection with a receiving host, error detection, flow control, routing and arranging messages in a readable format when they arrive at the receiving host. Of interest to this study is the network layer where routing and message segmentation of packets are performed. Routing devices provide routing services between different hosts and/or routing devices in this layer. There are various points of attack in a network environment, however, this study focuses on the security aspects of routing devices. This study places emphasis on the availability of these devices and trust in these devices as service providers.

1.2 Problem Statement

The main problem or research question addressed by this research is: **how does one assess the trustworthiness of a component, such as a routing device, of the network layer?** The scope of this research is limited to the following sub-questions that arise from the main question:

What services are provided by routing devices?

Routing devices are the main components of a network environment, without which communication among different hosts or nodes is impossible. Different services are provided by routing devices based on the routing protocols implemented on the routing devices. It is essential to determine the functionality and architecture of routing devices to know what their capabilities are.

What are the trust/security requirements for routing devices in a network environment?

The current trends in attack technology have indicated that attackers tend to use distributed denial of service (DDoS) attacks on Internet infrastructure [44]. These types of attacks are aimed at routing devices and are carried out by attackers from hosts or routing devices on the same network. Attackers exploit different vulnerabilities on routing devices. Therefore security requirements for routing devices are investigated for the purpose of managing and trusting a routing device.

How can trust be represented in a network environment?

The multi-disciplinary nature of trust has attracted different opinions about trust. Trust is a dynamic and subjective concept that allows for various interpretations. The context within which trust is applied also plays an important role in its definition. Trust is modelled differently in a variety of environments. Thus, to model trust, it is important to consider the various views and definitions of trust in different contexts. This assists in answering the main research question. It is also essential to distinguish between a service-oriented network environment and aspects of network dependability relating to quality of service. The purpose of a service-oriented environment and why such an environment is needed should be stated explicitly.

Which trust model is possibly suitable for a practical implementation on routing devices in a network environment?

Several trust models have been proposed and implemented by various authors. The subjective concepts of trust as understood by the implementers of trust models influence the implementation of the resulting trust models. Thus, a survey of related trust models is carried out and a motivation is provided for the best suited trust model identified for routing devices in a network environment. The choice of trust model for a network environment must meet security/trust requirements as identified in the first research question above.

1.3 Terminology Used

To avoid misunderstandings, the terms used in this study are elaborated on below. These terms include *reputation*, *service-oriented environments*, *trust and security*, *routers*, *network layer of the OSI model.*

1.3.1 Reputation

Reputation can be considered as a collective measure of trustworthiness based on referrals or ratings from members of a community [49]. Word of mouth is one of many sources of reputation and the “most ancient mechanism in the history of human society” [35]. It can result in a community member recommending or discouraging the purchase of a product or use of a service. This is also known as feedback.

1.3.2 Service-Oriented Environments

The dawn of electronic commerce (e-commerce), electronic government (e-government), electronic health (e-health) services have been attributed to the advent of the Web. These Web-based services have given rise to a new business environment known as a *service-oriented environment* (SOE) which are instances of service-oriented architectures. A SOE is characterised as an open, collaborative, dynamic and distributed environment that is able to respond in a timely manner to consumer needs and business dynamics [29, 30]. To carry out business activities and complete various transactions, entities in this environment need to communicate with one another to publish their services, request a service and provide services to other entities as needed.

1.3.3 Trust and Security

There are several definitions of trust and security which are alluded to in subsequent chapters. However, for the purposes of this study and for simplicity's sake *trust* is defined as the characteristic of an entity that allows it to be dependable so that service requesters can rely on its serviceability. *Security* focuses on protecting an entity from attacks, intrusions and vulnerabilities.

1.3.4 Routers

Routers are network devices that are mainly responsible for routing and forwarding data packets across local area or wide area networks. A router is a hardware device but can also be a software application running on a host. It can be used to connect a local area network (LAN) to a wide area network (WAN) or vice versa. Routing protocols and algorithms on routers control the route traversed by a packet to get to a destination. The destination may be another router or a host on another network. The router is the primary network *routing* device that is of relevance to this study.

1.3.5 Network Layer of the OSI model

The Open Systems Interconnection (OSI) model provides a layered framework for the design of network systems that allow communication between all types of computer systems [37]. One of the layers in this framework is the network layer which is basically responsible for source-to-destination delivery of data packets across multiple networks [37]. The network layer is needed where two hosts are connected to different networks and data packets need to traverse the networks via an interconnecting network device. The network layer provides the routing functionality between networks.

1.4 Methodology

In approaching this study, a literature study of the main concepts was done, followed by the design of a prototype and implementation of the prototype as a proof of concept. The literature survey involved exploring trust and the various trust models in detail. A detailed examination of network environments, routing protocols and various routing devices was carried out.

A suitable trust model is chosen which led to the design of a customized trust model and the network environment topology. The design was implemented practically to prove the concept and tested to validate the design.

1.5 Dissertation outline

This dissertation is laid out as follows:

Chapter 1: Introduction

The current chapter includes an introduction of the concepts discussed in this dissertation, the research questions related to this study, a definition of some terms used in this dissertation and the presentation of the research methodology.

Chapter 2: SONE - Overview of Routers and Network Layer components

Chapter 2 describes a service-oriented network environment (SONE). The chapter goes on to explain the layers of the ISO/OSI model with emphasis on the network layer and its components. The architectural properties of routers are also provided including security vulnerabilities in a network environment and attempts to combat these vulnerabilities on the routing protocol level.

Chapter 3: Trust and Reputation Systems

This chapter provides a background on trust by providing different definitions by various authors and deriving a definition for this project. Trust models and reputation systems in research literature are discussed and evaluated. This chapter expands on and gives reasons for the choice of reputation model adopted for this study.

Chapter 4: Requirements of a TSONE

Chapter 4 describes how trust can be represented in a trusted SONE (TSONE). The need for security on the network layer is explained and the requirements of a TSONE environment are determined.

Chapter 5: Design of a TSONE

Chapter 5 gives a brief overview of the design of the TSONE framework as proposed by the current research. A diagrammatic representation of the different parts of the framework is provided.

Chapter 6: Detailed Design of TSONE

This chapter elaborates more on the design of TSONE. The components in the component class diagram are explained in detail.

Chapter 7: Computing and Updating of Trust Levels

Chapter 7 gives an explanation of trust levels as used in a TSONE. A statistical function is provided to calculate and update trust levels. The impact of the trust level on the routing device is also discussed.

Chapter 8: Prototype Implementation

This chapter describes the implementation and operation of the prototype in detail.

Chapter 9: Conclusion and Future Work

This chapter concludes the dissertation and future work in this area of study is proposed.

Chapter 2

Service-Oriented Network Environment - The Network Layer and Routers

> *When computers are networked, their power multiplies geometrically. Not only can people share all that information inside their machines, but they can reach out and instantly tap the power of other machines, essentially making the entire network their computer*
>
> – *Scott McNeely*

2.1 Introduction

In a network environment, routing devices (representing service providers), announce or advertise their routing services to neighbouring routing devices that require routing services. This type of service-oriented environment (later referred to as a service-oriented network environment (SONE)) is based on the concept of service-oriented architecture (SOA). SOA is well discussed in current research literature and serves as a basis for the discussion of SONE in this chapter.

Chapter 2 aims to answer the first two research questions posed in the previous chapter namely: *What services are provided by routing devices? What are the trust/security requirements for routing devices in a network environment?*

The chapter is structured as follows: Section 2.2 includes an overview of the service-oriented architecture followed by the definition and discussion of a service-oriented network environment in Section 2.3. The layers of the ISO/OSI model are discussed in Section 2.4 with special emphasis on the network layer and services provided in that layer. A routing device is also discussed here. Section 2.5 contains an overview of the components of a routing device that provides the reader with a better understanding of the functionality of a routing device. Security requirements in network environments are discussed in Section 2.6, followed by an overview of routing protocols and a detailed discussion of security attacks in a network environment.

2.2 Service-Oriented Architecture

Service-oriented architecture (SOA) is discussed in this section as a precursor for the service-oriented network environment (SONE).

2.2.1 Definition

There are different definitions of a SOA in the literature and these definitions are discussed here in order to arrive at a suitable definition for this study. Organisations and societies that have defined SOA includes amongst others, the World Wide Web Consortium (W3C), the component based development and integration forum (CBDI) and the organisation for the advancement of structured information standards (OASIS).

W3C architecture group defines a SOA as "a form of distributed systems architecture." [93]. The W3C also provide a model of a SOA and properties of the model that focus on its implementation. However, this definition has been considered a technical definition [83] that does not reflect the business-IT alignment of a SOA [59]. Krafzig et al. [55] also defines a SOA as "a software architecture that is based on the concepts of an application front end, service, service repository and service bus". Krafzig et al. emphasise that a service's interface is an important part of a SOA. The service interface specifies how to access the functionality of a service.

Another definition from CBDI describes a SOA is:

> The policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer. Services can be invoked, published and discovered through a service registry and are abstracted away from the implementation using a single, standards-based form of interface [83].

This definition provides a broad overview of a SOA in terms of service interoperability and independent implementation. However, interoperability depends on policies that are defined to enable service/application functionality.

OASIS define a SOA as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" [73]. The capabilities referred to above are services under the control of different service providers. Thus, a service provider has a service that can be utilised by a service requester to complete a task.

According to the above definitions, a SOA consists of the following (Figure 2.1):

a) *Service providers* that own services and have capacity to provide services as required.

b) *Service requesters* that are in need of services and can discover services that match these needs.

c) A *Service registry* where service providers can publish their services and service requesters can find a suitable service.

The perceived value of a SOA is that it provides a framework for matching needs and services and for combining services to address those needs [73].

The author's definition of SOA is a summary of the above definitions and reads as follows: *a service-oriented architecture is a framework that allows services to be published and discovered through a service registry by using a standard protocol.* A SOA provides a platform for service providers and requesters to interact despite the fact that they might never have communicated with each other previously. The protocols used for communication are standardised and can be used by various (service) entities.

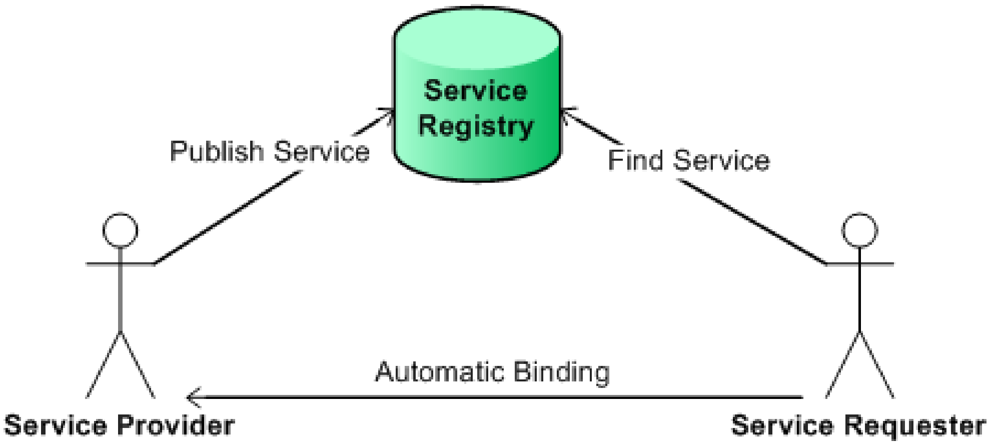


Figure 2.1: Service-Oriented Architecture

2.2.2 Instances of SOA

The terms 'SOA' and 'Web services' are often used interchangeably ([14, 83]). However, Web services are instances or implementations of SOA [14]. Another instance of a SOA includes, among others, service-oriented computing (SOC). Web services and SOC are also known as service-oriented environments (SOE) [29].

Service-oriented computing is the computing paradigm that utilises services as fundamental elements for developing applications and domain [74]. Services perform functions that allow organisations, via their information systems, to expose their core competencies programmatically over the Internet using standard (XML-based) languages and protocols [74, 32]. Organisations can use SOC's interoperability property to define and execute business processes. Integration between different information systems is possible to enable cooperation between business partners. Thus SOC reinvents the way organisations work together, for instance, common tasks in a business process can be easily outsourced to external service providers for performance and cost reasons [24]. SOC intends to make a collection of software services accessible via standardised protocols [24]. Since SOC is based on Web services, the latter has already defined the standard language and protocols used in SOC.

Web services is defined as a middleware technology that offers standard communication interfaces to foster ease of communication between heterogeneous applications over the distributed network environment [29]. Web services like SOC provide interapplication communication using XML-based messages via Internet-based protocols or standards. These standards include, extensible mark-up language (XML), simple object access protocol (SOAP), Web services description language (WSDL) and universal description, discovery and integration (UDDI). Service providers publish their available services to the service registry (Figure 2.1) using the *WSDL*. Service requesters looking for available services use the *UDDI* protocol to find the service that they need and service requesters bind themselves to an appropriate service provider by using the SOAP protocol. Messages between the three interacting roles are sent via the hypertext transfer protocol (HTTP) encoded into *XML* format so that messages are understood by all agents in the environment. The architectural benefits of Web services, which are similar to SOA benefits, include loose coupling, platform independence, self description, and discovery [74, 83, 93, 46, 59].

2.2.3 Characteristics of a SOA

The characteristics of a service-oriented architecture are given below. These characteristics are later adapted as SONE characteristics.

- *Loose coupling*: This indicates that a service is not permanently attached to the service requester. Services can be invoked by all service requesters. Services are logically decoupled from service requesters and can be reused. However, service requesters are coupled with a service as they know what the services are and what they can accomplish [74].

- *Implementation neutrality*: The implementation of an interface is not programming language dependent. This gives the programmer freedom to implement a service in a programming language of his/her choice. Each implementation must be unique and the implementation details of the service should not be visible or discourage service requesters. [83, 93]

- *Flexible configurability*: Service-oriented systems must be able to adapt to their environment as needed. These systems are subject to change because of the dynamic environment they are part of thus different components must bind to each other as quickly as possible without loss of correctness. [93]

- *Persistence*: Services do not have a long lifespan but because of the dynamic and heterogeneous nature of this environment services must exist long enough to detect an exception. Correction action to handle exceptions must be specified and action taken by others must be monitored for future reference. Services should exist long enough to engender *trust* in their behaviour because they are engaged dynamically and *reputation* might be the only means available to gauge their reliability [46].

- *Granularity*: Interactions between participants in this environment must be modelled at high-level granularity. Coarse granularity reduces dependencies among participants and reduces communications to fewer messages of greater significance [46, 59].

- *Teams*: Agents or participants in this environment must be grouped together in teams rather than in a central structure. Participants grouped together in different teams can focus on providing different services: that is, a team can focus on providing a particular service and another team can provide another service. [46]

A SOA aims to provide a framework and a set of policies and practices to ensure adequate delivery of services from service providers and consumption of services by service requesters. The SOA framework is deployed within a distributed systems environment that is characterised by heterogeneous and autonomous services including service providers and requesters. Although trust and security constitute one of the non-functional requirements of a SOA [14] a SOA does not address these requirements. Therefore, an additional conceptual framework and architectural elements are required [73]. The current research project's trusted service-oriented network environment (TSONE) places emphasis on trust for a network environment as a type of service-oriented environment. TSONE is explored in subsequent chapters. The following section elaborates on a service-oriented network environment.

2.3 SONE: Service-Oriented Network Environment

Thus far, Web services, service-oriented computing and their overarching service-oriented architecture have been examined. These environments consist of host or autonomous agents that provide various services. In this section, the idea of a service-oriented environment is extended to a network environment which focuses on network devices, that is, routers as *service providers* and *service requesters*. Each routing device has a routing table that specifies the router to which data packets can be forwarded. Thus a router also has a *service registry* that consists of information collected from other network devices in the environment. The following subsection provides a definition of a service-oriented network environment (SONE).

2.3.1 Definition of SONE

Based on the definition of a SOA provided above (Section 2.2.1), a SONE is defined as *a collaborative environment where network devices utilise their resources to publish and discover services available in a network environment* [15]. The service context of a SONE is a network environment. A SONE is a type of service-oriented environment that embraces the characteristics of a service-oriented architecture.

- *Loose coupling*: Packet routing and forwarding are some of the services provided by a router. These services are not permanently attached to one specific router but available to all routers in a network environment.

- *Implementation neutrality*: Operating systems for routers are vendor-specific and they don't affect the routing service provided by routers.

- *Flexible configurability*: Routers in a network environment adapt to the environment by building up a routing table based on the environment configuration.

- *Persistence*: Routers and their services have a long lifespan as long as they are active in the network environment and are available for other routers to route packets via them.

- *Granularity*: Interaction between routers is managed by the Internet control message protocol (ICMP). The details of messages passed between routers are encapsulated in IP datagrams.

- *Teams*: On a logical level, routers are grouped together in an autonomous system managed by an administrator for a particular domain. Also, routers can be divided into sub-networks.

The *interoperability* of routers in a SONE is important because routers depend on each other to provide routing services. A SONE provides an environment where routers can interact regardless of their underlying architecture [15]. A typical network environment is similar to a SONE, that is, there is no substantial difference. However, the author defines a SONE to specify the focus of this research project. Focus is placed on network devices, such as routers that provide services in a network environment. For this research project, routers are the primary network device that is discussed for a SONE. Routers exist on the network layer of the International Standards Organisation's Open System Interconnection (ISO/OSI) model. The following section provides a brief overview of the model with emphasis on the network layer.

2.4 ISO/OSI Model

The International Standards Organization (ISO) deals with various international standards. An ISO standard that addresses computer network communication is the Open System Interconnection (OSI) model. An open system is a set of protocols that allow any two different systems to communicate regardless of their underlying architecture [37] - this allows for fast and uninhibited communication between systems.

At this stage the author assumes that the reader has knowledge of the ISO/OSI model and no further information is provided about it. (If further information is needed please consult the following references: [48, 37, 57].)

The author depicts the different layers of the OSI model in Figure 2.2. The diagram indicates the relevant network device for each layer and for a number of layers, the layers are numbered from bottom to top. Communication between a sender and receiver begins at the top of the layer for the sender and at the bottom of the layer for the receiver.

The different layers of the OSI model and their corresponding network devices are briefly explained, followed by the functions performed at each layer. The gateway network device is introduced here because it operates on all seven layers of the OSI model. It is a protocol converter that accepts packets formatted for one protocol (for example, AppleTalk) and converts them to a packet formatted for another protocol (for example, TCP/IP) before forwarding the packet towards its destination.

1. Layer 1 - The physical layer coordinates the functions required to transmit a bit stream over a physical medium [37]. This involves moving each bit within the frame from one node to another. A repeater functions at this layer: this device simply recreates an incoming bit by boosting its transmission energy and transmitting the bit to its outgoing interfaces.

2. Layer 2 - The data link layer transforms the raw message received from

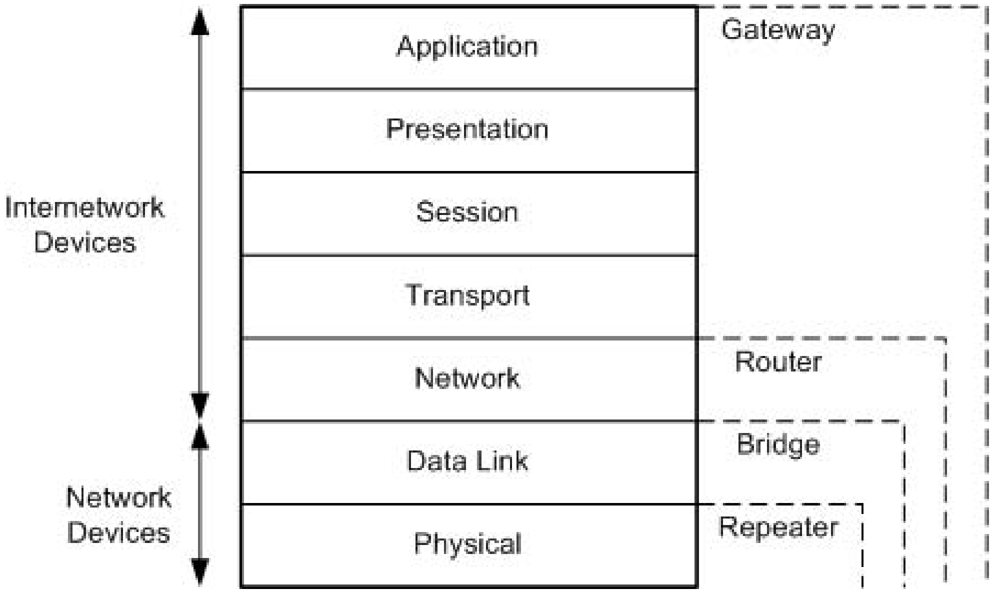


Figure 2.2: OSI model with network devices at different layers

the physical layer to an error-free message. The relevant network device in this layer is the bridge and this device operates on data packets for forwarding and filtering.

3. Layer 3 - The network layer is responsible for routing packets from one host to another [57]. This layer ensures that packets move across their required routes from their origin to their destination. This layer also provides routing across multiple networks. The *router* operates in this layer as well as in the preceding layers (physical and data link layer). A router obtains the network address of the destination host from the packet and uses the routing protocol to determine the best possible route for transmission. The packet is then passed on to the next router on that route.

4. Layer 4 - The transport layer provides a source-to-destination routing service for an application layer message. This differs from the network

layer which is responsible for one packet only.

5. Layer 5 - The session layer controls and maintains communication between two communication systems. The session layer keeps the communication session alive in a situation where there might be silence due to bulk data transfer.

6. Layer 6 - The syntax and semantics of exchanged information are monitored at the presentation layer. This is especially necessary when different encoding or encryption systems are used.

7. Layer 7 - The application layer provides a suitable user interface for interpretation of what is contained in the transmitted message. Support is provided for services such as electronic mail and remote file access.

The seven layers can be further divided into three subgroups [37]. The *network support layers* (Layers 1, 2 & 3) deal with the physical aspects of moving data from one device to another, for example, physical addressing and reliability. Layers 5, 6 and 7 are the *user support layers* – they allow interoperability among unrelated software systems. Layer 4 links the two subgroups and ensure that lower layers data are in a form that the upper layers can use. The upper layers are always implemented in software while the lower layers can be implemented in both hardware and software (with the exception of the physical layer that is always hardware bound).

The network layer of the OSI model is important in a SONE because of the routing and forwarding services carried out on this layer. The router is the primary network device responsible for routing data packets to their destination and it exists on the network layer. The network layer provides a basis for a SONE because of the routing and forwarding services and the service provider routers. The next subsection elaborates on the functions of the network layer in relation to a SONE.

2.4.1 Network Layer Functions

The network layer basically transports data packets from one host to another. Other functions performed on the network layer include the following [37, 57]:

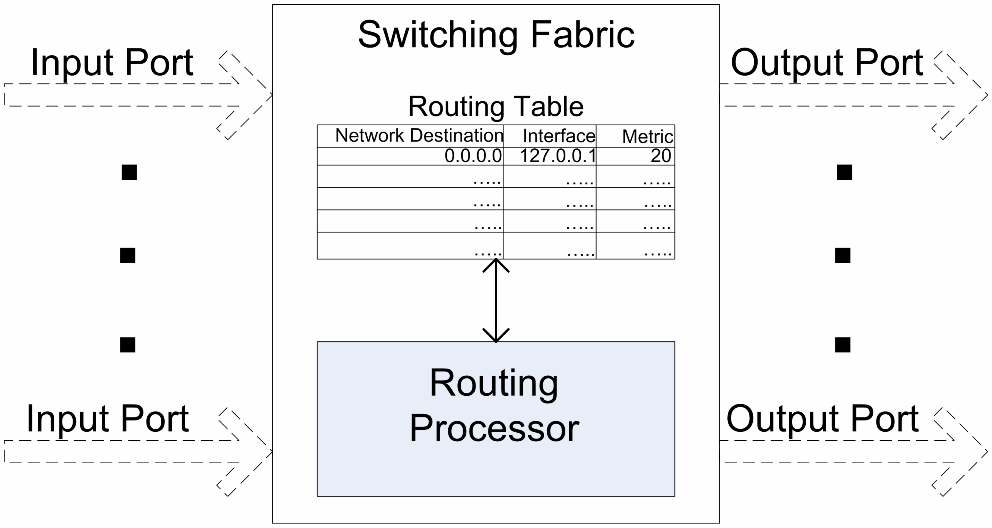
- **Error control** makes the logical channel between two hosts more reliable. To maintain a consistent service in a SONE, error control is crucial for communication between two hosts. In a situation where packets are dropped due to, for example, due to overloading the network layer has protocols that inform the sender to resend the missing packets.

- **Path determination** is done on the network layer with the help of routing algorithms. To provide a routing and forwarding service in a SONE, the network layer determines the paths to a network destination. This information is contained in the routing table which resides on the router.

- **Switching** involves moving data packets as they arrive on the router's input to the appropriate output port of the router. Switching is a type of path determination that occurs in the router. The routing table on the router contains information for the router to perform switching. One of the resources referred to in the SONE definition above is the routing table used by the router to route data packets.

2.5 Architecture of a Router

In the previous section a SONE was defined and an overview was provided of the network layer and the router. In this section, the architecture of a router is explored. A router consists of: input ports, the switching fabric, the routing processor and the output ports. The author depicts a router's architecture in Figure 2.3.



Network Destination	Interface	Metric
0.0.0.0	127.0.0.1	20
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....

Figure 2.3: Router Architecture

An *input port* on a router performs the physical layer functionality of accepting and terminating incoming connections to the router. The *switching fabric* is used to determine the output port of incoming packets. The choice of the output port is made using the routing tables computed by the *routing processor*. A copy of the routing table is stored and updated at the input port in most routers with unlimited processing capabilities. The *output port* of the router receives packets from the switching fabric and stores them on the output port's memory before transmitting them.

In a SONE, the input and output port of routers are connected to different networks. Thus, the router's switching fabric routes packets from the input ports to their respective network destination via the output ports. Routers are aware of other routers and networks in their vicinity via their routing table. Routing tables are updated according to new information received from neighbouring routers. New information received by routers could be as a result of new information about the network environment acquired by a router that has been propagated to other routers via the Internet Control Message Protocol (ICMP). Routing tables and routing protocols are briefly discussed in the following subsections.

2.5.1 The Routing Table

Routing tables contain the destination network address and the next router or next hop towards the destination. Routes to a network destination are contained in a routing table. Routes are discovered via sources such as directly connected networks, static routes that are manually configured by the network administrator and through routing protocols implemented on the router. An example of a routing table on a Cisco router is given below.

```
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     20.0.0.0/24 is subnetted, 1 subnets
```

```
O        20.20.20.0 [110/1010] via 30.30.30.1, 01:22:00, Serial0
     40.0.0.0/24 is subnetted, 1 subnets
O        40.40.40.0 [110/1010] via 30.30.30.1, 01:22:00, Serial0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        30.30.30.0/24 is directly connected, Serial0
C        30.30.30.1/32 is directly connected, Serial0
```

The first few lines of the routing table output provide abbreviations and their corresponding descriptions. The routing processor discovered three networks: 20.0.0.0/24, 40.0.0.0/24 and 30.0.0.0/8. These networks were discovered through the OSPF (O) routing protocol and directly connected networks (C).

The square bracket next to the network that was discovered contains a default administrative distance and route metric [distance/metric]. Sources of routing information received have an attached measure of trustworthiness called the administrative distance. The lower the administrative distance, the more trustworthy the source [62]. That implies that an unknown route source will have a large administrative distance.

A couple of routing protocols, such as, open shortest path first (OSPF) and enhanced interior gateway routing protocol (EIGRP) allow the network administrator to assign a cost or a metric to a route. The *metric* assigned to a route is usually based on the type of communication media on the router's interface. However, a network administrator can also change the cost attached to an interface on a particular route due to delay, throughput or type of service (TOS). Different metrics are normally used for each TOS, such as, bulk data transfer or video data.

2.5.2 Routing Protocols

Routing consists of routing algorithms (static) and routing protocols (dynamic). At the heart of a routing protocol is a routing algorithm that finds the best path from a source to another destination based on least cost and minimal delay. Routing protocols are used to determine path from sender to destination. Path determination is done based on two routing methods, namely inter-domain routing and intra-domain routing. *Inter-domain routing* (also known as multiple autonomous system (AS)) is employed when routing has to be done across administrative domain boundaries whereas *intra-domain routing* (also known as single autonomous system) is used when routing occurs within a single network under one administrative authority. This authority owns the router in its domain but not necessarily all the links that connect the routers in the intra-domain. The most commonly used routing protocols within an autonomous system are routing information protocol (RIP), open shortest path first (OSPF) and enhanced interior gateway routing protocol (EIGRP). OSPF is used as a routing protocol in this study for the development of the proof-of-concept hence a brief discussion of OSPF is provided next.

2.5.2.1 OSPF

OSPF allows for information dissemination [37] about other autonomous systems (ASs) in the environment. This is done by using the link state routing protocol which is a means of sharing information among the routers in an area. OSPF divides an AS into areas and routers in that area maintain an identical database describing the autonomous system's topology [69]. For example, a router in an autonomous system that is using the OSPF routing protocol will know what type of topology is employed in its immediate and surrounding network area. Thus network and link state information is sent, not only to immediate network areas but also to other networks that have reach ability to the *area border router* (ABR). This type of router summarises information about an area and sends it to other areas by using inter-AS routing. However, the details of an area remain invisible to all routers outside the area.

2.5.2.2 Border Gateway Protocol (BGP)

Inter-domain routing employs BGP to route packets across networks and administrative boundaries. This protocol distributes path information about each autonomous system from sender to receiver. Unlike intra-domain routing, BGP does not manage the internal details of a network such as cost of routing and selection of routing path. BGP provides a mechanism to distribute path information among interconnected autonomous systems but leaves the policy for making the actual route selection up to the network administrator [57]. Therefore it is possible for a network administrator to implement a policy that routes traffic from an organisation's network through another network.

The network or route information acquired by routers in a SONE is important for forwarding incoming data packets. If incorrect information about a route is propagated to other routers data packets could end up at the wrong destination. This is evidenced in an incident that occurred in Pakistan where Internet requests and data packets intended for YouTube [11] were diverted to Pakistan [27, 67, 72]. This example is one among many security threats in a SONE that are examined in the following section.

2.6 Security threats in Network Environments

The need to secure network environments and network infrastructures is becoming increasingly important. The increase in cyber terrorism and various network attacks on network infrastructure, such as routers, has made it important to focus both on physical security for a networking device and logical security, for example, software security. In a SONE, network attacks are primarily focussed on the software level, in other words the routing protocol and the router's operating system. As early as 1989 Bellovin [23] published that the abuse of the routing mechanism and protocols is probably the simplest protocol-based attack available. There are a variety of ways to do this, depending on the exact routing protocols used.

More than a decade later in 2001, the Computer Emergency Response Team (CERT) [44] presented a technical report on denial of service attacks and pointed out the selective targeting of routers. One of the most recent and disturbing trends is an increase in intruder compromise and use of routers. Intruders using vendor-supplied default passwords deploy routers to gain unauthorised access to and control of other routers. Routers are being used by intruders as platforms for scanning activity, proxy points for obfuscating connections to various networks and as launch points for packet flooding DoS attacks. Routers make attractive targets for intruders because they are generally more part of the network infrastructure than computer systems. Of extreme concern is the potential of routers to be used for DoS attacks, directed against the routing protocols that interconnect the networks in the Internet [44].

One of the main problems with the networks in Internet is that security was not built into the underlying protocol suite, that is, the TCP/IP protocol. There are a number of serious security flaws inherent in the protocols–in particular routing protocols [23]. The following subsection elaborates on the type of security attacks prevalent in a network environment.

2.6.1 Types of attacks in a network environment

Attacks in a network environment often exploit routing protocols that are being used on the network. Routing protocols are software functions on routers and therefore an attack on the routing protocol is also an attack on the router. The functions of a router include: forwarding packets and using routing protocols to build up routing tables [45, 57]. A routing attack can cause considerable damage to the domain when an attack occurs. Huang et al. [45] and Chakrabarti and Manimaran [28] define the taxonomy of network routing attacks as follows:

DNS hacking attacks [28]: Domain name system (DNS) is a distributed hierarchical global directory that translates machine/domain names in to numeric IP address. DNS can map human memorable names to numerical addresses. DNS hacking is typified when an attacker takes over a victim's domain name without his/her consent, and is also known as DNS hijacking. This is due to the lack of authentication and integrity of data held within DNS as well as the protocols that use host names as an access control mechanism. An example of DNS hacking is *spoofing*, where an attacker masquerades as a DNS server and feeds the host the wrong information.

Routing table poisoning [28]: Routing tables are used by routers to exchange routing information and/or updates between routers. Poisoning of routing tables is achieved by the malicious modification of routing information in routing tables. This can result in incorrect entries in the routing table and could lead to congestion and an overwhelmed host which will probably take the host out of service.

Packet mistreatment [28]: This attack happens while a packet is in transit. In this type of attack the malicious router manipulates packets by adjusting their destination address resulting in congestion or denial of service. The problem becomes intractable if the packets start triangle routing, that is, when packets are routed in a loop formation around the network.

Acquiring routing information [45]: This happens when an intruder monitors and/or records routing exchanges between authorised routers to sniff for routing information. The intruder can also analyse the traffic to determine the network topology and determine the bandwidth allocation for an interface. This is not harmful to the user or the network until the intruder uses the information that has been gathered against the network user or the network infrastructure.

Denial of Service[28]: In these sort of attacks, the packets are routed correctly but the destination becomes a target of the attackers. A Denial of Service attack is usually directed at a specific host with the aim of putting it out of service. This attack may be carried out by individuals or groups who may use such attacks for personal gain. DoS can become extremely dangerous and hard to prevent if a group of attackers coordinate their efforts. DoS attacks are categorised into two types: ordinary and distributed attacks. In ordinary DoS attacks an attacker uses a tool to send packets to overwhelm the target system forcing a reboot and in the process the attacker spoofs the source address. In a distributed DoS (DDoS) attack, the attacker makes use of multiple attack servers also known as agents to coordinate the attack against a single host.

BGP is the standard inter-domain routing protocol on the Internet. However, BGP is susceptible to *all* of the attacks described above and more, such as: misdelivery or non-delivery of user traffic, fabricated BGP message from a fictitious BGP speaker, network congestion, packet delays and violation of local routing policies. Research efforts to address threats and vulnerabilities in BGP include secure-BGP (S-BGP) [53, 52], secure origin BGP (soBGP) [90] and pretty secure BGP (psBGP) [88]. These efforts make use of authentication and validation methods to verify BGP speakers for an autonomous system (AS) and for authenticating autonomous system numbers. They also verify IP prefix origination to determine that an IP prefix owner is the true owner of the prefix. psBGP uses centralised and decentralised trust models to authenticate and verify various information in BGP.

2.6.2 Trust and Security Requirements for a SONE

To conclude this chapter, trust and security requirements for a SONE are specified. These requirements are based on the security threats and routing protocols examined previously in this chapter.

- A SONE requires an application that can evaluate the trustworthiness of routers. Trustworthiness should be based on reliability and availability of routers. Events on routers in a SONE could be collected and analysed to determine a router's trustworthiness.

- Although routing protocols are at the heart of a SONE they are vulnerable to security attacks. The original design of routing protocols does not include security functionality hence routing protocols require additional security applications. Since routing protocols are hard coded into routers' operating system they cannot be manipulated to add extra functionality. As discussed above, research efforts exist to implement additional security into routing protocols but they are not yet available on routers' operating system.

- Routing tables are necessary to determine the forwarding path of a data packet in a network environment. As depicted above routes to different networks are discovered through various methods such as directly connected networks, static routes and routing protocols implemented on the router. A route metric or cost is also attached to routes discovered via routing protocols so as to determine if traffic is routed to a network or not, that is, if the router is used or not. The integrity of routers and their routing tables must be maintained to accurately determine the forwarding router on the path.

2.7 Conclusion

This chapter reviewed the network layer of the ISO/OSI model and the router as a network device on this layer. This analysis has been done to identify services provided by network devices, such as routers. Types of attacks targeted at routers were discussed and from this discussion the author concluded that trust and security requirements for a SONE are needed. The requirements for trust and security in SONE and their implementation are explained in Chapter 4. The following chapter 3 investigates trust and reputation systems in literature.

Chapter 3

Trust and Reputation Systems

...on what do you rest this trust of yours?

Isaiah 36:4

He who stands by what he has allowed to be known about himself, whether consciously or unconsciously, is worthy of trust.

– Niklas Luhmann [60]

Without the meditative background that is criticism, works become isolated gestures, historical accidents, soon forgotten.

Milan Kundera

3.1 Introduction

Trust is a social concept that can be integrated into any social context where interaction occurs between different parties and a probability of misbehaviour exists. Trust plays an important role in any context where risk of any kind might be involved. For instance, two routing devices in a SONE, have no knowledge of each other but need to request services. There is always a risk that the service provider may neglect the service agreement and provide a substandard service or no service at all. Trust models, also known as reputation systems, keep track of interacting parties' behaviour in order to evaluate their trustworthiness.

The remainder of this chapter attempts to answer the last research question: *Which trust model is possibly suited for practical implementation on routing devices in a network environment?* This research question allows for an in-depth examination of trust in different contexts. The service provision context in a network environment enables a classification of trust into a virtual environment that differs from a human trust environment.

The chapter comprises of two main sections. Section 3.2 provides different definitions of trust and the section concludes with an appropriate definition of trust that is suited for a network environment with reference to this research project. Section 3.3 investigates different trust models available in literature and provides reasons for the choice of model chosen for this project. Section 3.4 distinguishes between trust as it applies to this research project and trust attributes in routing dependability.

3.2 Trust

Different views on and definitions of trust exist, due to its interdisciplinary attributes. Trust has found its way into economics, sociology, psychology, business, law and computing. However McKnight and Chervany [66] argue that different disciplines provide narrow definitions of trust because only a particular aspect of trust is measured in any interdisciplinary effort. For example, economists choose to categorise rationality in the domain of economics and irrationality in the domain of sociology therefore they will rather view trust as a "rational choice" [91]. Williamson [91] concludes that it is contradictory to use the term *calculative trust* to describe commercial exchange for which cost-effective safeguards have been devised to support an efficient exchange – trust should, at best, be viewed as a choice that was arrived at due to a rational process. This view of trust - even though effective - is applicable to the economics discipline only. Other disciplines view trust on different levels such as risk, confidence or probability. Different definitions of trust exist in literature, and this section explores different approaches to trust by comparing a dictionary definition of trust to scientific definitions [66] by Gambetta [39], Barber [21] and McKnight and Chervany's trust-related construct [66].

3.2.1 Dictionary definition of Trust

The Oxford English Dictionary [10] gives a few definitions for trust which will here be categorised into two definitions. The first definition defines trust as *confidence* in something or someone and *reliance* on something or someone.

> **Definition 1**: Confidence in or reliance on some quality of a person or thing, or the truth of a statement.

Trust involves choices between alternatives in spite of the risk of being disappointed [68]. Confidence is the degree of certainty attached to the expectation that a friendly action will be reciprocated. Thus there is a recursive relationship between confidence and trust [60]. An increase in the level of confidence encourages "more" trust. Self-confidence allows us to better accept unexpected problems and makes any insecurity 'bearable' [12]. Adler [16] and Baier [19] argue that reliance on something or someone is not necessarily an indication of trust. We may rely because we have to or because it is the best or the only choice available to us [12]. Reliance on another individual depends on the inherent good nature of that individual. This is different from their dependable habit [19].

The second definition introduces *trustworthiness* as an attribute of trust.

> **Definition 2**: The quality of being trustworthy; fidelity, reliability, loyalty, trustiness.

Trustworthiness is seen as a property of a trustee, while trust involves the temporal 'relationship' between the truster and the trustee [12]. Trust involves an individual's expectations regarding of actions of others [34]. Trustworthiness concentrates on overall disposition [34] - what is the trustee's natural characteristic, that is, what is the trustee's background? What is the trustee's course of action? What have the trustee been prone to do in the past? Answering all these questions will hopefully form an opinion about the trustee's trustworthiness or reputation.

3.2.2 Gambetta's definition of trust

Gambetta [39] defines trust as a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both *before* he can monitor such action (or independently of his capacity ever to be able to monitor it) *and* in a context in which it affects *his own* action. This definition of trust intimates the existence of a probabilistic distribution if the trustee is unknown and the truster is ignorant of the trustee's action. Unfortunately this eliminates the *risky* part of trust if a truster's decision is based on probability that the trustee may not be trustworthy. In this respect, trust concerns not future actions in general but all future actions that condition our present decisions [39].

Gambetta's [39] view of trust also allows for trustees to have a degree of freedom to disappoint the truster's expectations. This applies both ways for trustees and trusters. Trustees must have the freedom, whenever possible, to either exit, betray or defect. However, if the trustee is free to do whatever he/she wants to do, then there will be no need for trust; the truster should rather hope than trust. On the other hand, the truster is also free to leave the relationship if it turns out to be more risk than trust. Luhmann [60] summarises trust as a device for coping with the freedom of others.

3.2.3 Barber's perspective of trust

Barber [21] clarifies trust with an analysis of "trustless" or distrustful aspects of American society. His views on trust are widely discussed ([12, 39, 61, 63, 92]) and are as follows:

1. Expectations that the natural order – both physical and biological – and the moral social order will persist and be more or less realised

2. The expectation of technically competent role performance, for example, expert knowledge, technical facility or everyday routine performance [66]

3. Expectations of fiduciary obligation and responsibility (i.e. in our social relationships we have moral obligations and a responsibility to demonstrate a special concern for other's interest above own)

Social relationships especially social structural and cultural variables [21] constitute the main thrust of Barber's view of trust. Expectations are placed on social systems [63] such as the government to maintain social order and behave selflessly. This type of trust can be juxtaposed with *trusting intention* [66] – one of six trust-related constructs proposed by McKnight and Chervany. Trusting intention is the extent to which one party is willing to depend on the other party in a given situation with a feeling of relative security, even though negative consequences are possible.

Trusting intention involves some form of risk, which is a term synonymous with trust. An individual bestows trust at his/her own risk [82]. Trust requires a previous engagement on the truster's part – it presupposes risk [61]. Expectation of a favourable outcome is always anticipated when trust is involved but risk of an unfavourable outcome is always a possibility. Trust is only required if a bad outcome would make you regret your action [61].

Different perspectives on trust have been discussed. Each researcher discussed above present a different view of trust as it applies to their field of study. A common view shared by Gambetta [39], Barber [21] and Luhmann [60] is that a decision to trust involves risk and there is a probability of an unfavourable outcome. A trust relationship is not formed immediately after an interaction but a social relationship is formed upon which trust is built.

3.2.4 McKnight and Chervany's Six Trust-Related Constructs

There are different types or forms of trust that are available in literature and these concepts cannot be represented exclusively. However, McKnight & Chervany [66] have presented the following six trust concepts in scientific and everyday usage.

1. *Trusting intention* (defined above) trusting exists on a personal basis rather than on a group or societal level. Trusting intention typifies *potential negative outcomes*, dependence, feelings of security, situation-specific context and a lack of reliance on control.

2. *Trusting Behaviour* is a voluntarily dependence on another person in a specific situation with a feeling of relative security even though negative consequences are possible. *Dependence* indicates a behavioural action that implies an acceptance of risk [65]. When the truster decides to give the trustee a fiduciary obligation [66] the trustee has a measure

of power over the truster. The action to trust is based on trusting intentions.

3. *Trusting Beliefs* refer to the confidence one has in believing that a trustee is trustworthy in a particular situation. Trustworthiness in a trustee implies that the trustee will act in the truster's best interest. The belief in a trustee's trustworthiness is based on confidence in the trustee's actions underlined by a relationship with the trustee. A relationship must exist between the truster and the trustee for the truster to place confidence in the trustee's benevolence, honesty, competence and predictability.

4. *System Trust* is the belief placed in impersonal structures [66] that necessary support systems are put in place in case of an unexpected event. Also known as context trust [49], this type of trust is similar to Barber's [21] view of trust in terms of social structures and cultural variables. McKnight and Chervany distinguish between two types of impersonal structures: structural assurances and situational normality. Factors of this type of trust include critical infrastructures, insurance, legal system and law enforcement.

5. *Dispositional Trust* describes the truster's general trusting attitude. It is a pervasive attitude toward oneself and the world [12, 66]. It is also described as the truster's general tendency to trust across different situations. How much trust is extended towards a trustee and reaction to feedback is based on dispositional trust. Trust disposition is a major part of who we are and it is rooted deeply in childhood experiences [25].

6. *Situational Decision to Trust* is the extent to which a truster intends to place trust in a given situation every time a particular situation arises - regardless of the truster's beliefs. In this *situation*, the truster's belief

in the benefits of the situation outweighs the perceived risk.

3.2.5 Summary: Definition of trust

According to the above definitions, trust has the following attributes:

1. It is a confidence in something or someone.

2. It involves risk.

3. It has a subjective probability.

4. It is also a form of expectation placed on the trustee.

The author's own definition of trust with regard to a service-oriented network environment is as follows:

> *Trust is a level of reliance placed on a service provider, based on its accessibility and the expected outcome of the service provided.*

This definition will be used for the remainder of this dissertation. This type of trust can be categorised under McKnight and Chervany's concept of system trust due to the impersonal nature of routing devices and the support systems in this environment. Trust models are support systems that are in place in case of an unexpected event. Different trust models are discussed in the following section.

3.3 Trust Models and Reputation Systems

Trust models are computational efforts to represent trust relationships and trust values in a virtual environment. These models attempt to assign a (numeric) trust value to an interaction either to encourage or restrict an interaction. Trust models are useful for the collection of trust values from the interaction among agents, from recommendations or from the history of agents [29]. Developments of trust models have been motivated by firstly by concerns of security in distributed systems where it is difficult to verify users' identity before interacting, secondly by the need to manage vast quantities of complex data and filter signal from noise [12], and thirdly by encouraging long-term relationships between interacting strangers on an e-commerce platform [79]. Trust has been modelled in the following areas of research: computer network security, electronic commerce, on-line auctions and peer-to-peer systems. Computer network security research is relevant to this study of trust. The implemented trust model of Abdul-Rahman and Hailes [12], Mui et al. [70]'s implemented trust model as well as eBay's [8] reputation system are discussed in the following subsections.

3.3.1 Abdul-Rahman and Hailes: Reputation Model

Abdul-Rahman and Hailes [12] propose a reputation model called Network Trust Opinions (Ntropi). This reputation model is a type of recommender trust model [92, 41] that allows participating agents to share trust values with one another. This brings up one of the (disputed) properties of trust – its transitivity. Trust is not transferable to other objects or to other people who trust [61], because trust is subjective to the trustee and to the context where trust is applied. Abdul-Rahman and Hailes suggest that agents be used in their Ntropi reputation model to recommend other trusted agents to one another -in other words, if Alice trusts Bob and Bob trust Cathy, then Bob can *recommend* Cathy to Alice. Thus, based on the trust relationship between Alice and Bob, Bob can conditionally transfer his trust relationship experience with Cathy and recommend Cathy to Alice. This is known as conditional transitivity [13] and the conditions that permit conditional transitivity include the following:

1. Bob's recommendation of Cathy to Alice must be communicated explicitly as a recommendation.

2. Recommender trust must exist in the system.

3. Alice is allowed to make judgements about the 'quality' of Bob's recommendation.

4. Trust is not absolute, because the trust relationship between Alice and Cathy may be different to Bob and Cathy's relationship.

Ntropi handles two types of trust: Interpersonal Trust and Dispositional Trust. Interpersonal trust is concerned with an agent's reputation, while dispositional trust indicates how trust is granted initially, how fast trust is built and how fast it is destroyed [12]. Dispositional trust is also known as an agent's policy. Every agent has a policy that governs its reaction to events such as threshold parameters, other agents and their recommendations and dynamics of interpersonal trust. This policy is maintained by the software agent's (human) owner either manually or through an automated process.

Furthermore, Abdul-Rahman and Hailes define experience as information gathered in a trust relationship personally with another agent or a recommendation from another agent. Trustworthiness of an agent is measured by looking at an agent's direct experience with other agents and reputation of an agent with another agent. Ntropi distinguishes between direct trust and recommender trust. Direct trust deals with an agent's direct experience with another agent which is subsequently evaluated on one of the following five trust levels: Very Trustworthy, Trustworthy, Moderate, Untrustworthy and Very Untrustworthy. Recommender trust deals with trust associated with a recommender agent and it is also evaluated based on one of the five trust levels defined.

Agents in Ntropi also go through one of these phases in their trust relationship: Unfamiliar, Fragile, Stable and Untrusted. Transitions from one phase to another depends on threshold parameters set in an agent's policy. Each of these concepts, experiences, phases and trust, are context specific. Therefore an agent may be in different phases with other agents based on the context. Agents in this model compute their own trust value about another agent with whom they are in interaction. The trust value as discussed above is context specific and is either direct or indirect (recommender) trust.

3.3.2 Mui, Mohtashemi and Halberstadt: Computational Model of Trust and Reputation

Mui et al. [70, 71] introduced the concept of *reciprocity* and cooperation in reputation models. The basic idea of reciprocity is that individuals react to positive actions of others with positive responses while negative actions elicit negative responses. In the social sciences, reciprocity strategies have been proposed in tit-for-tat strategy scenarios that have been studied extensively by game-theoreticians in the Prisoners' Dilemma game [18]. Basically the game illustrates a hypothetical situation where two criminals are arrested and in the process of questioning them separately the authorities try to get the prisoners to either *cooperate* or *defect*. The prisoner's decision, either to say nothing (i.e. cooperate with each other) or to defect (i.e. incriminate each other) will earn them equal sentences. Otherwise, if one of the prisoners defects in exchange for a shorter sentence, the other prisoner would have no "move" left to make. The criticism on and the flaws of the game will not be discussed in this study they are merely used here to illustrate how cooperation and defection work in the context of reciprocity. According to Mui et al. not all interacting parties will cooperate, unless they are publicly bound to an agreement. Some will cooperate only in contexts where some form of reciprocation in their favour is expected.

Reciprocity is defined as a mutual exchange of deeds such as favour or revenge. Two types of reciprocity are discussed: direct and indirect reciprocity. Direct reciprocity is the interaction between two concerned agents, while indirect reciprocity refers to the interaction between two concerned agents but interceded by mediating agents. Reciprocity is measured in this model either by viewing it as a social norm shared by agents in the society or by viewing it as a dyadic reciprocity between two agents. In other words, no expectation is placed on interacting agents to reciprocate, but an agent's interaction should have an influence on the agent's reputation as a reciprocating agent. Mui et al. emphasise that reputation is an important attribute for reciprocative actions. They continue to define reputation as the perception an agent creates through past actions about its intentions or norms while trust is a subjective expectation an agent has about another's future behaviour based on the history of their encounters.

3.3.3 eBay's Feedback system

Trust among strangers is difficult to establish because of the lack of past histories or the prospect of future interaction [79]. There is always a tendency for strangers to free-ride or misbehave since their behaviour will hold no future implications. Reputation systems seek to establish the shadow of the future to each transaction by creating an expectation that other people will look back on it [79]. Reputation systems should therefore have the following attributes [79]:

- Long-lived entities that inspire an expectation of future interaction.
- Capture and distribution of feedback about current interactions (feedback must be visible for future purposes).

- Use of feedback to guide trust decisions.

An example of a commercially implemented reputation system is eBay [8]. There are many other sites with implemented reputation systems such as, Yahoo! Auction [94], Amazon [1], Bizrate [77] but the feedback system of eBay's online auction system has been widely discussed (see [80, 78, 79, 49, 36, 29, 95]). The sellers and buyers come together as strangers who have no fore-knowledge about each other. They are registered under a pseudonym that is visible to other buyers and sellers of the auction system. Although eBay allows the buyers and sellers (hereafter referred to as users) to choose any online pseudonym so as to encourage anonymity, eBay requires a valid email address for users in order to verify their details. Buyers don't have to provide their credit card details when registering but buying any goods in future will require a valid credit card detail. Sellers, on the hand, have to provide their credit card details to eBay for verification purposes.

eBay uses a feedback system to record the user's reputation as either a "good" or "bad" user. The feedback system is also called a reputation system. When users buy or sell on eBay they get an opportunity to leave voluntary feedback about their experience. This is replicated by their trading partner. Users build up a reputation based on feedback given by other users after a transaction. Giving feedback depends on the completion of a transaction between a seller and another winning bidder. Feedback can either be positive, negative or neutral and may include a short comment. A numerical rating of +1, 0 or -1 is given for positive, neutral and negative feedback respectively. Even though it is a voluntary action eBay encourages its users to leave a feedback and wherever possible to try and resolve an issue before giving neutral or negative feedback comments because they are permanent on a user's profile. Feedback comments are useful for determining a buyer or seller's trustworthiness and also for deterring free-riders.

Reputation systems like eBay's feedback system as described by Chang et al. [29] are useful for collecting information about products and the quality of services delivered by anonymous and unknown business providers. This system also has the following benefits [29]:

- It assists in the verification of sellers and fair trading.
- It provides a technological platform for social recommendation and trustworthiness measures.
- It builds up value-added relationships by improving loyalty between buyers and sellers, the site and the customers, providers and end-users.
- It can assimilate recommendations relating to trustworthiness from former users who can share their experiences.

There are two main types of reputation systems – centralised and distributed. In a centralised architecture, a central authority gathers reputation values and acts as a repository for the participants in the reputation system. Participants look up each other's reputation scores in the central repository before transacting with each other. A distributed architecture allows each participant to record and maintain the reputation value of other participants with whom they have been in direct contact. Participants need to find each other's reputation value before transacting with one another. Collecting, calculating and distributing reputation values are up to the participant. Direct experience with another participant will usually carry more weight than received reputation values from other participants.

In this study of trust we have used the terms trust models and reputation models interchangeably because trust is used synonymously with reputation. Gathering trust values about an entity shows the reputation of that entity independent of the quality of the reputation. Likewise, the reputation of an entity shows how worthy or unworthy of trust it is. Although this may sound like linguistics semantics Jøsang [49] differentiates between trust systems and reputation systems.

1. Reputation systems produce an entity's public reputation score as seen by a whole community. On the other hand, trust systems produce a score that reflects the relying party's subjective view of an entity's trustworthiness.

2. Transitivity is an implied concept in reputation systems while it is an explicit component of trust systems.

3. Input in a trust system generally consists of subjective and general measure of reliability while information about specific transactional events is used as input in a reputation system.

3.3.4 Summary: Motivation for Ntropi

The characteristics of virtual environments differ from those of a physical environment where trust is implicit. Reputation-trust models, are needed to model trust for human and non-human entities to interact in virtual environments. The current research project proposes trust in a network environment. Despite the fact that this environment requires a decentralised reputation-trust a centralised architecture is employed using the Ntropi reputation-trust model proposed by Abdul-Rahman [12].

Most of the trust models available in literature are reputation-based. That is, they collect input from agents in their environment. The models discussed above are relevant for the environment they are proposed for. They are often cited in research literature and have a good "reputation" among experts of computer security ([29, 33, 64, 70, 92, 87]). However, the model chosen for implementation in this study is Ntropi.

- Ntropi is a general cross-application model of trust. It is not bound to a single application domain.

- Ntropi is an uncomplicated reputation trust model that encourages participation and input from agents in its implementation environment.

- Ntropi reflects the dominant trend in types of model currently proposed and implemented. [92].

- Ntropi has a wide influence with computer security and trust experts.

3.4 Trust metrics

Clarification on dependability in routing is given below due to the large number of research efforts in the area of dependable routing and because this project's premise might be confused as a research under such category. Noteworthy references in dependable routing include those by Hollick et al. [42], Castro et al. [26] and Pirzada et al. [76].

Measuring trust in this study, that is, trust in routing devices in a network environment is done based on routing device availability and reliability. Availability in terms of security is the concurrent existence of service availability for authorised service users only [17]. This includes timely response to requests, work-arounds to hardware faults, prevention of abrupt loss of information and ease of use of service as it was intended [75]. Service availability is essentially a prevention of a denial of service attack. Computer security's past success has focused on confidentiality and integrity the full implementation of availability is security's next great challenge [75].

3.4.1 Dependability in Routing

The International Telecommunication Union's Telecommunication Standardisation sector (ITU-T) [85] defined four performance concepts or building blocks for the traditional telephone networks (PSTN) and integrated service digital networks (ISDN). These building blocks are intended to measure performance across the network and are as follows: quality of service (QoS), serveability, trafficability performance and dependability. These blocks are further divided into two parts (see Figure 3.1), namely application and service performance and network performance. The part that is of interest to this study is network performance, which consists of trafficability performance and dependability.

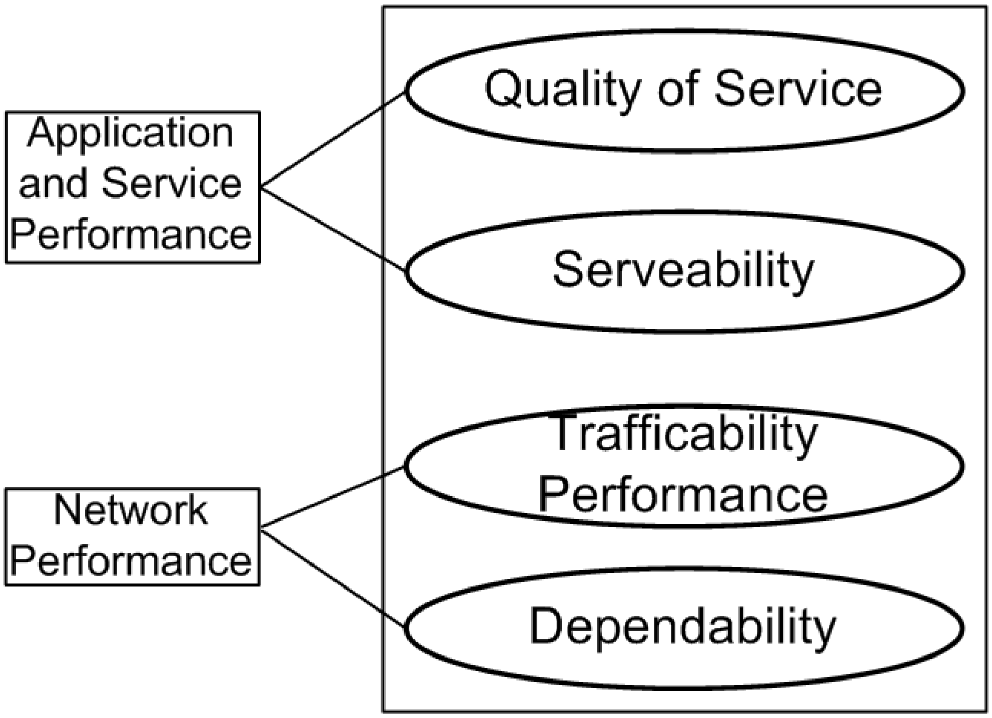


Figure 3.1: Performance concepts building blocks adapted from ITU-T [85]

Trafficability performance is the ability of an item to meet a traffic demand of a given size and other characteristics under given internal conditions [85]. Trafficability performance, also known as, traffic engineering is examined in detail in the ITU-T recommendation [84]. Although it lies outside the scope of this study, traffic engineering addresses issues such as network traffic routing, traffic distribution in a network environment, blocked route, internal traffic and outgoing network traffic.

Dependability is the collective term used to describe availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance [85]. The concepts of dependability consist of three parts: threats, attributes and means, as shown in the dependability tree in Figure 3.2.

For the purposes of this study, our focus is on two of the attributes of dependability namely, *availability* and reliability. Avizienis [17] defines these attributes as "*readiness and continuity of correct service*" respectively. This study examines availability as it applies to the three pillars of security: confidentiality, integrity and availability. And not availability as it is implemented in QoS dependability.

Even though QoS dependability (described above) has the same similarities as this study, its implementation - routing dependability - involves an in-depth examination of routing protocol architecture. On the other hand, this study focuses mainly on services provided by routers or a routing system and their availability on a very high-level. The underlying routing protocol architecture is explicitly ignored and attention is given to the routing devices and how trust can be used to measure their availability.

3.5 Conclusion

Trust models are essential in virtual environments to gather participants' actions which are then summarised as trust attributes for each participant. In

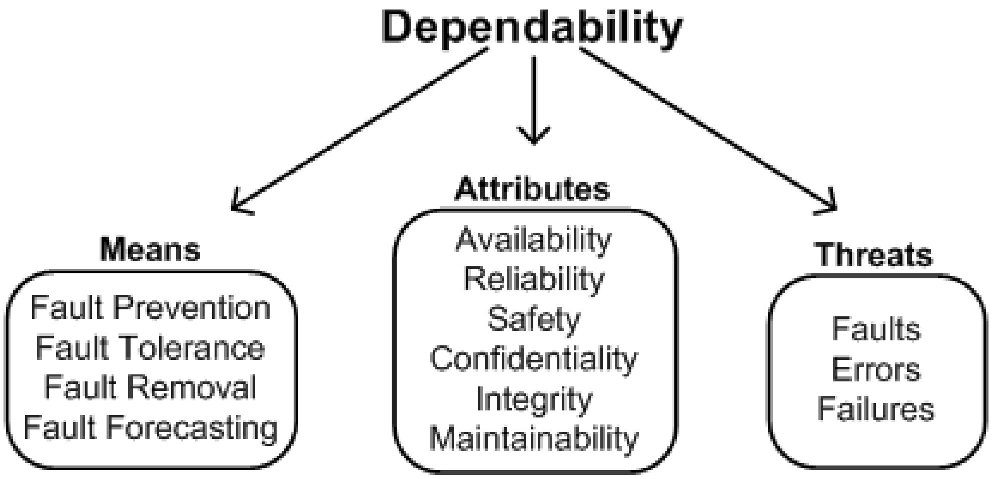


Figure 3.2: The dependability tree adapted from [17]

answering the research question posed at the onset of this chapter, the Ntropi trust model is indicated as the model to be used to determine trust attributes of routing devices. This chapter furthermore provides the reason for using the Ntropi model and specifies the importance of trust among routing devices in a network environment. The chapters that follow will explore how trust is implemented in a service-oriented network environment.

Chapter 4

Requirements of TSONE

> *The two most important requirements for major success are: first, being in the right place at the right time, and second, doing something about it.*
>
> – *Ray Kroc*

4.1 Introduction

In preceding chapters, a service-oriented network environment (SONE) was introduced. SONE was defined as a *collaborative environment where network devices utilise their resources to publish and discover services available in a network environment*. The service context considered in this study is a network environment while a router is the service provider and service requester. As discussed in Chapter 2, a SONE can be modelled as another instance of a service-oriented architecture (SOA) but an SOA requires additional conceptual framework and architectural elements for trust [14, 73]. Chapter 3 presented different perspectives on trust and gave a definition for trust as it applies to this research project: *Trust is a level of reliance placed on a ser-*

vice provider based on its accessibility and the expected outcome of services provided.

This chapter and the subsequent one endeavour to answer the research question: *How can trust be represented in a network environment?* The end result that is sought is to demonstrate a method to model trust in a service-oriented network environment (TSONE). The section below commences by discussing the traditional network layer security and why trust is an effective alternative. The requirements of the TSONE model identified by the author are presented in Section 4.3. The chapter is concluded in Section 4.4.

4.2 Network Layer Security

Security is provided at the OSI's network layer by a suite of protocols called IP security protocol [51, 86], also know as IPsec. IPsec was developed by the IP security protocol working group under the IETF to bring TCP/IP up to today's security standard [54]. Security at the network layer involves maintaining the confidentiality and integrity of packets before sending them out on the network. This is accomplished by using any of the available encryption methods, for instance symmetric key encryption, public key encryption or public key encryption using session keys.

Any of the encryption standards will prevent packet mistreatment - one of the security attacks discussed in earlier. In addition to confidentiality and integrity, the network layer can also provide source authentication [57]. This is done when the destination host/router receives a packet and the source of the packet is authenticated against the router address originating host. Source authentication can prevent DNS hijacking attacks.

The IPsec protocol suite provides two protocols for network layer confidentiality, integrity and authentication, namely authentication header (AH) protocol and encapsulation security payload (ESP) protocol. Before explain-ing the AH and ESP protocols a brief examination of a security agreement (SA) is provided.

An *SA* is network-layer or a simplex logical connection between two communicating IP endpoints that provides security services to the traffic carried by it using AH or ESP protocols [54]. The SA is a handshake between a network source and network hosts and it is composed of the following [57]:

1. A security protocol (AH or ESP) identifier,
2. The source IP address
3. A connection identifier known as the security parameter index (SPI)

The *AH* protocol provides a mechanism for data integrity and source authentication. After an SA has been established between a source and destination host, the source sends secure datagrams that consist of an AH header inserted between an IP packet data and the IP header. When the destination host receive the packet, it determines the SA before authenticating the packet's integrity by using shared keyed-hash message authentication code (HMAC).

The ESP protocol on the other hand, provides network-layer confidentiality as well as authentication mechanisms. The ESP fields in an IP packet consist of the packet's data wrapped between ESP header and trailer segments and ESP authentication mechanism as the trailer of the IP packet. The packet's data and ESP trailer are encrypted with the data encryption standard cipher block chaining (DES-CBC).

The network layer, through the IPsec protocol, provides adequate security for IP packets travelling in a network environment. IP packets are protected against attacks targeted at compromising the integrity and/or secrecy of an IP packet. However, the network devices that route IP packets are still vulnerable to attacks such as routing table poisoning and distributed denial of service. An example of a distributed DoS attack is the route hijacking of YouTube's advertised route [22, 27, 47, 67], which propagated to most of the Internet Network before it was curtailed. The issue of trust was raised - both in the Internet network or any network environment [47] and in a service-oriented environment ([14]). In the next section, trust in service-oriented network environment (TSONE) is presented.

4.3 Requirements for TSONEs

The requirements for TSONEs are discussed in this section. These requirements are a combination of attributes of reputation systems discussed in Chapter 3 and distributed service-oriented architecture principles discussed in Chapter 2. Reputation and trust determination are part of the core functionality of a TSONE, along with characteristics of service-oriented architecture adopted in the network environment. The list of requirements for TSONEs includes following:

- **Persistent and long-lived**: As long as routers are set up and actively routing packets in a network environment a TSONE must be capable of determining their lifespan and longevity. A network administrator cannot control the persistence of a router but should be able to view the router's accessibility to other routers in the environment.

- **Implementation-independent and flexible configuration**: A TSONE must be able to interface or connect with any network environment regardless of the different routing hardware devices. The TSONE also needs to be flexible regarding the routing protocol employed in the network environment.

- **Capturing of activity and current events in the network**: The TSONE needs to be able to capture events in the network environment via the routers. The router reports on events in the network environment and also on changes in the router's system.

- **Using activity to guide trust decisions**: Like typical reputation systems a TSONE must use activities from routers to determine the trust attributes of the routers. The priority or importance of activity will indicate a router's availability and reliability in a network environment.

- **Scalability**: Given the maximum size of an autonomous system (AS), a TSONE should be able to manage routers in the largest AS. The network management server (NMS) implements the trust model - thus both should be able to scale to the size of the network environment.

Practically, it is impossible to manage all the routing devices on the Internet due to its enormity. However, if each network on an organisational level is implemented as a TSONE and border routers on the edge of networks are monitored in terms of their availability and reliability, intrusion attacks can be minimised on the Internet. The design of the TSONE model is presented in the following chapter.

4.4 Conclusion

This chapter commenced by introducing IPSec protocol on the network layer security. The requirements for a trusted service-oriented network environment (TSONE) were discussed as a possible solution to introducing trust in a service-oriented network environment. The design of a TSONE and its various components are discussed in the next chapter.

Chapter 5

Design of TSONE

A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.

– *Douglas Adams*

Many things difficult to design prove easy to perform.

– *Samuel Johnson*

5.1 Introduction

This chapter presents the TSONE design which illustrates the different components of the TSONE model. The TSONE model consists of an environment that allows routing devices' activities to be collected. These activities are subsequently used to determine the trust level of routing devices. An external entity such as a network administrator monitors the trust level of routing devices in the network environment and updates the routing table accordingly. The following section contains a brief overview of the components that make up the TSONE model.

5.2 The TSONE model

The proposed trusted service-oriented network environment (TSONE) is modelled as a reputation system that captures interactions and behaviours of routing devices to build trust in a SONE. In this environment, behavioural attributes of routing devices are collected in a central location and the trustworthiness of each router is determined based on transactional activities collected from the routers. These activities include (but are not limited to) routing and/or forwarding data packets, broadcasting router availability, error or information reporting, network management reports and many more. The transactional activities are collected as a form of feedback about a router's behaviour. This conforms to the properties of trust as observed by Luhmann [60], Gambetta [39] and Barber [21]:

- Trust is measurable and evolves over time.
- Trust is dependent on a specific situation where risk is accepted during interactions. That is, trust is the basis on which interactions occur.

Since trust modelled in a TSONE is derived from reputation, reference is made to the attributes of reputation systems discussed in the previous chapter:

- Long-lived entities that inspire an expectation of future interaction
- Capture and distribution of feedback about current interactions and
- Use of feedback to guide trust decisions

The TSONE model is shown in Figure 5.1. This model consists of two interdependent components: the network environment and the network management server (NMS). The former (the network environment) consists of routers while a reputation trust model is implemented on the later (the NMS).

The network administrator interacts with the NMS to monitor the network environment.

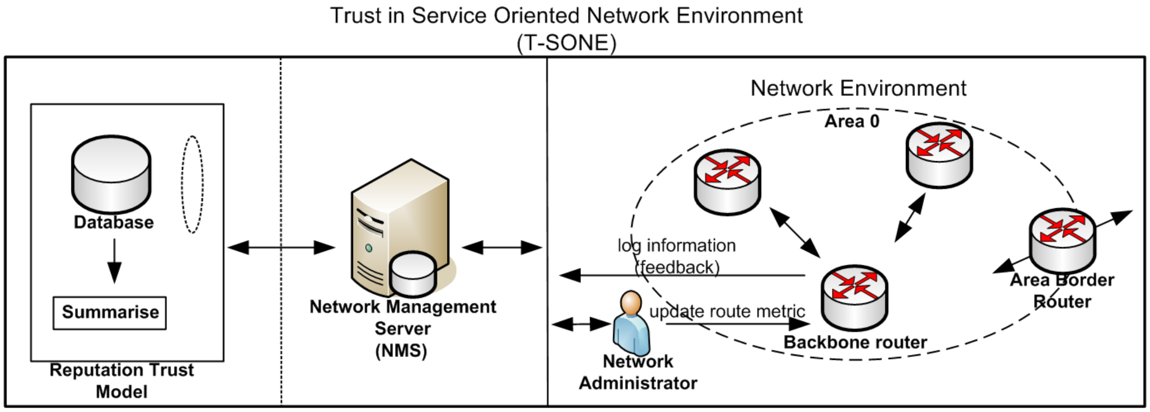


Figure 5.1: TSONE model

Figure 5.1 depicts the different parts of the TSONE model. The network environment represents the service-oriented network environment part of the model, while the trusted component of the model is implemented as a reputation trust model on the NMS. A description of the network environment, the NMS and the trust model follows below.

5.2.1 Network Environment

The network environment consists of routing devices that route packets across the network. For testing purposes it is modelled as a local area network, but it can also be an autonomous system (AS) (i.e. a group of networks under a centralised administrative system, such as a university or an organisation. The network administrator can use any routing protocol for this environment as long as all the router interfaces are accessible via Telnet or any other remote access protocol. The network administrator must also be able to control network traffic through the interfaces as needed, either by increasing or decreasing the interface or path *cost* on the router. If the path cost on a router's interface is increased, the routing tables of its neighbouring router's will be affected as well. This implies that other routers will not forward packets to the router's interface because of the high path cost.

Area Border Router (ABR): The ABR is a router that is connected to multiple autonomous systems (AS), later referred to as areas. That is, its interfaces are attached to multiple areas. ABRs run multiple copies of the routing algorithm for each area that they are part of. ABRs also condense topological information of their attached areas for distribution to the backbone. The backbone in turn distributes the information to the other areas [69].

Backbone Router: The open shortest path first (OSPF) routing protocol has an area where routing information is distributed to other parts of the network. This area is called Area 0 and it is also known as the backbone area. The backbone area consists of router(s) with multiple interfaces in different areas, in other words ABRs. However, not all routers in the backbone are ABRs. Routers with all interfaces connected to the backbone are considered to be internal routers [69].

Part of the trust functionality of the network environment is that all the routing devices on the network must log their activities to the NMS. This obviates a situation where a covert routing device is installed on the network and starts routing packets. The presence of the new router will be apparent to the network administrator who can then take the necessary action to prevent attacks on the network. If an attacker gains access to the network environment by any other means (such as unauthorised access to a router's startup configuration), the trust model via the NMS will identify the new router to the network administrator. Event logging to the NMS and network administrator intervention is essential for the network environment.

5.2.2 Network Management Server (NMS)

The network administrator interacts with the NMS to maintain the routing devices on the network. The NMS consists of the following:

- The TSONE application that manages all the routing devices in the network.
- The *KiwiSyslog* [9] third party application that collects events from the routing devices via the syslog protocol.
- The syslog file where all the syslog messages are stored.
- The database that stores the routers' information and a subset of the syslog events.

For clarification purposes, the syslog protocol provides a means for computer systems to send event notification messages across IP networks [58]. These messages are collected on a syslog server or an application that implements the syslog protocol. An example of such application is KiwiSyslog used as a daemon on the NMS.

5.2.3 The Trust Model

As stated in the preceding chapter, the network trust opinions (Ntropi) model [12] is the trust model customised for this research project. The model was designed and developed for autonomous agents in a distributed network environment. The agents in Ntropi aggregate their own trust opinions about another agent that they have interacted with. They can also get recommendations about other agents with whom they have not interacted yet. Trust types in Ntropi are divided into two: direct trust and recommender trust. Although most of the concepts of Ntropi do not apply to TSONE, TSONE adopts the five trust level defined in Ntropi. The NMS provides a platform for the network administrator to interface with the different functionalities and manage different protocols on the network via the routing devices.

5.3 The design of TSONE

In addition to the three components of the TSONE model described above, there are subcomponents that are necessary for the network environment and the NMS. These subcomponents or modules are depicted in UML (Universal Modelling Language) diagrams. The diagrams are designed using UML due to the fact that it is a standard system modelling language recognised worldwide [38].

5.3.1 Functionality of TSONE

The Use Case Diagram is used to describe the functionalities of TSONE (see Figure 5.2). The diagram contains four major elements: the **system** which is included in a boundary, the **actors** that interact with the system, the **use cases** that represent services performed by the system and the lines that represent **relationships** between the elements.

Functionalities of TSONE include the following:

- Logging events that affect the router in the network environment. The relevant log information is extracted to the database.

- Updating the routing table. This is done by the network administrator based on information provided by the trust model.

- Calculating trust levels for the routing devices. Information extracted from logged events are stored in the database where the trust model accesses them and calculates the trust level for a particular router.

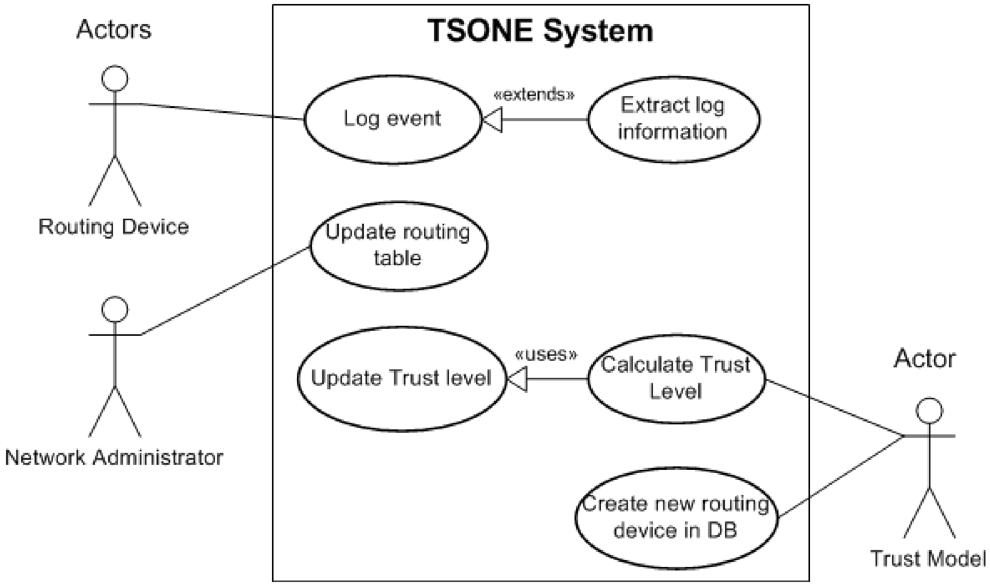


Figure 5.2: Use Case Diagram of a TSONE

- Creating new routing device instances in the database based on event logs. The trust model stores the information of a new router in the database when the new router starts logging its events to the server.

5.3.2 Implementation of TSONE

The component class diagram, in general, depicts the dependencies between the different parts of a system. The TSONE component class diagram is shown in Figure 5.3. The process begins in the network environment where routers send their various activities on the network to the NMS. These events are collected in the syslog daemon and stored in a text file. The NMS extracts the necessary information from the log file and save the information in the database. Information extracted is used in the trust model to determine routers' trust attribute. The trust attribute derived from the trust model is used by the network administrator to update a router's routing table.

Some parts of the TSONE model have been discussed above: the network environment, the trust model and the network management server. A more detailed explanation follows of additional components of TSONE shown in Figure 5.3 (the network administrator, the router, the syslog daemon, and the database).

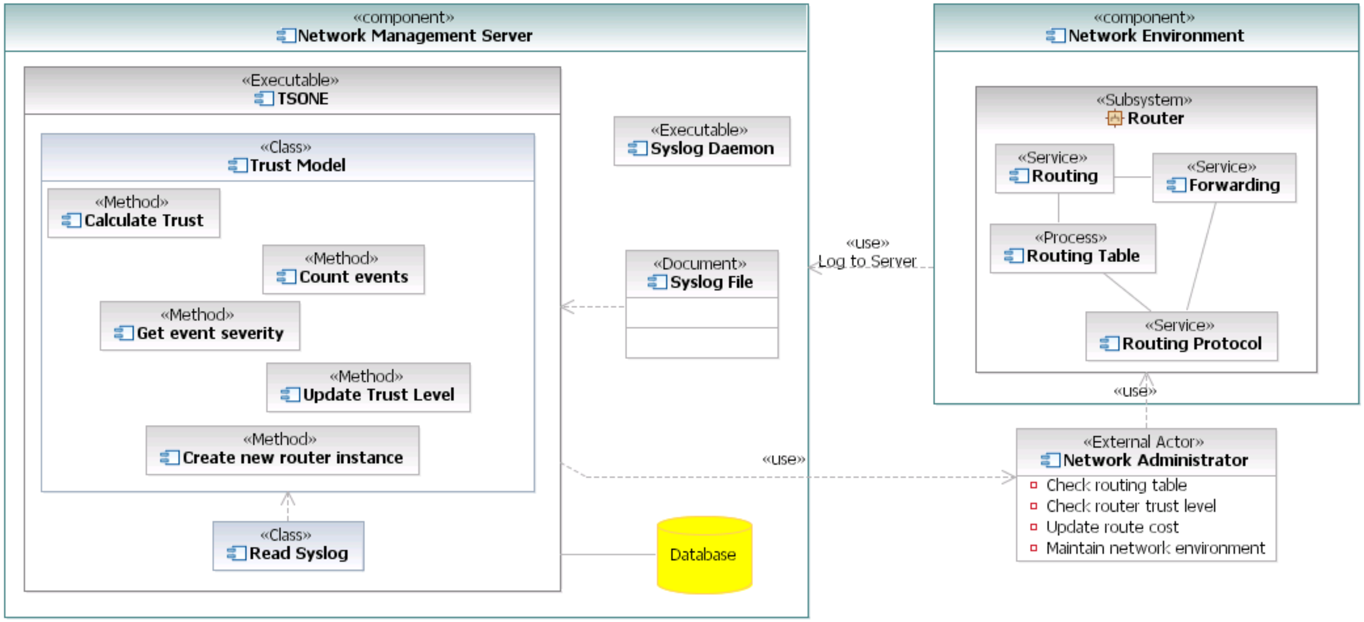


Figure 5.3: UML Component class Diagram of the TSONE model

5.3.2.1 The network administrator

The network administrator is an external (human) actor that maintains the network environment. The human component in virtual trust has not been discussed in this research project because determining the *trustworthiness* of humans is both a social and academic research question. Nevertheless, it is essential that a network administrator is included in a TSONE because the automating of updates in a network environment is impossible without authentication to the router's operating system. Establishing the trustworthiness of a network administrator for a TSONE is therefore left up to the necessary screening processes involved when delegating the responsibility.

Provided that a network administrator is bona fide, his/her responsibilities include a comprehensive maintenance of the network environment as well as the following:

- Checking the routing tables in the network environment
- Checking the router trust level periodically as it is updated in the trust model
- Updating route costs of affected routers in the network environment

5.3.2.2 The router

A router's functionality was discussed in preceding chapters thus its functionalities as a subsystem in the network environment are discussed next. The network administrator has access to the router's operation system through an application that allows the network administrator to make changes to the router's internal system. A router in TSONE must be able to:

- send its activities to the NMS using the syslog protocol
- implement a routing protocol supported by the network environment and

- accept changes made to routes' metrics in a routing table

5.3.2.3 The Syslog Daemon

The syslog daemon is an application invoked in the background that is constantly polling routing devices for syslog event messages. The daemon is implemented on the NMS. When the event messages are received the daemon saves the events in a text file modelled as a document component in the component diagram (Figure 5.3).

The system message log (syslog) is a software programme that saves system messages in a log file or direct messages to other devices [31]. Syslog has the following features:

- Provides logging information for monitoring and troubleshooting.
- Provides for the selection of the type of logging information captured.
- Allows for selecting the destination of captured logging information.

A syslog message can be divided into three parts: the *header* part consists of the time stamp and the host name, the *priority* part consists of the facility and severity of message generated and the *message* part contains the text of the message. The following is an example of a syslog message:

```
2008-02-25 20:08:32  Local7.Alert  30.30.30.2 Module 2 inserted
```

The different parts of the above syslog message are specified below:

$$\underbrace{2008-02-25\ 20:08:32}_{Timestamp}$$

$$\underbrace{Local7.Alert}_{Facility\ Severity}$$

$$\underbrace{30.30.30.2}_{Hostname}$$

$\underbrace{Module\ 2\ inserted}_{Message}$

A facility in a router can be a protocol or a module within the operating system. The severity of a message reflects the importance of the message generated. The severity level is numerically coded from zero (0) to seven (7). The lower the number the more severe the message. Tables 5.1 and 7.1 depict the different facilities and severity levels that can be attributed to a syslog message.

5.3.2.4 The Database

The database is the core of the trust model. As discussed previously, trust models are effective if behavioural activities can be captured and used to determine trustworthiness over a period of time. Thus, the database's functions for a TSONE include the following:

- Capture and store information about devices' activities from the device's log file.
- Maintain a consistent information for the lifetime of the device in the network environment.
- Distribute device information when needed to determine devices' trustworthiness.

An entity relationship diagram (ERD) and a detailed explanation of the entities are provided in the following chapter.

Table 5.1: Syslog Facilities [58]

Code	Facility
0	Kernel Messages
1	User level Messages
2	Mail system
3	System daemons
4	Security or Authorisation Messages
5	Messages generated internally by Syslogd
6	Line printer subsystem
7	Network news subsystem
8	UUCP daemon
9	Clock daemon
10	Security or Authorisation Messages
11	FTP daemon
12	NTP subsystem
13	Log audit
14	Log alert
15	Clock daemon
16	local use 0 (Local0)
17	local use 1 (Local1)
18	local use 2 (Local2)
19	local use 3 (Local3)
20	local use 4 (Local4)
21	local use 5 (Local5)
22	local use 6 (Local6)
23	local use 7 (Local7)

Table 5.2: Severity level scale [58]

Code	Severity
0	Emergency: System is unusable
1	Alert: Action must be taken immediately
2	Critical: Critical conditions
3	Error: Error conditions
4	Warning: Warning conditions
5	Notice: Normal but significant condition
6	Informational: Informational messages
7	Debug: Debug-level messages

5.4 Conclusion

This chapter has provided a high-level design outline of a TSONE. The system and the different components that make up a TSONE were represented in the Universal Modelling Language (UML). Given the main problem statement of this work, namely assessing the trustworthiness of a routing device, the proposed design aims to answer the question. The proposed design intends to satisfy the requirements of a trust model. The next chapter contains a detailed explanation of the component class diagram and its components.

Chapter 6

Detailed Design of TSONE

Everyone rises to their level of incompetence.

– Laurence J. Peter

The difference between something good and something great is attention to detail.

– Charles R. Swindoll

6.1 Introduction

In Chapter 2, a service-oriented network environment (SONE) was introduced, after which existing trust models were reviewed in Chapter 3. A trust model that was deemed appropriate for this environment was chosen and reasons were provided for choosing the specific trust model. The requirements of a trusted service-oriented environment (TSONE) were elaborated on in Chapter 4. The next chapter (5) presented an architectural diagram of the TSONE model as shown in Figure 6.1. The current chapter continues to present the prototype design of a TSONE.

Chapter 6 also includes the following Section 6.2 elaborates on the classes that make up the component class diagram. Section 6.3 gives an overview of the preliminary work that was performed to set up the lab environment for the prototype implementation. The section explains how the network components were set up, including database connectivity, collection of Syslog messages and calculation of trust levels. This chapter is concluded in Section 6.4.

6.2 Detailed Design of a TSONE

This section provides a detailed component class design of a TSONE model. The UML component diagram is reproduced in Figure 6.1. The main components of a TSONE system, the trust model class, the executable Syslog Daemon and the Syslog file are within the borders of the bold black line.

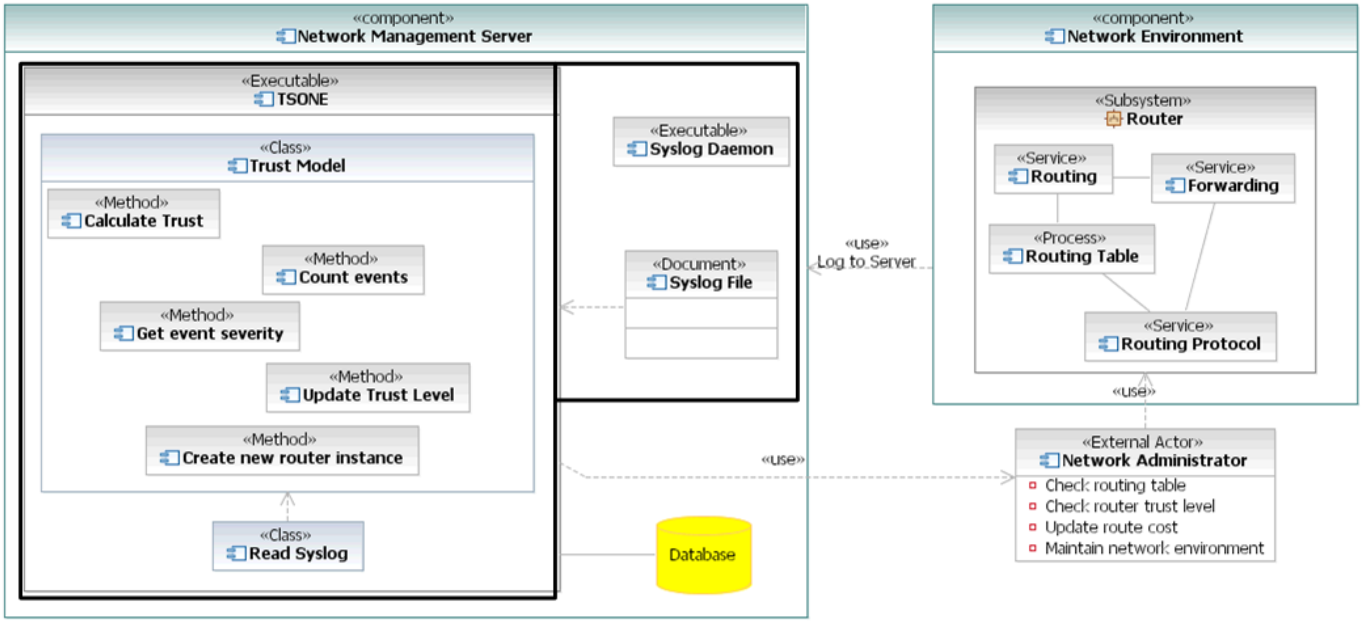


Figure 6.1: UML Component class Diagram of the TSONE model

For implementation purposes the classes in the component class diagram above are shown in Figure 6.2 below. The relationships between these classes are illustrated through their dependencies.

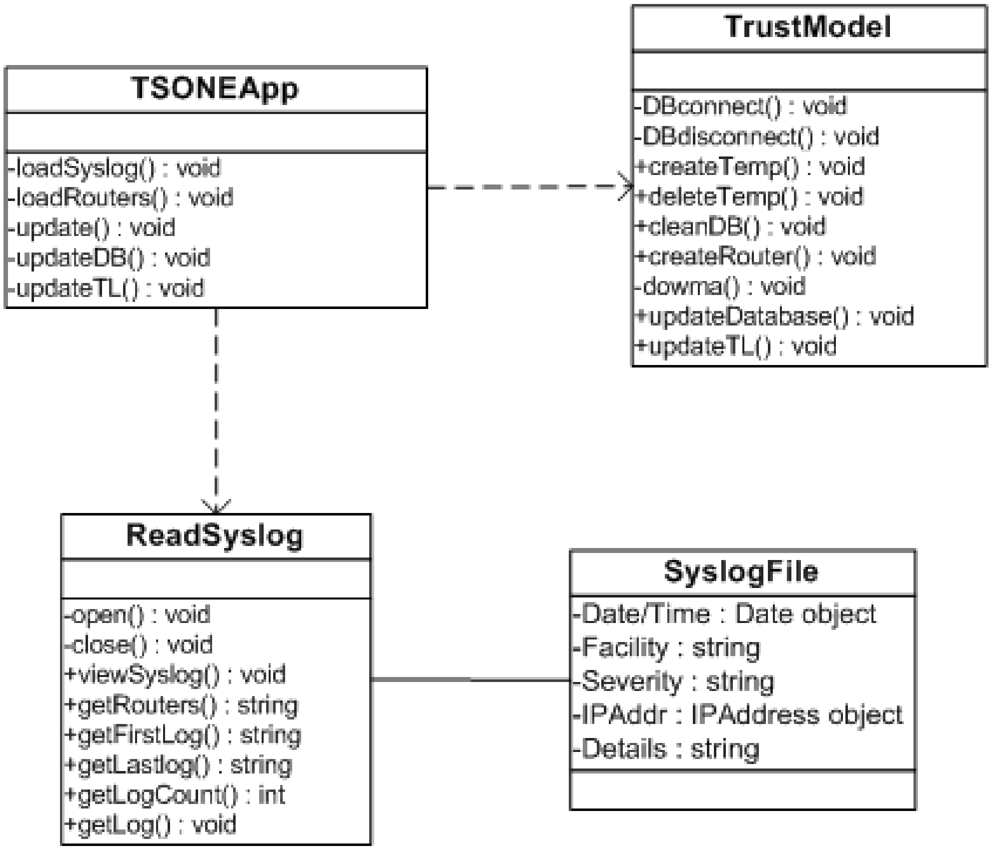


Figure 6.2: Class Diagram of the TSONE model

Figure 6.2 depicts three classes as well as their attributes and methods - *TSONEApp*, *TrustModel* and *ReadSyslog*. These classes constitute the TSONE system. The dashed arrows indicate dependencies between the classes while the source class (TSONEApp) depends on the target class (TrustModel and ReadSyslog) to implement its methods. The horizontal line between *ReadSyslog* and *SyslogFile* indicates an interdependency between the subclass and the document respectively. The three classes are explained below.

6.2.1 TSONEApp

The TSONEApp invokes the main methods of the TSONE application. The network administrator can load routers from the Syslog file, create routers for the Trust model Database (DB), calculate the trust Level and phase value, update the DB, and view the Syslog file.

The methods implemented in the `TSONEApp` class are explained below:

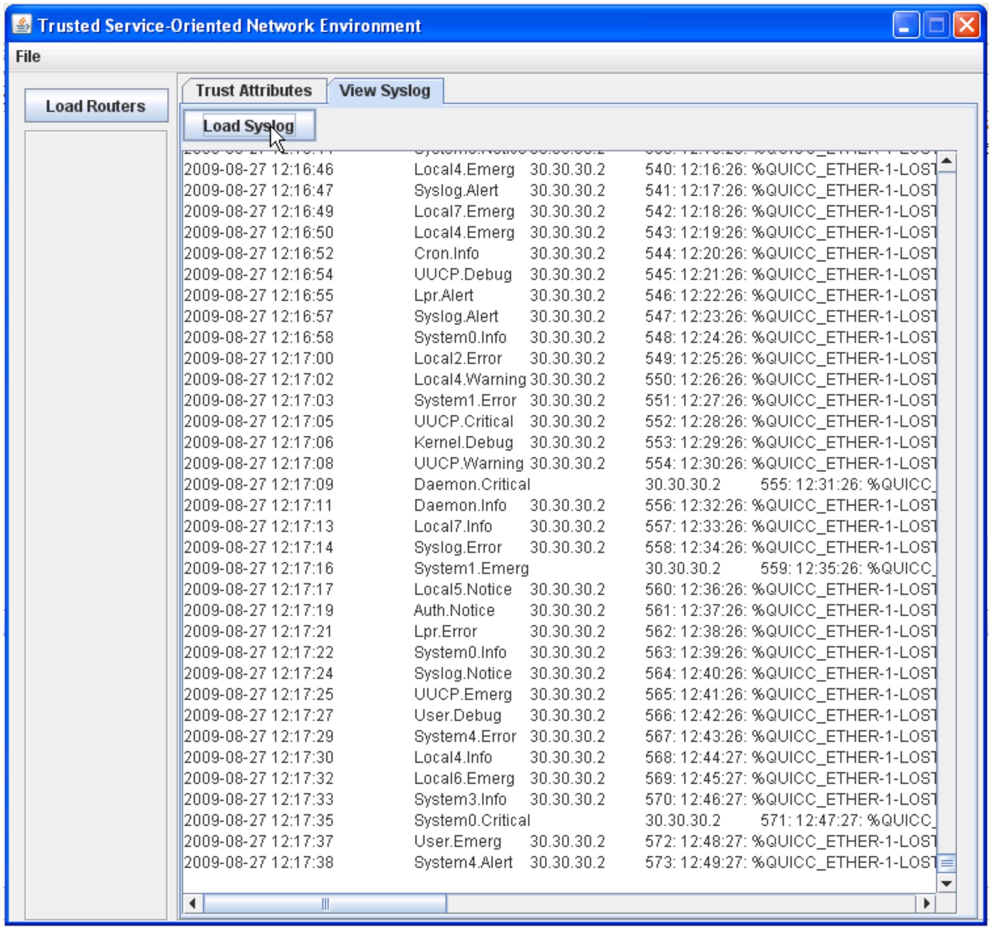
- *load()* calls the `getrouters()` array of String method of the `ReadSyslog` class to return an array of non-repeating IP addresses from the Syslog file.

- Figure 6.3 illustrates how *LoadSyslog()* loads the latest Syslog file to the text area.

- *create()* invokes the `createNetDev()` method of the `TrustModel` class to create a new instance of a network device with default parameters in the trust model DB. If an instance of the network device is present in the DB, the attributes of the network device are displayed in the relevant fields.

```
"INSERT INTO Router values(" +
"INET_ATON(’" + jtfIp.getText().trim() + "’)" + "," +
"’Unfamiliar’" + "," +
"’0’" + ")";
```

- *updateDB* invokes the `updateDatabase()` method of the `TrustModel` class to update the network device activities according to the latest Syslog information.



```
2009-08-27 12:16:46    Local4.Emerg   30.30.30.2    540: 12:16:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:47    Syslog.Alert   30.30.30.2    541: 12:17:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:49    Local7.Emerg   30.30.30.2    542: 12:18:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:50    Local4.Emerg   30.30.30.2    543: 12:19:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:52    Cron.Info      30.30.30.2    544: 12:20:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:54    UUCP.Debug     30.30.30.2    545: 12:21:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:55    Lpr.Alert      30.30.30.2    546: 12:22:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:57    Syslog.Alert   30.30.30.2    547: 12:23:26: %QUICC_ETHER-1-LOST
2009-08-27 12:16:58    System0.Info   30.30.30.2    548: 12:24:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:00    Local2.Error   30.30.30.2    549: 12:25:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:02    Local4.Warning 30.30.30.2    550: 12:26:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:03    System1.Error  30.30.30.2    551: 12:27:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:05    UUCP.Critical  30.30.30.2    552: 12:28:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:06    Kernel.Debug   30.30.30.2    553: 12:29:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:08    UUCP.Warning   30.30.30.2    554: 12:30:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:09    Daemon.Critical              30.30.30.2     555: 12:31:26: %QUICC_
2009-08-27 12:17:11    Daemon.Info    30.30.30.2    556: 12:32:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:13    Local7.Info    30.30.30.2    557: 12:33:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:14    Syslog.Error   30.30.30.2    558: 12:34:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:16    System1.Emerg                30.30.30.2     559: 12:35:26: %QUICC_
2009-08-27 12:17:17    Local5.Notice  30.30.30.2    560: 12:36:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:19    Auth.Notice    30.30.30.2    561: 12:37:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:21    Lpr.Error      30.30.30.2    562: 12:38:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:22    System0.Info   30.30.30.2    563: 12:39:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:24    Syslog.Notice  30.30.30.2    564: 12:40:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:25    UUCP.Emerg     30.30.30.2    565: 12:41:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:27    User.Debug     30.30.30.2    566: 12:42:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:29    System4.Error  30.30.30.2    567: 12:43:26: %QUICC_ETHER-1-LOST
2009-08-27 12:17:30    Local4.Info    30.30.30.2    568: 12:44:27: %QUICC_ETHER-1-LOST
2009-08-27 12:17:32    Local6.Emerg   30.30.30.2    569: 12:45:27: %QUICC_ETHER-1-LOST
2009-08-27 12:17:33    System3.Info   30.30.30.2    570: 12:46:27: %QUICC_ETHER-1-LOST
2009-08-27 12:17:35    System0.Critical             30.30.30.2     571: 12:47:27: %QUICC_
2009-08-27 12:17:37    User.Emerg     30.30.30.2    572: 12:48:27: %QUICC_ETHER-1-LOST
2009-08-27 12:17:38    System4.Alert  30.30.30.2    573: 12:49:27: %QUICC_ETHER-1-LOST
```

Figure 6.3: Load Syslog on TSONE App

- *updateTL()* is used to update the trust level of a network device. This is also a method call to `TrustModel`'s `updateTL()` method which is explained in Section 6.2.2.

- *updatePV()* also invokes `TrustModel`'s `updatePV()` method to determine a new phase value for a network device.

6.2.2 TrustModel

The `TrustModel` class is the core class of the TSONE model. It is responsible for capturing a network component's activities in the Syslog file, storing this information in a DB and aggregating it into a trust level attribute. The trust level attribute shows an indication of a network component's activities over a period of time. The functions of this class are explained below:

- *DBconnect()* establishes a secure connection to the database and *DBdisconnect()* terminates the connection to prevent unexpected storage or deletion from the database.

- *createTemp()* and *deleteTemp()* create and delete temporary Syslog file during the `updateDatabase()` process. This ensures a consistent state for the active Syslog file.

- *cleanDB()* is a method to clean up the database after trust level calculation. This ensures that the database is left in a consistent state after trust level and phase value calculation.

- *createRouter()* creates an instance of a network device in the database. This method is called from the `update()` method of the `TSONEApp` class.

- *updateDatabase()* populates the database with new event information in the Syslog file. This method is invoked from the `updateDB()` method of the `TSONEApp` class.

- *updateTL()* updates the database with the new trust level attribute for a network device. The method takes in the network device as an argument parameter. This method is invoked from the `TSONEApp` class.

6.2.3 ReadSyslog

This class is used to extract different attributes from the Syslog file. The Syslog file has the following attributes as shown in Figure 6.2: *Date/Time, Facility, Severity, IP Address and Event Details*. The `ReadSyslog` class has the following methods that extract parts of the events in the Syslog file for storage in the database.

- *open()* and *close()* are methods used to open and close an instance of the Syslog file for reading.

- *viewSyslog()* loads the current Syslog file in a text area on the GUI for the network administrator to peruse the file. This method is invoked from the `viewSyslog()` method of `TSONEApp` class.

- *getRouters()* loads the IP addresses of the network devices as an array from the logged events in the Syslog file. These addresses are then displayed in a non-repeating manner on the GUI for access by the network administrator. This method is invoked from the `load()` method of class `TSONEApp`.

  ```
  Enumeration e = ht.elements();
  String [] iparray = new String[ht.size()];
  for (int i = 0; i < ht.size(); i++){
    iparray[i] = (e.nextElement()).toString();
  }
  return iparray;
  ```

- *getFirstLog()* gets the date and time of the first event logged to the Syslog file. This is used in tandem with `getLastLog()` to determine how long a network device has been in the network environment.

- *getLogCount()* gets the number of events logged by a network device in the Syslog file.
- *getLog()* extracts date/time, facility and IP address of an event logged in the Syslog file by a network device. This method has an IP address as an argument parameter.

The interaction between the classes explained above is illustrated in the sequence diagram in Figure 6.4 below.

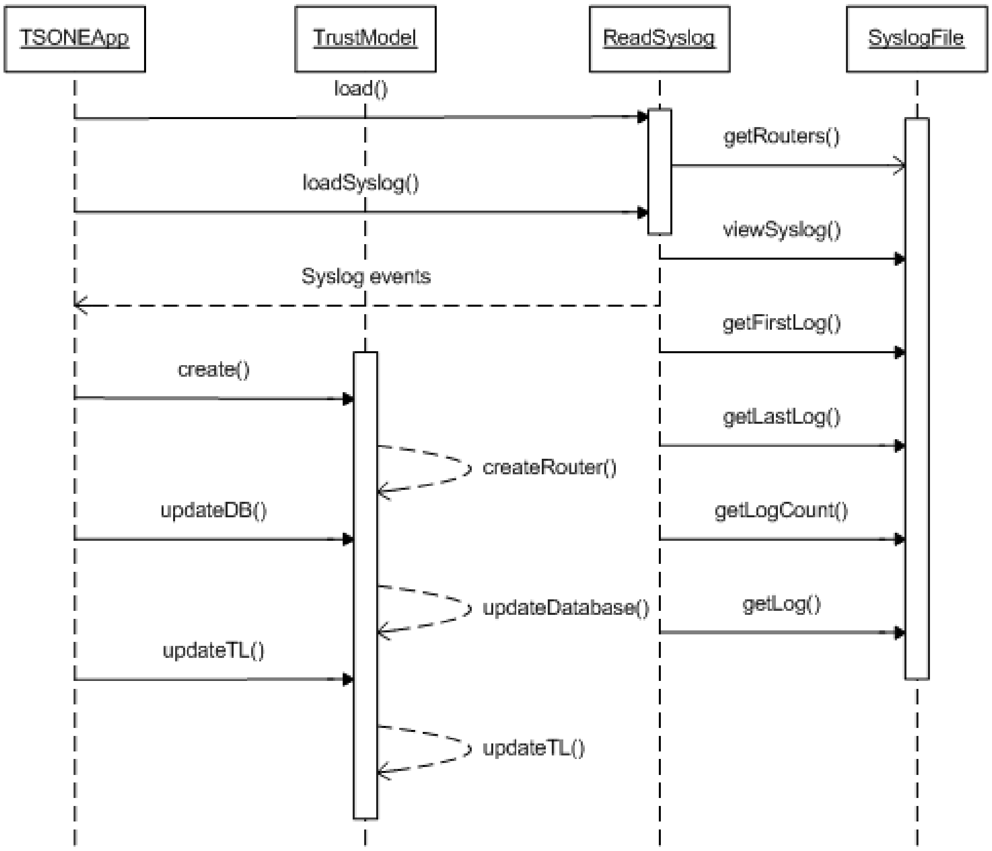


Figure 6.4: Sequence Diagram of a TSONE model

6.3 Lab Environment

The goal of this prototype is to implement a trust model and ultimately to determine the trustworthiness of a routing device. The classes created for the prototype of the trust model and network environment aim to simulate the activities in this environment. This environment was simulated based on the activities of the actors in a TSONE model as indicated in the Use Case diagram in the previous chapter. The actors are the routing device in the network environment, network administrator interacting with the network management server (NMS) and the trust model. Their activities are depicted in the activity diagram in Figure 6.5.

Figure 6.5 has three partitions for the functions of the actors. The following sections will give a detailed explanation of the test environment of the actors.

6.3.1 Routing Device in the network environment

To model a network environment, routing devices are needed to capture the activities in this environment. These devices would need to be connected together, must be able to communicate and must be able to report or log their activities to a central location.

The author of this dissertation approached Cisco Systems in South Africa to assist with this research and they obliged by providing routing devices to set up a network environment. These devices come with the proprietary Cisco IOS (Internetwork Operating System) software. The software and modes of access on these devices have become the de facto access methods in terms of command-line shortcuts for routing device software.

Cisco also donated two routers (Cisco 1601R Series Cisco 1760 Series) and a switch (Catalyst 4200 Series). The routers are connected as shown in Figure 6.6.

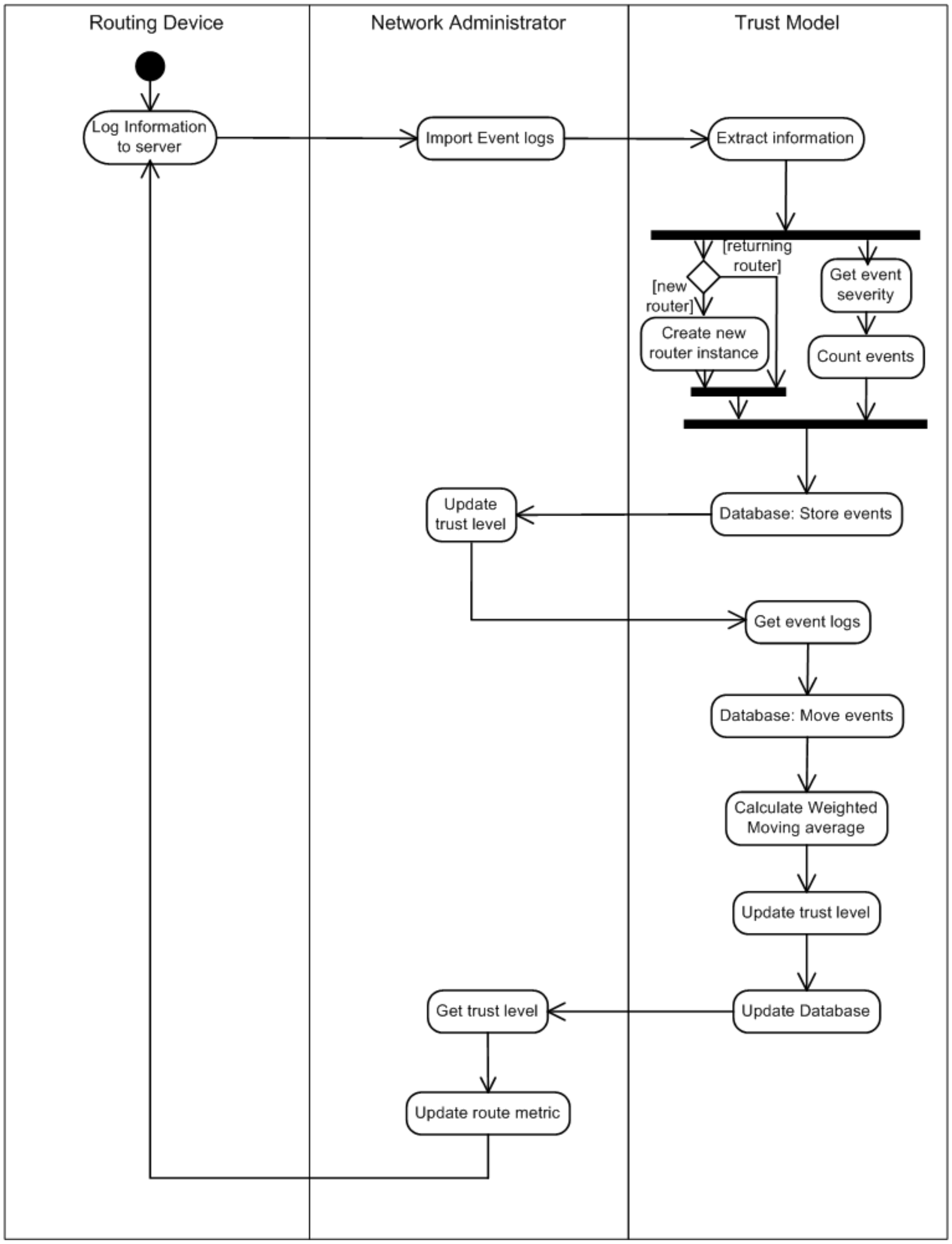


Figure 6.5: Activity Diagram of a TSONE model

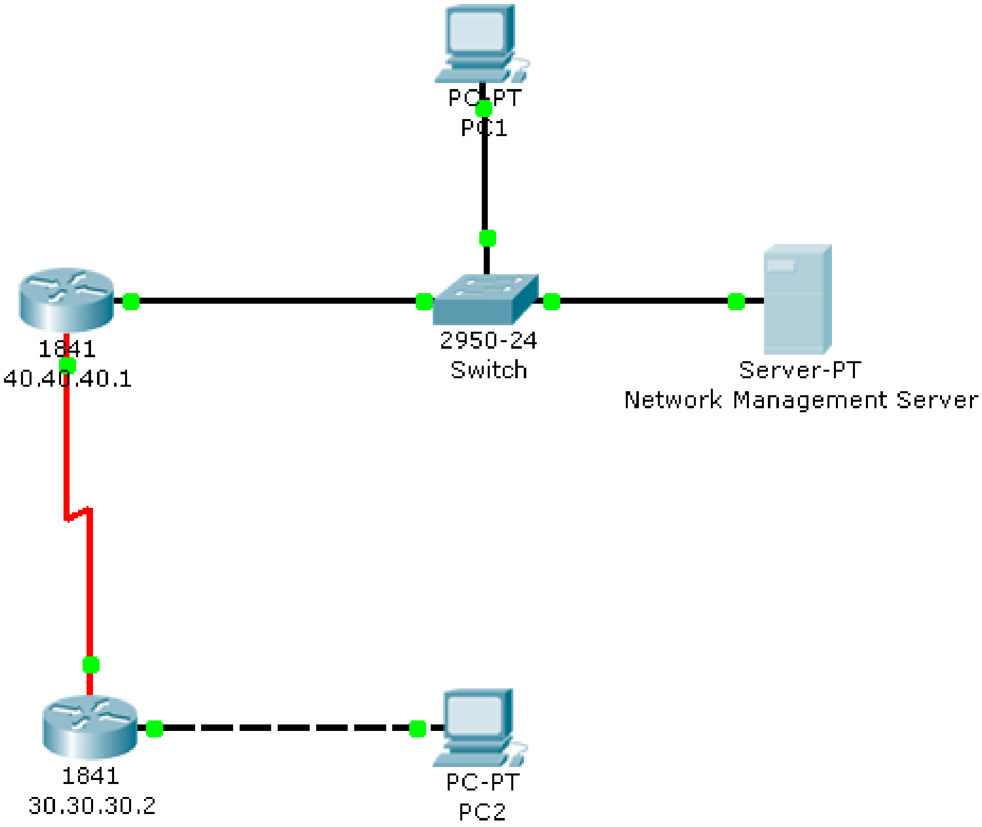


Figure 6.6: TSONE Network Environment

The routing protocol used in the environment illustrated in Figure 6.6 is the open shortest path first (OSPF) routing protocol. With the routing protocol used the network could be divided into areas. *Area 0* is controlled by the Backbone Router, while network traffic into *Area 1* goes through the Area Border Router. The network management server (NMS) is not a specific area because it would need to manage the different areas autonomously. The network is further divided into subnetworks namely: `30.30.30.0/24`, `40.40.40.0/24` and `20.20.20.0/24`. The aim of the division is to separate the network into two different areas. The areas are configured on the switch, while the routers and the terminals are configured separately with their spe-

cific areas as follows:
40.40.40.1 - Backbone Router
40.40.40.10 - User Terminal 1
30.30.30.2 - Area Border Router
30.30.30.3 - User Terminal 2
20.20.20.10 - Network Management Server

The user terminals are used to generate random traffic between the different subnetworks and areas.

The device code on the routers and switch (1841 and 2950-24 respectively) in Figure 6.6 can be ignored as they are used only to identify the type of equipment used.

6.3.2 Network Adminitsrator

In Figure 6.5, the network administrator block can access the Syslog messages, read and write to the database and interact with the network device via the network management server (NMS). The TSONE application (TSONEApp) and third party software applications are invoked from the network management server (NMS). To communicate with a network device a computer system needs an interface like a software application to interpret communication protocol messages. A brief overview of the applications on the NMS is given below.

- **Kiwi Syslog** [9] is a third party application that acts as an interface between a network device and a server. The application uses the simple network management protocol (SNMP) to accept Syslog messages from a network device. It can be installed in two modes: as a Windows service or as an interactive Windows application. In Windows service mode the user can leave the application running in the background while not being logged on the system. This is the other way round with the interactive installation mode, since Kiwi Syslog allows Syslog messages to be processed in various ways:

- Display Syslog message in real-time in a scrolling window
- Log Syslog message to a text file
- Forward the message to another application
- Log to a database
- Alert a network administrator via SMTP or a short message service (SMS) of a high severity message

For the purposes of this research, Syslog messages were logged to a text file called `Syslog.txt` on the NMS. Kiwi Syslog was the only application that was given write access to the file and TSONEApp was given read access to the file. That is, no other application had permission to read from or write to the Syslog file.

- **HyperTerminal** [4] is a terminal emulation application that is capable of connecting to systems through TCP/IP Networks, dial-up modems and communication (COM) ports. Hyperterminal is a lightweight application from Hilgraeve Inc. packaged with the Windows Operating System. This application is used to connect to the routing devices. The configuration and setting up these devices were done via the Hyperterminal. The lightweight version of this application does not allow simultaneous connections to the routing devices therefore communication with the routing devices has to occur one at a time.

- **MySQL** [6] is an open source relational database management system (RDBMS). This DBMS can implement several databases and run as a server for simultaneous access on a system. MySQL was the preferred

DBMS because it was under the free software licence, GNU General Public License. MySQL provides an uncomplicated implementation for users that are looking for a simple database implementation. The DBMS also uses multiple storage engines which allow users to choose how tables are stored in their database based on the amount of transactions and simultaneity of transactions. The TSONEApp trust model's database is implemented with MySQL. The tables used are discussed in the trust model subsection.

- **TSONEApp** is the prototype application written for this research. The TSONEApp can be invoked as an executable application. At the heart of this application is a trust model used to determine the trustworthiness of a network routing device. The application was written in the Java programming language. The NMS has a Java Runtime Environment (JRE) that is needed to start this application.

6.3.3 Trust Model

As explained in preceding chapters, a subset of the network trust opinion (Ntropi) [12] trust model was implemented in this project.

To capture routing devices' activities and use these activities to determine devices' trustworthiness require a consistent storage area such as a database. MySQL was used as the DBMS. An entity relationship diagram for the database is shown in Figure 6.7.

The database tables and their uses are explained below.

- **counter** is a table for an index pointer. This table contains the number of events in the Syslog file and is used for updating the database when there are new events in the Syslog file.

- **logdetail** contains the events of the Syslog file, except the message

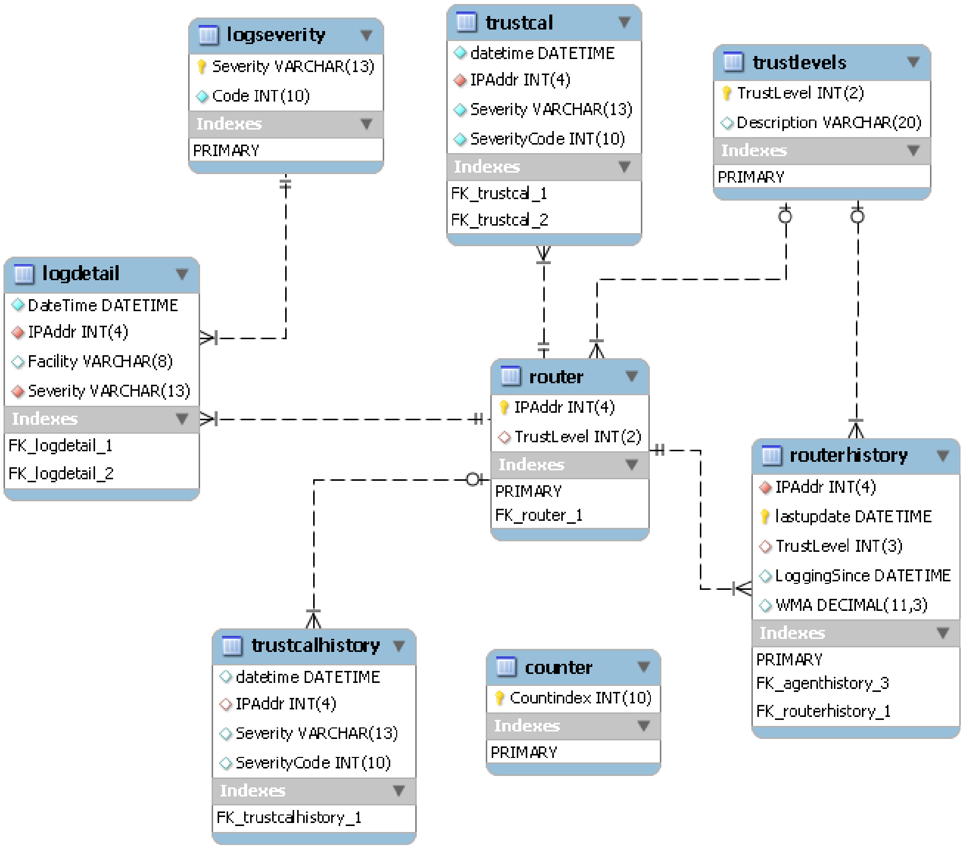


Figure 6.7: ERD for database

detail. These events are stored here before the trust level and phase values are calculated.

- **logseverity** stores the severity levels as given by the Syslog RFC [58].
- **router** stores the routing device information such as: IP address, phase value and trust level.
- **routerhistory** contains the history of a routing device. This includes previous calculation values such as: trust level, phase values, running average and median.

- **trust** contains the trust level values and their corresponding descriptions.
- **trustcal** stores log events temporarily for events calculation. That is, events are moved from `logdetail` and stored here temporarily while calculation is done on the severity values.
- **trustcalhistory** stores the historical log events after calculation. That is, after calculation has been done on the events in `trustcal` the events are moved to `trustcalhistory` and deleted from `trustcal`.

6.4 Conclusion

This chapter described the design of the TSONE model and the lab environment that led up to the implementation and testing in the next chapter. The goal of this prototype is to determine whether it is possible to assess the trustworthiness of a routing device. A trust model has been identified as a possible way to collect past activities of a routing device. These devices and information regarding their activities are collected and used to determine their trust level. The next chapter describes how the event logs are used to compute the trust level.

Chapter 7

Computing and Updating Trust Level

7.1 Introduction

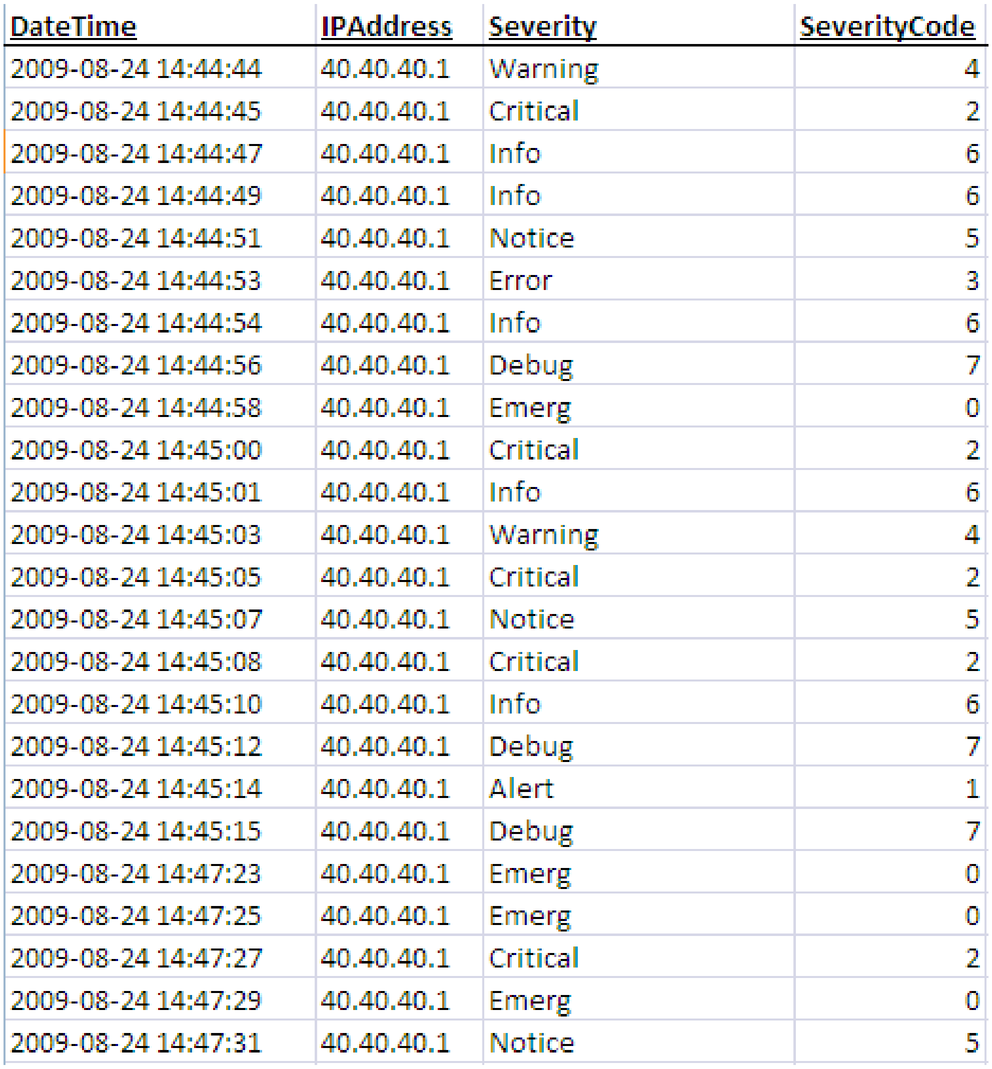
In the previous chapter the detailed design of TSONE was presented. Trust Level was introduced in Chapters 5 and 6 as part of the data structures adopted from the Ntropi [12] reputation-trust model. Trust level can be determined or calculated in various ways and the calculation depends on the environment the trust model is implemented for. The following section (7.2) gives an explanation of trust level as used in a TSONE. Section 7.3 elaborates on the detailed application of statistical functions to effect a change on trust levels. The impact of the trust level on a routing device is discussed in Section 7.4 while the chapter is concluded in Section 7.5.

7.2 Trust Level

Trust level is defined in Ntropi [12] as a scale to represent trust. The trust scale is labelled as: *Very Trustworthy, Trustworthy, Moderate, Untrustworthy* and *Very Untrustworthy* and it is represented numerically as *+2, +1, 0, -1* and *-2* respectively. The trust level scale is ordinal but the difference between two trust levels in the scale is mathematically undefined [12]. That is, the difference between +2(*Very Trustworthy*) and +1(*Trustworthy*) is *not* the same as between -1 (*Untrustworthy*) and -2 (*Very Untrustworthy*). The labels for the trust level are merely placeholders to facilitate the calculation of trust values.

The above trust level scales have been adapted to represent trust level calculation in our proposed Trusted Service-Oriented Network Environment (TSONE). Trust level is used in TSONE to indicate the trustworthiness of a network device as a service provider. To assess the trustworthiness of network devices, events from a network device(also known as, Syslog) are collected in a file. The parts or columns of the Syslog file were explained in Chapter 5 and those parts used in trust level determination include *DateTime, IP Address, Severity* and *Severity Code*. An example of the Syslog taken from one of our test routing devices is shown in Figure 7.1.

The IP address in the Syslog file example shown in Figure 7.1 is one of the routing devices used for this project's proof of concept. The test network consists of two network devices as described and shown in Chapter 6. The severity of Syslog messages is mapped to a numeric value also known as the severity code. The severity code is used to determine trust levels. A graph depicting the severity code mapped against the date and time for each



DateTime	IPAddress	Severity	SeverityCode
2009-08-24 14:44:44	40.40.40.1	Warning	4
2009-08-24 14:44:45	40.40.40.1	Critical	2
2009-08-24 14:44:47	40.40.40.1	Info	6
2009-08-24 14:44:49	40.40.40.1	Info	6
2009-08-24 14:44:51	40.40.40.1	Notice	5
2009-08-24 14:44:53	40.40.40.1	Error	3
2009-08-24 14:44:54	40.40.40.1	Info	6
2009-08-24 14:44:56	40.40.40.1	Debug	7
2009-08-24 14:44:58	40.40.40.1	Emerg	0
2009-08-24 14:45:00	40.40.40.1	Critical	2
2009-08-24 14:45:01	40.40.40.1	Info	6
2009-08-24 14:45:03	40.40.40.1	Warning	4
2009-08-24 14:45:05	40.40.40.1	Critical	2
2009-08-24 14:45:07	40.40.40.1	Notice	5
2009-08-24 14:45:08	40.40.40.1	Critical	2
2009-08-24 14:45:10	40.40.40.1	Info	6
2009-08-24 14:45:12	40.40.40.1	Debug	7
2009-08-24 14:45:14	40.40.40.1	Alert	1
2009-08-24 14:45:15	40.40.40.1	Debug	7
2009-08-24 14:47:23	40.40.40.1	Emerg	0
2009-08-24 14:47:25	40.40.40.1	Emerg	0
2009-08-24 14:47:27	40.40.40.1	Critical	2
2009-08-24 14:47:29	40.40.40.1	Emerg	0
2009-08-24 14:47:31	40.40.40.1	Notice	5

Figure 7.1: Example of Syslog file

routing device is shown in Figure 7.2.

Figure 7.2 also illustrates how the severity code changes per events are reported in the Syslog file. The following section elaborates on how the trust level is calculated in a TSONE environment.

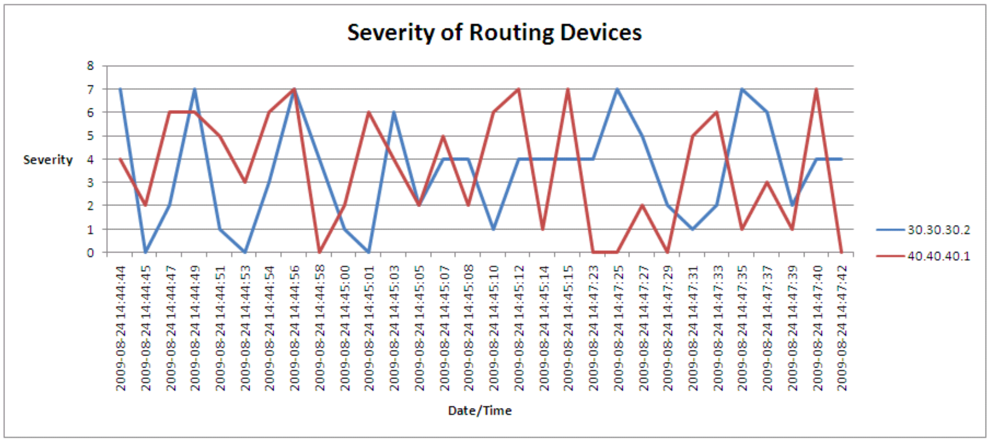


Figure 7.2: Severity of Routing Devices

7.3 Determining Trust Levels

Computing trust is a difficult area [13] because trust is not tangible but based on qualitative attributes. These attributes are based on intuition. In a virtual environment it is impossible to model intuition - therefore levels of trust are defined. Trust levels, in their use, are application specific. That is, they are defined for the environment they are applied in.

To compute trust in TSONE, the Ntropi trust model was used and most of the data structures were adapted for the TSONE environment. Trust levels are affected when events occur that either increase or decrease the trust level in a routing device. Given the service-provision environment, services could only be measured based on events that occurred on the routing device. These events are collected in a Syslog file and assigned a priority (severity) and a corresponding numerical value to indicate the severity of the event. The severity level scale is reproduced in Table 7.1.

Two methods to computing or determining a trust level were considered. The first is a *direct observation* of the changes in severity code (i.e. a one-to-one mapping of the severity codes to the trust levels). However, this does not

Code	Severity
0	Emergency: System is unusable
1	Alert: Action must be taken immediately
2	Critical: Critical conditions
3	Error: Error conditions
4	Warning: Warning conditions
5	Notice: Normal but significant condition
6	Informational: Informational messages
7	Debug: Debug-level messages

Table 7.1: Severity level scale [58]

represent the true distribution of the changes in severity code for a specific time period. The second approach, *Statistical functions* is a more favourable approach to computing a trust level by using descriptive statistics. This branch of statistics includes, among many others, the ability to calculate the central tendency of the severity codes.

Given a set of severity codes, their central tendency is an indication of the central value calculated for a specific period of time. This can be calculated by using any one of these methods:

- **Median**: This is the middle value that separates two halves of a data set.

- **Simple Moving Average (SMA)**: This is the same as calculating an average. It is "moving" because it is calculated over a moving time series of a data set.

- **Running Average (RA)**: This is similar to SMA however the average for the past and current time series is added together to arrive at a new average.

- **Weighted Moving Average (WMA)**: This type of average also includes previous data in its calculation. More weight is given to the most recent data.

The median calculation, unlike the SMA, RA and WMA, does not include the outliers of the severity code. In other words, end values are not considered but only the mid-point values. Since all the severity codes need to be considered to arrive at a suitable central value the median calculation is eliminated from further investigation. The following sub-sections give an explanation of SMA, RA and WMA in the context of the TSONE environment. Events collected from the lab environment are used to explain these methods.

7.3.1 Simple Moving Average

A simple moving average is similar to computing an average over a specific period of time. This is done by adding up the data for the time period and dividing it by the total number of time slots for the period observed. The formula to calculate SMA is given as follows:

$$\bar{s}_t = \frac{s_1 + s_2 + ... + s_n}{n} \tag{7.1}$$

Where $\bar{s}_t$ is the SMA for time period t and s is the severity codes for the observed events in that time period.

For example, an SMA can be calculated on the severity code for a random number of events. The sum of 55 events' severity code is *182*, therefore the SMA for 55 events is 3.3090. An example of SMA calculations is shown in Figure 7.3. The illustration shows the SMA in tandem with the severity code observations for one of the routing devices.

Computing average for a series of data tends to show the trend or central tendency of the data. The graph above illustrates the *general* trend of the

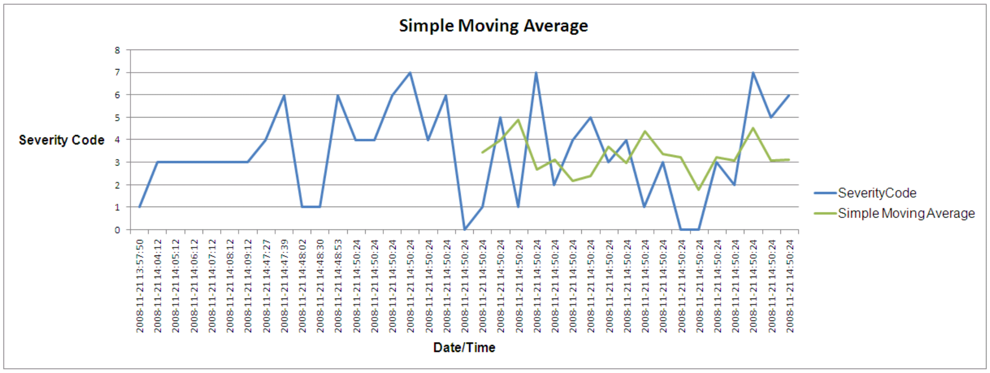


Figure 7.3: Simple Moving Average

severity code for the past time period. The above example of SMA calculation is intended to demonstrate how an SMA is calculated and how past events constitute the only important factor for calculating this type of average. The running average (RA) is explained below.

7.3.2 Running Average

Moving average, as defined for this environment, calculates the average of the observed events for the entire time period. In other words, past events are included in the calculation of the current observed events. Therefore the total number of events at any particular time is the sum of all events in the lifespan of the routing device. The formula to calculate WMA is shown below.

$$\bar{r} = \frac{\bar{r}_{t-1} + \left(\frac{s_1+s_2+\ldots+s_n}{n}\right)}{2} \tag{7.2}$$

Where $\bar{r}$ is the running average over the entire period and $\bar{r}_{t-1}$ is the running average from the previous calculation. For example, to follow from the SMA calculation 40 random events are taken after observing the 55 events. The sum of the 40 random events is 155 and the average of these events is

3.7804. The resulting running average is 3.547.

A graph depicting running average calculations is shown in Figure 7.4.

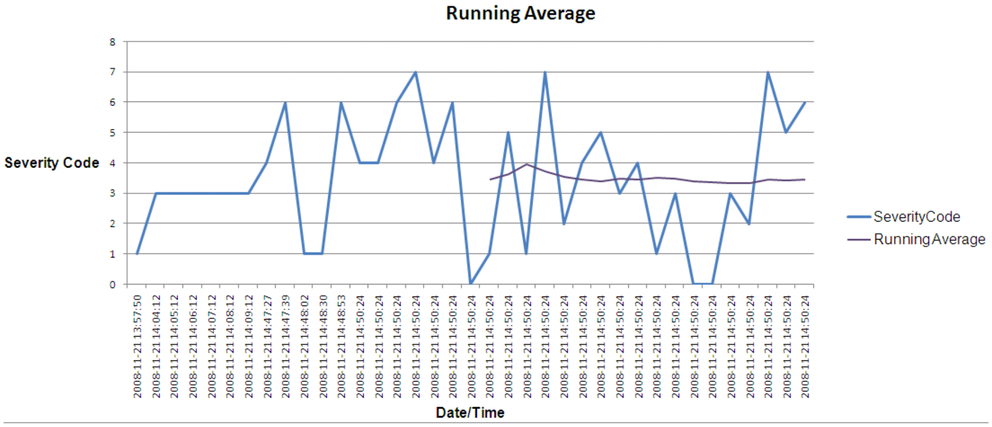


Figure 7.4: Running Average

7.3.3 Weighted Moving Average

A weighted moving average (WMA) is a means of smoothing random fluctuations by putting a declining weight on older data [43]. WMA, also known as exponentially weighted moving average, is used for trend forecasting and stock price data analysis in a financial environment. Weighted average assigns lesser importance to past data and assigns a greater weight to the more recent data by multiplying these values with a constant between 0 and 1.

Calculating a Weighted Moving Average

The formula to calculate the WMA has been adapted to the TSONE model and is given as follows:

$$\bar{w}_t = [c * (s - \bar{w}_{t-1})] + \bar{w}_{t-1} \tag{7.3}$$

Where $\bar{w}_t$ is the weighted average for time period t, c is the multiplier constant, s is the most recent event severity code and $\bar{w}_{t-1}$ is the previously calculated WMA. The constant c is given as follows:

$$c = \frac{2}{n+1} \tag{7.4}$$

n is the number of events for a specified time period.

To calculate the WMA on the severity code, a random number of events between 20 and 100 are taken from the Syslog file. For example, the first WMA calculation will not have all the variables required such as, the previously calculated WMA ($\bar{w}_{t-1}$). Therefore, a simple average of the first set of events is taken as WMA in the first calculation. Hence, using the same 55 events in the SMA calculation average is 3.3090. This will then be the previous WMA ($\bar{w}_{t-1}$) for the next WMA calculation.

If 41 events are chosen for the next calculation the constant c for 41 random events will be 0.0476. Therefore WMA calculated for 41 random events with a recent severity code of 6 is:

$$[0.0476 * (6 - 3.3090)] + 3.3090 = 3.4371 \tag{7.5}$$

An illustration of the above example is shown in a graph in Figure 7.5. A snapshot of one of the routing devices' events is taken for a time period. The graph shows the WMA in tandem with the severity code observations.

7.3.4 Summary

The WMA calculation aimed to show the relationship between the previously calculated WMA and the current WMA. The constant (c) is the weight applied to the current WMA calculation so as to give previous severity codes less weight than current severity codes. Figure 7.6 shows a comparison between the WMA, SMA and RA. Unlike the SMA and the RA, the WMA

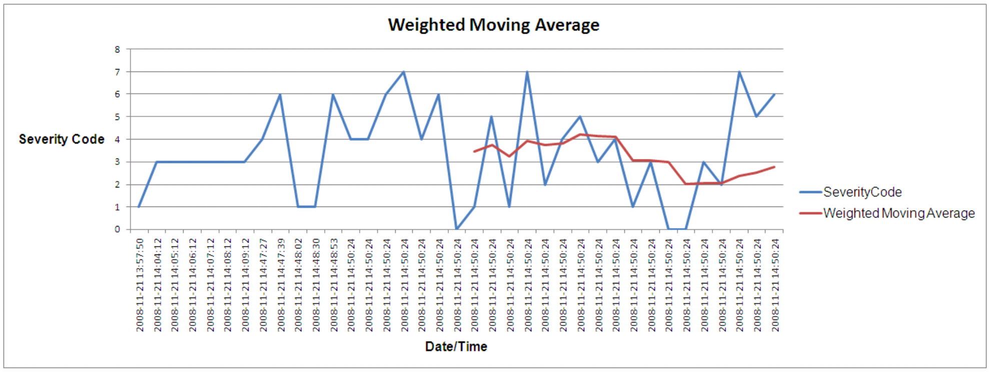


Figure 7.5: Weighted Moving Average

takes past and present data into consideration but less weight is given to the previously calculated WMA. The calculations are based on trend analysis, in other words, the general impression and central tendency of the events' severity code.

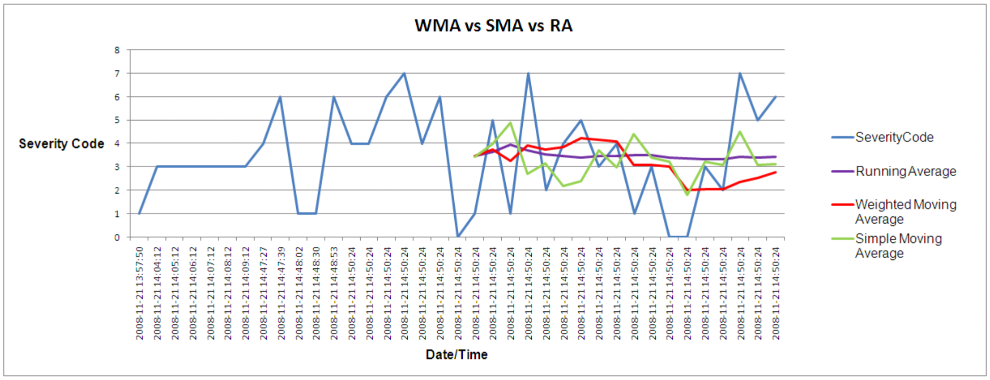


Figure 7.6: Weighted Moving Average vs Simple Moving Average vs Running Average

While the SMA shows the trend for the time period calculated the RA shows the trend for the entire time period. Since equal weight is given to the data, the RA line tends towards a straight line (Figure 7.4). While the WMA is sensitive to recent severity code changes past severity codes are also taken into consideration. For example, Figure 7.7 depicts the difference between WMA and SMA. The WMA calculation, shown as the red line in Figure 7.7, reacts differently to the recent severity code observation. In Figure 7.7, at point 'A' the severity code observation drops to 1, the WMA calculation reads 3.24 and the SMA calculation reads 4.90. The WMA gives more weight to the recent observation while also including past observations. The SMA calculation is the average for the observed period of time, hence the increase to 4.90 even though recent observation shows a decrease in severity code.

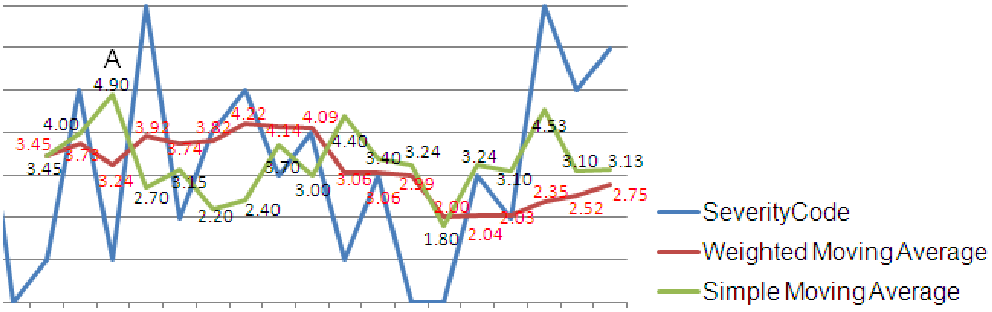
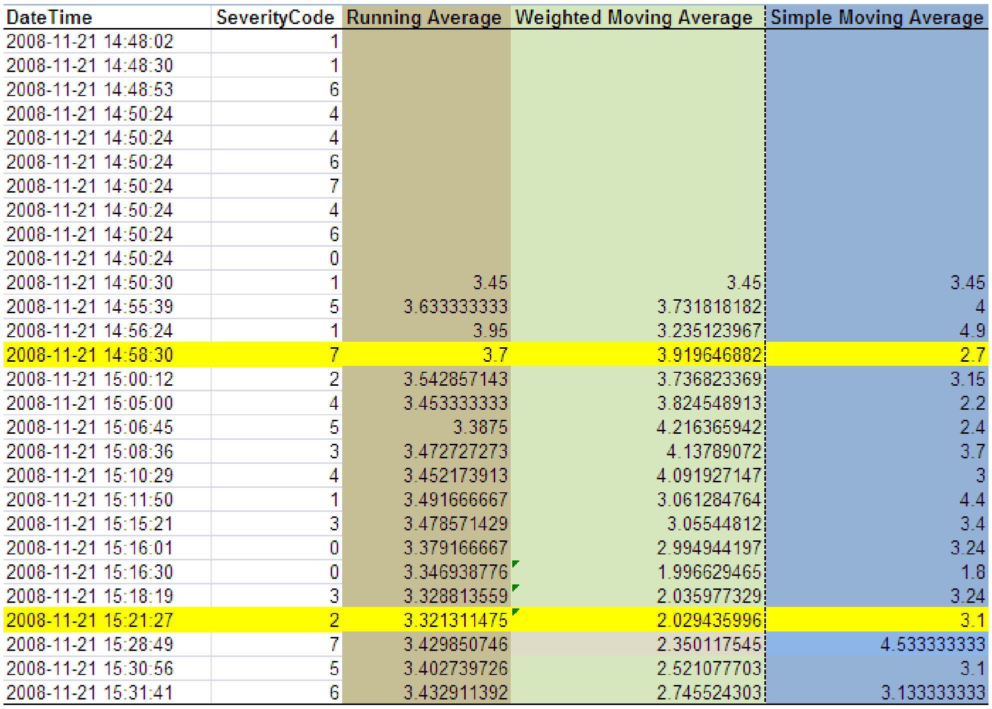


Figure 7.7: WMA average compared with SMA

To illustrate further, an extract of the Syslog events including calculations of RA, WMA and SMA is shown in Figure 7.8. The highlighted rows at Time '14:58:30' and '15:21:27' depict a drastic decrease and increase in severity codes from 7 to 2 and 2 to 7 as well. While there's a drastic change in the SMA calculation at Time '14:58:30' and the WMA calculation does not change as much. The RA does not change much either however it remains in the same range of values because its values are influenced by past average calculations. This makes RA tend towards a straight line. Although no significant change is noticed in the RA calculation at Time '15:21:27' either, a substantial change is noticed in the SMA calculation. The WMA changes accordingly while taking past values into consideration.



DateTime	SeverityCode	Running Average	Weighted Moving Average	Simple Moving Average
2008-11-21 14:48:02	1			
2008-11-21 14:48:30	1			
2008-11-21 14:48:53	6			
2008-11-21 14:50:24	4			
2008-11-21 14:50:24	4			
2008-11-21 14:50:24	6			
2008-11-21 14:50:24	7			
2008-11-21 14:50:24	4			
2008-11-21 14:50:24	6			
2008-11-21 14:50:24	0			
2008-11-21 14:50:30	1	3.45	3.45	3.45
2008-11-21 14:55:39	5	3.633333333	3.731818182	4
2008-11-21 14:56:24	1	3.95	3.235123967	4.9
2008-11-21 14:58:30	7	3.7	3.919646882	2.7
2008-11-21 15:00:12	2	3.542857143	3.736823369	3.15
2008-11-21 15:05:00	4	3.453333333	3.824548913	2.2
2008-11-21 15:06:45	5	3.3875	4.216365942	2.4
2008-11-21 15:08:36	3	3.472727273	4.13789072	3.7
2008-11-21 15:10:29	4	3.452173913	4.091927147	3
2008-11-21 15:11:50	1	3.491666667	3.061284764	4.4
2008-11-21 15:15:21	3	3.478571429	3.05544812	3.4
2008-11-21 15:16:01	0	3.379166667	2.994944197	3.24
2008-11-21 15:16:30	0	3.346938776	1.996629465	1.8
2008-11-21 15:18:19	3	3.328813559	2.035977329	3.24
2008-11-21 15:21:27	2	3.321311475	2.029435996	3.1
2008-11-21 15:28:49	7	3.429850746	2.350117545	4.533333333
2008-11-21 15:30:56	5	3.402739726	2.521077703	3.1
2008-11-21 15:31:41	6	3.432911392	2.745524303	3.133333333

Figure 7.8: Syslog events with Calculations

SMA changes drastically due to the limited time period used in the calculation. No significant change is noticed in the RA calculation because it is recurring, in other words, the average of previous calculations is used over again. However, the WMA calculations consider previous severity codes and accordingly assign a weight to them. While the SMA and RA have their usefulness in other application domains, these methods of calculating average are not useful with the current dataset. The WMA is consequently used in this project to determine the trend in severity code changes. The impact on trust levels is discussed in the next section.

7.4 Impact on Trust Level Determination

The calculations of the weighted moving average shown above identify trends in the severity code changes. That is, the impact of the WMA calculation on the trust level is related to the severity code changes. Since the WMA predicts the trend in severity code observations, there is a direct relationship between the WMA and the severity codes.

Hence, trust level is determined by matching the WMA results of the observed events' severity code to the corresponding trust level. This process is depicted in Table 7.2.

Trust Level	Severity Code
+2 (Very TrustWorthy)	WMA >= 5
+1 (Trustworthy)	5 < WMA > 4
0 (Moderate)	4 < WMA > 3
-1 (Untrustworthy)	3 < WMA > 1
-2 (Very Untrustworthy)	1 < WMA > 0

Table 7.2: Trust Level and Severity Code Comparison

According to Table 7.2, the trust level of the routing device calculated in the WMA example will be moderate because it is between 3 and 4.

The onus remains with the network administrator regarding the next step of action. The network administrator can either reconfigure the routing device for another network based on its trust level or increase the path cost associated with the routing device. Increasing the path cost for the routing device will deter the use of an untrustworthy routing device to a destination if a routing device with a much lesser path cost exists on the network. The network administrator could also ignore the trust level and make a decision based solely on the WMA for a particular timeslot.

7.5 Conclusion

This chapter set out to explain how trust level is calculated in a TSONE environment. As stated above, trust level determination depends on the environment the trust model was implemented for. For a TSONE environment where services provided by routing devices are essential severity of events reported to a Syslog file is crucial in determining the trustworthiness of a routing device. Statistical functions considered were the simple moving average, running average and weighted moving average. The weighted moving average is used in this project to determine the trust level in a routing device. The trust level of a routing device is based on the relationship between the severity code and the WMA. The following chapter presents the prototype to validate the TSONE concept.

Chapter 8

TSONE Prototype Implementation

If you're not scared or angry at the thought of a human brain being controlled remotely, then it could be this prototype of mine is finally starting to work.

– John Alejandro King, My War On Terror!

Every building is a prototype. No two are alike.

– Helmut Jahn

8.1 Introduction

In Chapter 5 the design of our Trusted Service-Oriented Network Environment (TSONE) model was discussed. The main functions of the model were identified in a Use Case Diagram and the components of the model were presented in a class diagram in Chapter 6. These components include the TSONE interface, trust model and syslog reader and their implementation constitutes the focus of this chapter. The following section briefly explains the aim of the prototype. Section 8.3 provides a description of the components that make up the prototype. A usage scenario is presented in Section 8.4 to illustrate the operation of the implementation. The conclusion is given in Section 8.5.

8.2 The Objective of the Prototype

The TSONE prototype aims to demonstrate the following features of the TSONE model:

- The use of a trust model for routing devices in a network environment.
- The automated computation of trust level for a routing device.
- The collection and use of past and current events to guide trust decisions.

The objective of the prototype is to implement the Ntropi [12] trust model so as to determine the trust level of routing devices. In order to simplify the TSONE model implementation, a subset of the data structure of the Ntropi trust model was used.

8.3 Implementation Description

The TSONE prototype can be used by a network administrator as a network monitoring tool in a network environment. The prototype monitors service provided by routing devices in the network. Events in the network are captured and stored in the Syslog file while the information of these events and routing devices is stored in the database. Everything is stored in the database as a central repository for the trust model to access information needed. The trust model uses the database to access routing devices' information and then uses the information acquired to calculate the corresponding trust level.

The prototype was developed on the Java platform and the database is based on the MySQL relational database management system.

For implementation purposes, the components explained below were implemented as Java classes. In order for the reader to understand the component names, each is differently than its Java class name. Hence, the TSONE Interface is implemented as *TSONEApp*, Trust Model as *TrustModel* and Syslog Reader as *ReadSyslog*. Figure 8.1 depicts the components. *TSONEApp* is the base class and implements the child classes– *TrustModel* and *ReadSyslog*. The database is exclusive to the *TrustModel* class, that is, the database is only read from and written to in the *TrustModel* class. The *ReadSyslog* extracts the relevant parts of the Syslog file. The following subsection discuss the components that make up the TSONE implementation.

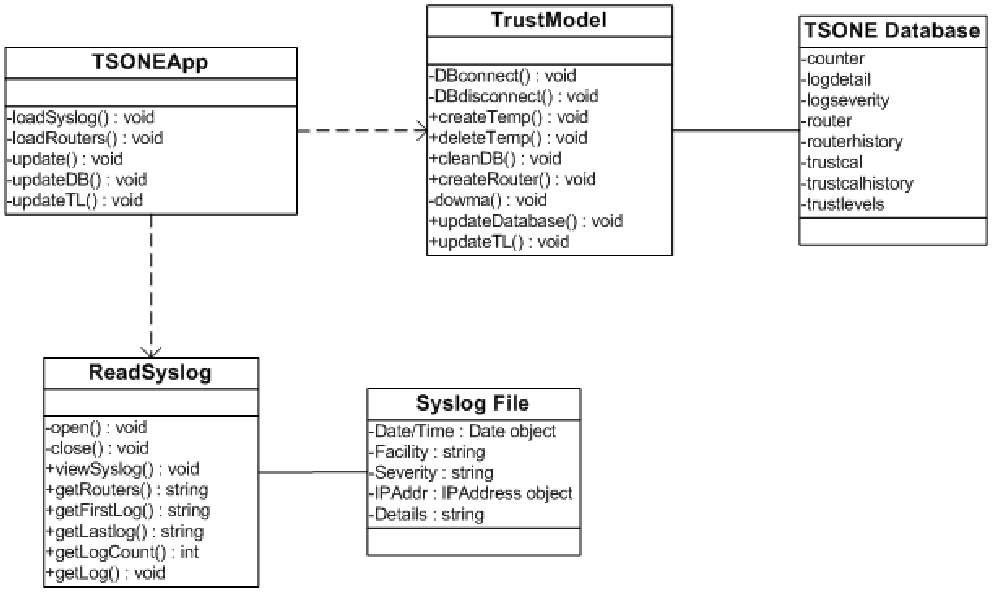


Figure 8.1: TSONE Components

8.3.1 TSONE Interface

The primary user of the TSONE system is a network administrator. The TSONE Interface (shown in Figure 8.2) serves as a point of entry into the system. The Trust Model and Syslog Reader components are triggered from the TSONE interface.

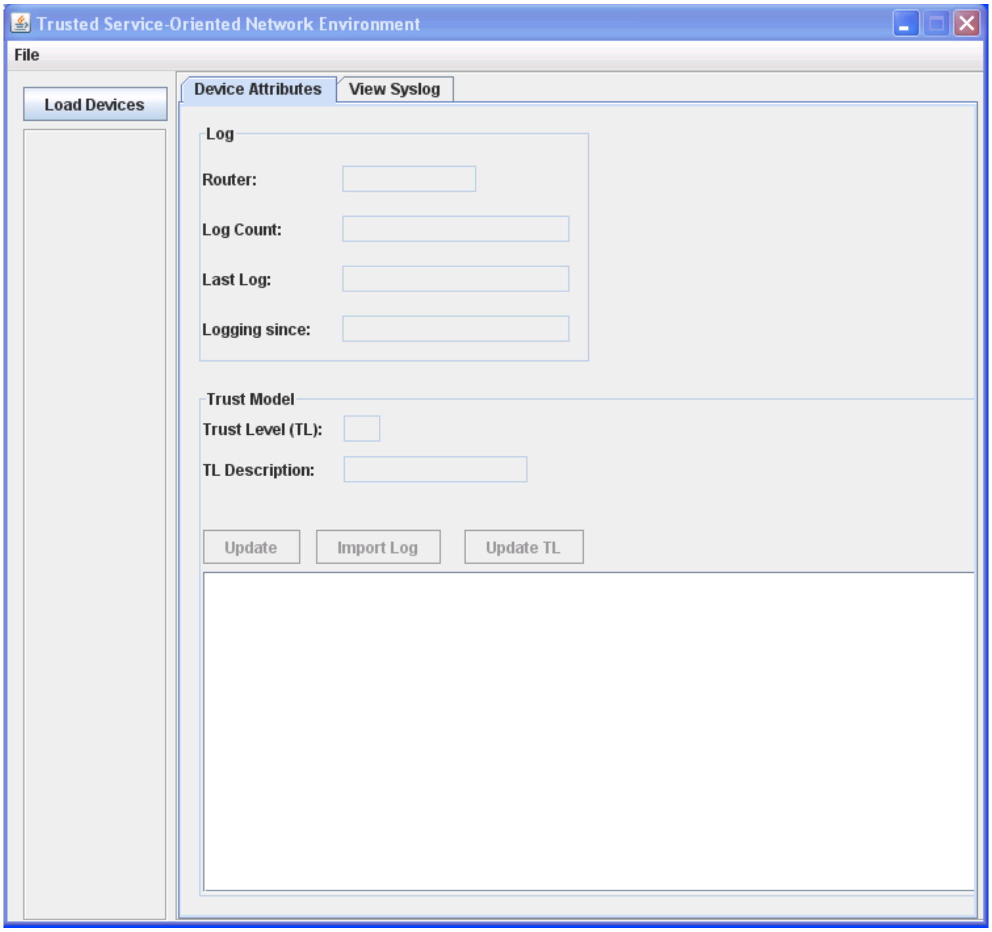


Figure 8.2: TSONE Interface

The TSONE interface, also shown in Figure 8.2, consists of a button to load the routing devices from the Syslog file. The device(s) are in-turn listed in the panel below the button. The Device Attributes tab shows the log information of the device that was selected in the left panel. The Syslog Reader component is invoked in the Log panel. Below the Log panel is the Trust Model panel. Trust attributes of the routing device selected above is displayed in this panel. The Trust Model component is invoked in the Trust Model panel.

The View Syslog tab, shown in Figure 8.3, loads the events in the Syslog file for viewing purposes. Since the events in the Syslog file change constantly, the network administrator can view the events in the file without interrupting the event writing process to the Syslog file. The Load Syslog button invokes *viewSyslog* from the Syslog Reader component.

8.3.2 Syslog Reader

The Syslog reader component extracts the logged event messages from the Syslog file. An event message saved in the Syslog file is made up of: date and time of the event, the facility that generated the event, the severity of the event, IP Address of the routing device from where the message originated from and the text of the message. Some parts of the event messages are not necessary to compute trust for a routing device, such as, facility and the full text of the message. The parts of the messages that are useful are extracted using the Syslog Reader component.

The useful part of an event message is extracted by using *regular expression (regex)*. Regex provides a functionality to match strings and patterns of characters. Since regex is written in a formal language Java provides a regex processor to interpret the formal language specification. Regular expression to extract IP address, date and time and facility and severity follows below:

```
String ipaddress = "(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2(?:[0-4][0-9]|5[0-5]))";
String dateTime = "(\\d{4})-(\\d{2})-(\\d{2}) [0-2][0-9]:[0-5][0-9]:[0-5][0-9]";
```

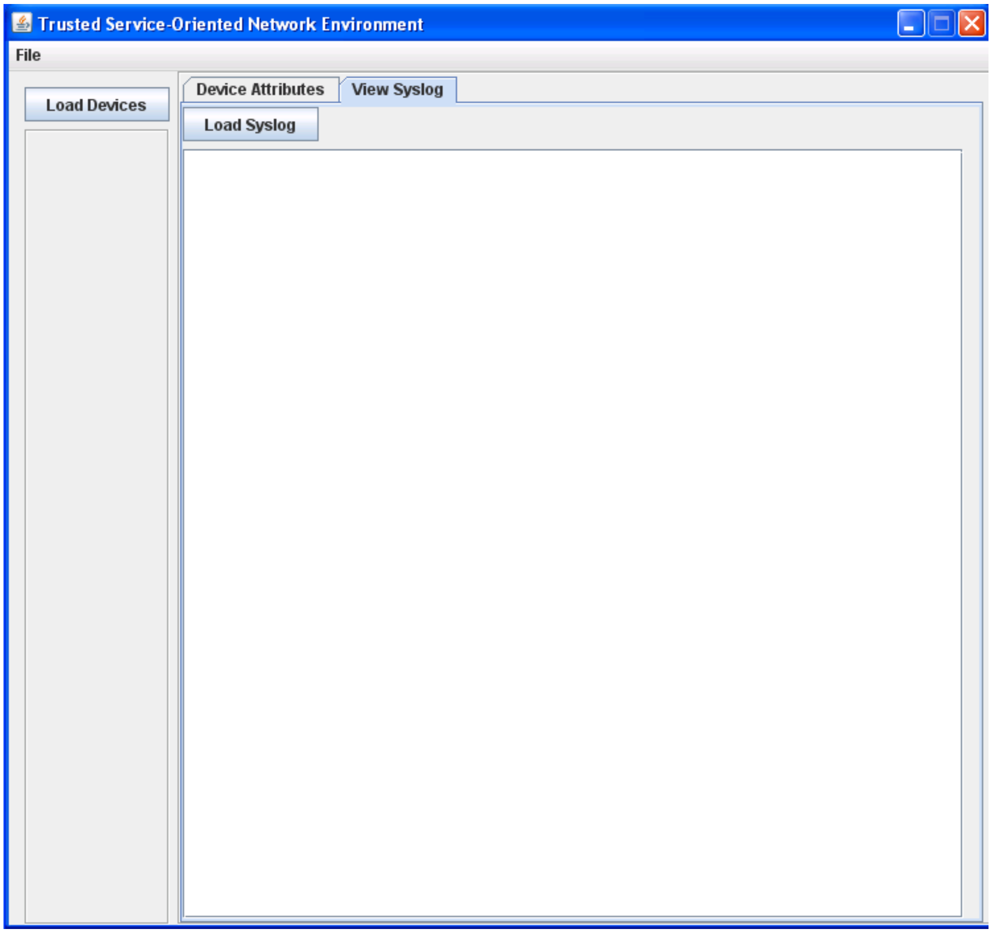


Figure 8.3: TSONE Interface – View Syslog

```
String facility_Severity = "\\w+\\d*\\.\\w+";
//Java's Regex Processor
Pattern IPPattern = Pattern.compile("^(?:"+ipaddress+"\\.){3}"+ipaddress+"$");
Pattern datePattern = Pattern.compile(dateTime);
Pattern facPattern = Pattern.compile(facility_Severity);
```

When the network administrator selects *Load Devices* the IP addresses of the routing devices in the Syslog file are listed in the panel below as shown in Figure 8.4.

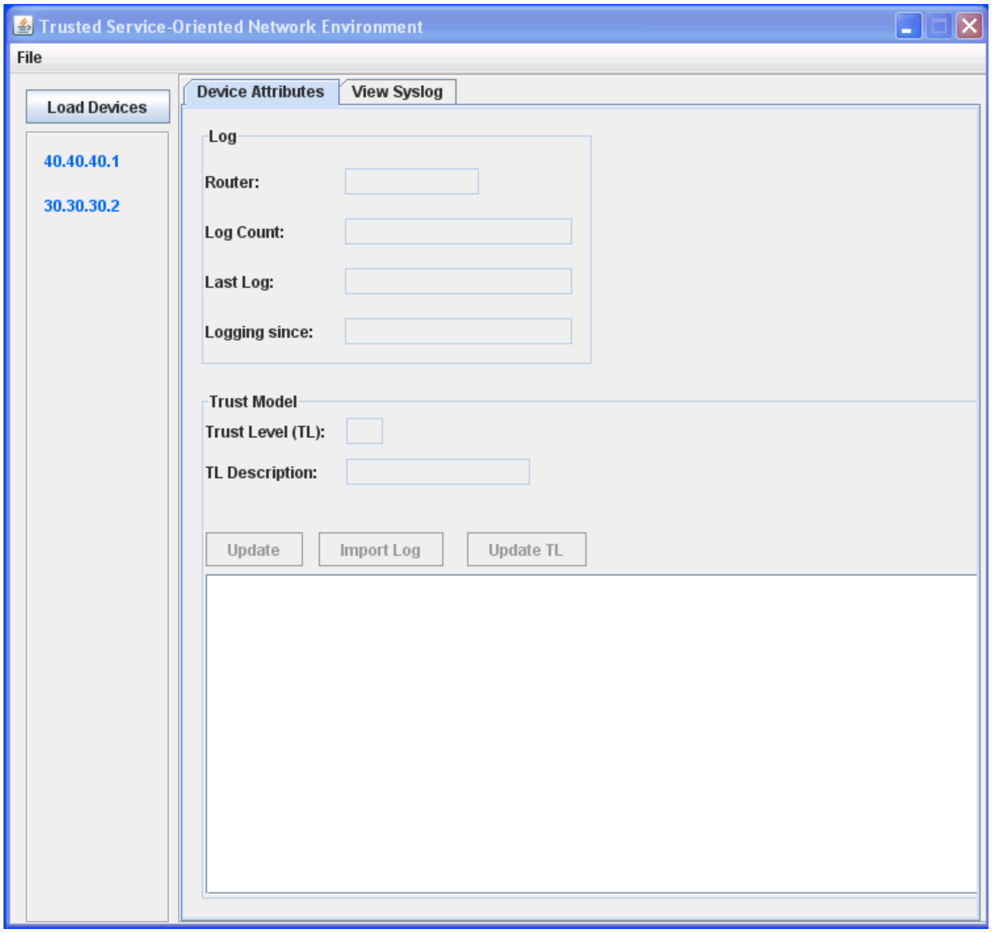
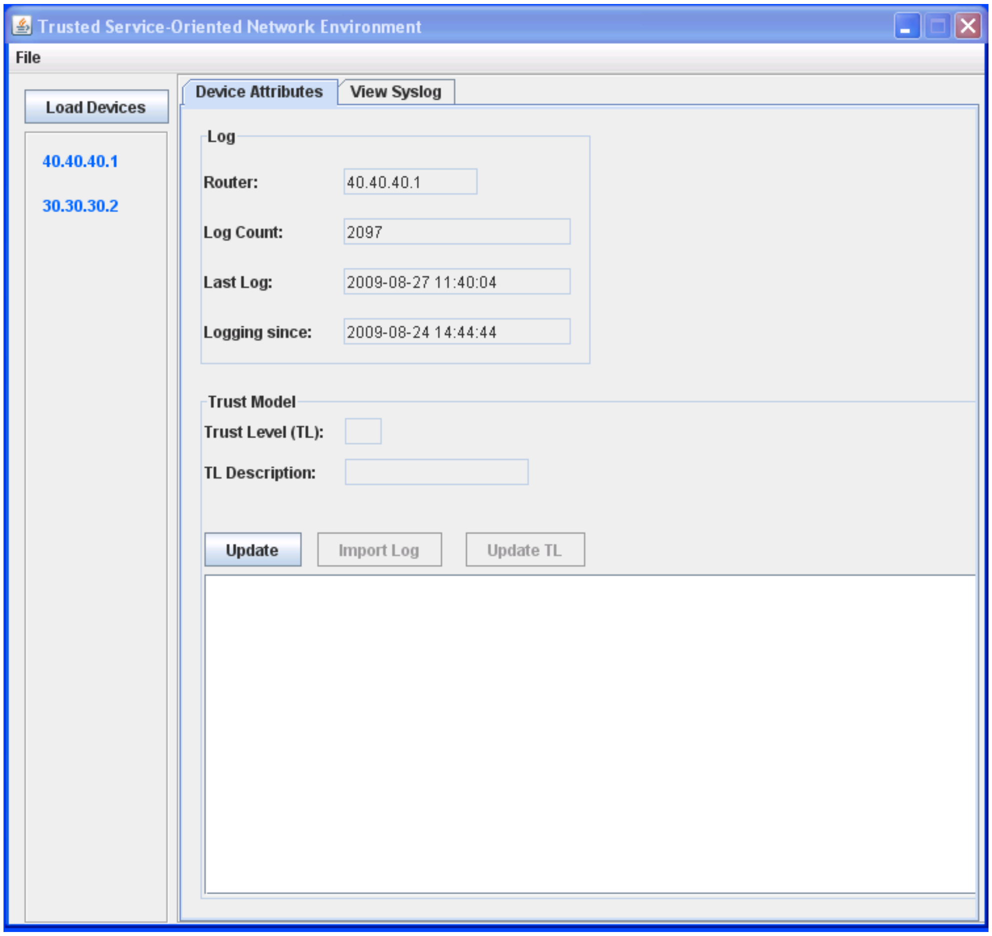


Figure 8.4: TSONE Interface – Load Devices

The IP addresses are extracted from the Syslog file. This is done to prevent a new router from appearing on the network. Routers in the TSONE environment are required to send their events to the Syslog file on the network management server (NMS). This requirement enables the TSONE Interface to manage a routing device, to determine trust level and view the events of a routing device.

The IP addresses of the routing device in Figure 8.4 are displayed as a link text object. This enables the network administrator to click on it and information about the router is available in the *Log* panel as shown in Figure 8.5. The Syslog file shows the routing device's details, the number of events generated by the routing device in the Syslog file, date and time of the last log event and the first log event in the Syslog file.



Router: 40.40.40.1

Log Count: 2097

Last Log: 2009-08-27 11:40:04

Logging since: 2009-08-24 14:44:44

Trust Level (TL):

TL Description:

Figure 8.5: TSONE Interface – Routing Device Details

8.3.3 Trust Model

The trust model is implemented as a class in Java. The trust level of routing devices is computed by invoking *updateTL* – a method of the trust model class used to update the trust level. To compute a routing device's trust level, an instance of the routing device is added to the database. Both the routing device related events extracted from the log file and routing device details are stored in the database. This is updated when a new trust level is computed and log events are updated.

The severity codes in the log events are used to determine the trust level. This is computed by using a statistical method known as the weighted moving average (WMA). This calculation uses the average of the severity codes of the events for the routing device. WMA takes previous calculations into consideration by applying a fractional weight to the previous trust level. The trust level is determined by matching the WMA to the trust level scale defined for Ntropi [12]. Abdul-Rahman also defined Phase Threshold in Ntropi for relationship phases that agents go through. Phase Threshold is not used in TSONE due to the autonomous nature of routing devices. In other words, routing devices are implemented by different equipment manufacturers and even though they are aware of each other via various routing and advertising protocols, it is impossible to observe their relationship with each other.

The database is the central repository for the trust model. It is implemented using the MySQL database management system (DBMS). MySQL is scalable and robust enough to handle the size of the database. Access and queries to the DBMS are implemented from the trust model class. The TSONE database (*TSONEDB*) is used to store relevant details from the logfile, maintain the trust level of a routing device and keep track of trust level computation for routing devices. Information given by TSONEDB can easily be corrupted if simultaneous writing by various processes occurs or if wrong information is presented. To prevent other processes from writing to the database at the same time connection to the database is opened for use and closed after use. If a connection remains open no process can write to the database.

8.4 Prototype Scenario

The scenario given below aims to demonstrate the use and operation of the prototype. The discussion focuses on determining a routing device's trust level, that is, computation of trust level and action taken after trust level determination. The action taken is network administrator dependent. Thus, action taken against a routing device with a low trust level is based on the network environment service provision policy. The creation of this policy is beyond the scope of this project however an example is provided of an action that could be taken.

The scenario is based on the test network environment set up for this project in Figure 8.6. The environment includes two routing devices, and for the purposes of this scenario one of the routing device's Syslog events have been fictitiously generated to decrease the trust level in the routing device. The discussion is structured as follows:

- The administrative interface of TSONEApp.
- Trust Model - Calculating the trust level for routing devices.
- Effect of a trust level on a routing device.

8.4.1 The administrative interface of TSONEApp

The basic functionality of the interface is explained below followed by the scenario development. The interface of the TSONE application (TSONEApp)

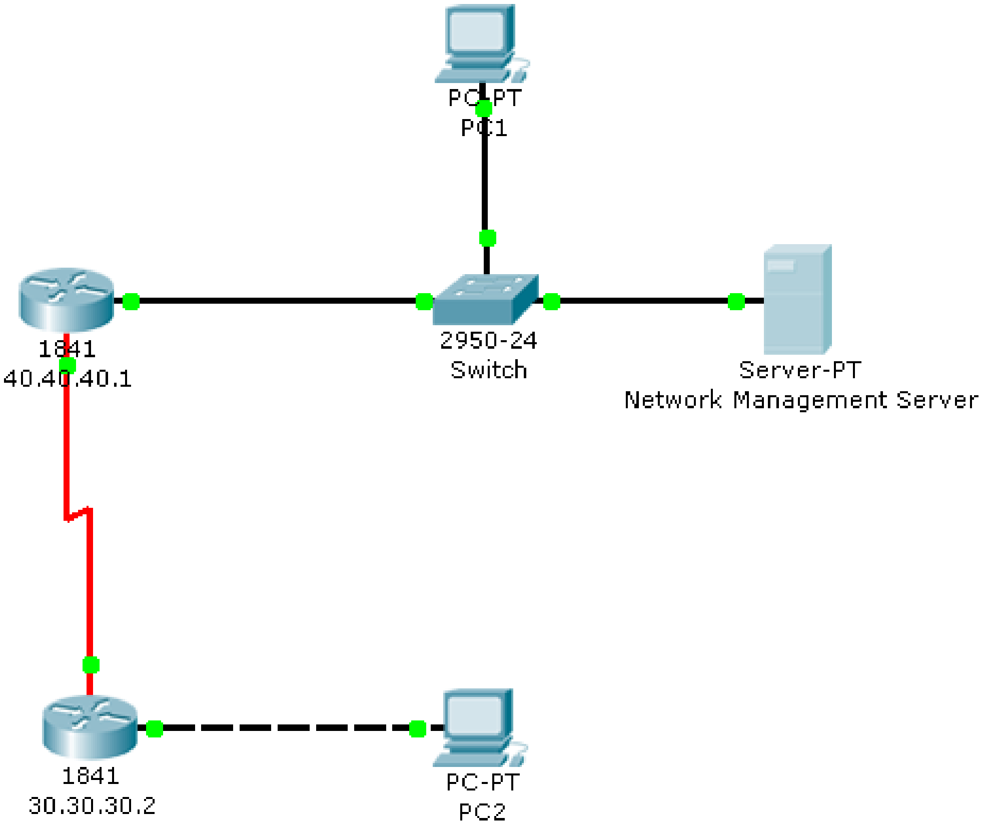


Figure 8.6: TSONE Network Environment

is shown in the figure below and it illustrates the administrative functionalities of the TSONEApp. The routing devices are loaded from the Syslog file by selecting the *Load Device* button. Loading the device from the Syslog file allows the network administrator to view information about the routing device such as, the number of events logged by the device and how old or new the device is on the network based on 'logging since' and 'last log'.

An instance of the routing device is created in the trust model database by selecting the *Update* button. If an instance of the router exists in the database, the *Update* function modifies the trust level and its description. The *Import Log* button either populates the database with the events in the log file or updates the database if new events have been added to the Syslog file.

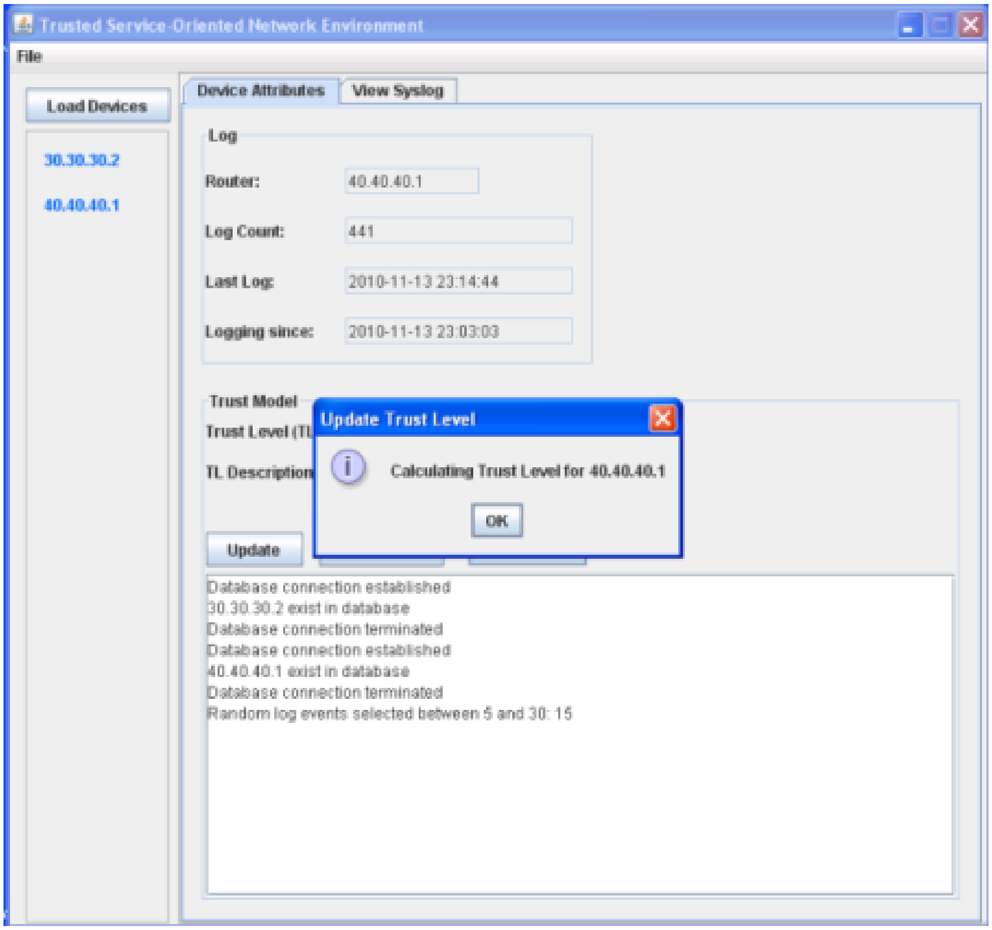
The *Update TL* button is used to compute the trust level for a routing device. The text area below the three buttons described above displays the status of an action invoked by the administrator. The *Load Syslog* button under *View Syslog* tab prints the Syslog file in the text area shown. The Syslog events are mainly for perusal purposes.

8.4.2 Computing the trust level for routing devices

For this discussion, it is assumed that both routing devices' (`40.40.40.1` and `30.30.30.2`) event logs have been imported into the trust model database. Thus, their trust level can be computed based on the logged events in the database. To determine the trust level for `40.40.40.1`, the administrator selects the *Update TL* button as shown in Figure 8.7. The application successively indicates that the trust level for the selected routing device is being calculated.

The result of the calculation is shown in the text area as depicted in Figure 8.8. A random number of log events between 5 and 30 are selected for calculation. The random number is between 5 and 30 because the weighted moving average (WMA) is more sensitive to lower numbers and less sensitive to larger values. Twenty-two (22) random log events for routing device `40.40.40.1` were chosen for calculation and the severity code of the last event in the selection is 6. The previously calculated WMA and the new WMA are also depicted. The severity code and the previous WMA are used in the WMA calculation. The new trust level is 0 which represents a *moderate* trust level.

The remainder of the information in the text area in Figure 8.8 generally indicates that the TSONE database is back in a consistent state. This infor-



Log

Router: 40.40.40.1

Log Count: 441

Last Log: 2010-11-13 23:14:44

Logging since: 2010-11-13 23:03:03

Calculating Trust Level for 40.40.40.1

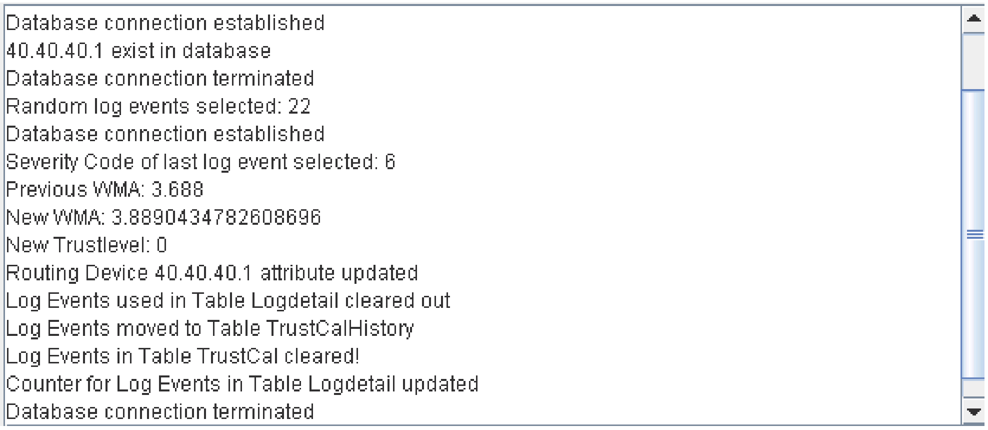
```
Database connection established
30.30.30.2 exist in database
Database connection terminated
Database connection established
40.40.40.1 exist in database
Database connection terminated
Random log events selected between 5 and 30: 15
```

Figure 8.7: TSONE Interface – Updating Trust level

mation is given to assure the network administrator that log events moved during calculation have been updated.

The second routing device, `30.30.30.2`, started out with a moderate trust level as shown in Figure 8.9. Even though the severity code of log events is 2, the WMA maintains a moderate trust level.

Figure 8.10 shows that the trust level of routing device `30.30.30.2`'s later dropped to reflect the severity code of the log events. As stated earlier ,the WMA gives a weight to previous WMA calculations along with attributes of



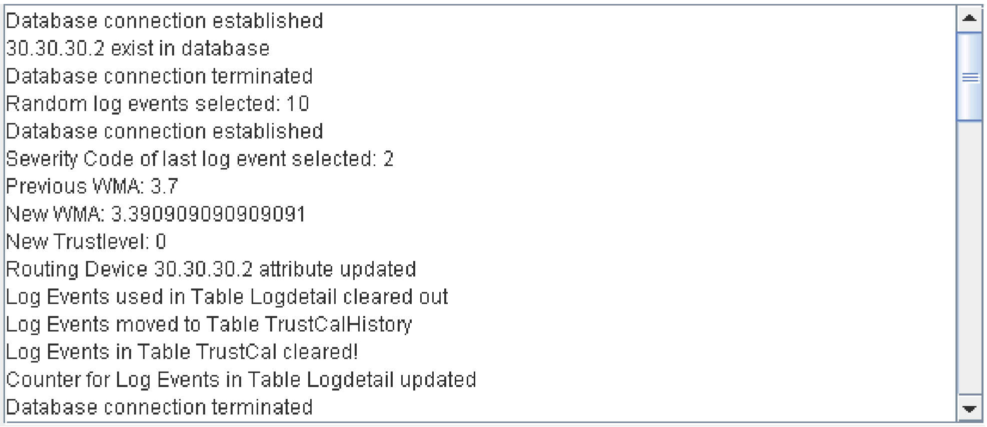
```
Database connection established
40.40.40.1 exist in database
Database connection terminated
Random log events selected: 22
Database connection established
Severity Code of last log event selected: 6
Previous WMA: 3.688
New WMA: 3.8890434782608696
New Trustlevel: 0
Routing Device 40.40.40.1 attribute updated
Log Events used in Table Logdetail cleared out
Log Events moved to Table TrustCalHistory
Log Events in Table TrustCal cleared!
Counter for Log Events in Table Logdetail updated
Database connection terminated
```

Figure 8.8: TSONE Interface – Result of Trust Level Calculation

the current WMA calculation.

According to Figure 8.10 twenty-one (21) log events relating to routing device `30.30.30.2` were taken to compute the new WMA. Before accepting the trust level given by the application, the network administrator might need to examine the log events that were used for this specific calculation. Trust calculation history (*trustcalhistory*) is a table in the database that stores past events used for WMA calculation. Log events in *trustcalhistory* for routing device `30.30.30.2` are shown in Figure 8.11.

Figure 8.11 illustrate the *trustcalhistory* table in the TSONE database that stores log events by router used for WMA calculations. The database converts the IP address of the routing device into a numeric integer of 4- or 8- byte addresses. This saves storage space compared to storing the IP addresses in string format. Thus, `30.30.30.2` is represented as 505290242 and `40.40.40.1` as 673720321. The log events have been manipulated to illustrate how the WMA would change given a low severity level. Hence only "Critical" events are in the log file.



```
Database connection established
30.30.30.2 exist in database
Database connection terminated
Random log events selected: 10
Database connection established
Severity Code of last log event selected: 2
Previous WMA: 3.7
New WMA: 3.390909090909091
New Trustlevel: 0
Routing Device 30.30.30.2 attribute updated
Log Events used in Table Logdetail cleared out
Log Events moved to Table TrustCalHistory
Log Events in Table TrustCal cleared!
Counter for Log Events in Table Logdetail updated
Database connection terminated
```

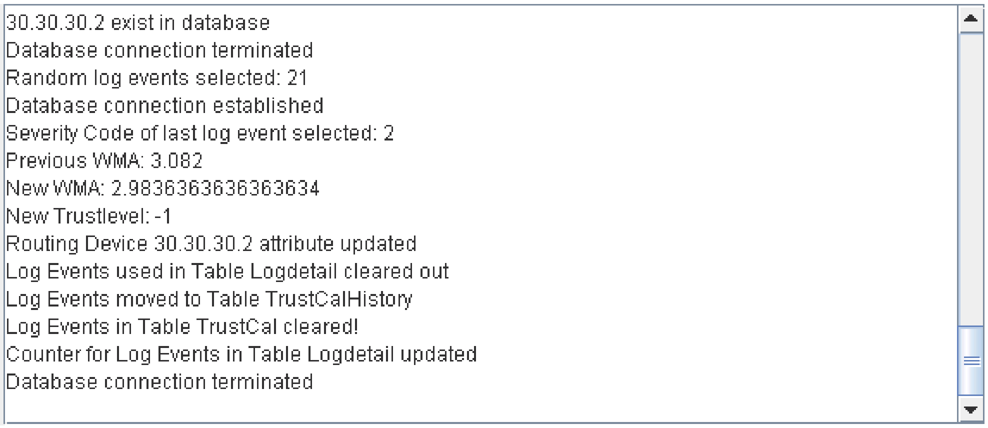
Figure 8.9: TSONE Interface – Result of Trust Level calculation for `30.30.30.2`

8.4.3 Effect of trust level on a routing device

The effect of the trust level (low or high) on the routing device is beyond the scope of this project because it depends on the network policy for different network environments. However, it is alluded to here so as to illustrate one of many actions that could be taken against a routing device with a low trust level. The solution suggested below is used for the open shortest path first (OSPF) routing protocol but it could be implemented differently for other routing protocols.

The TSONE network environment shown in Figure 8.6 represents the network environment and the devices used in the environment. Routing device `40.40.40.1` provides access to the `40.0.0.0` and `20.0.0.0` network via its Fast Ethernet interfaces. The connection to the `30.0.0.0` network is provided via its serial interface to routing device `30.30.30.2`. The routing table for these devices is as follows:

```
Router40.40.40.1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
```



```
30.30.30.2 exist in database
Database connection terminated
Random log events selected: 21
Database connection established
Severity Code of last log event selected: 2
Previous WMA: 3.082
New WMA: 2.9836363636363634
New Trustlevel: -1
Routing Device 30.30.30.2 attribute updated
Log Events used in Table Logdetail cleared out
Log Events moved to Table TrustCalHistory
Log Events in Table TrustCal cleared!
Counter for Log Events in Table Logdetail updated
Database connection terminated
```

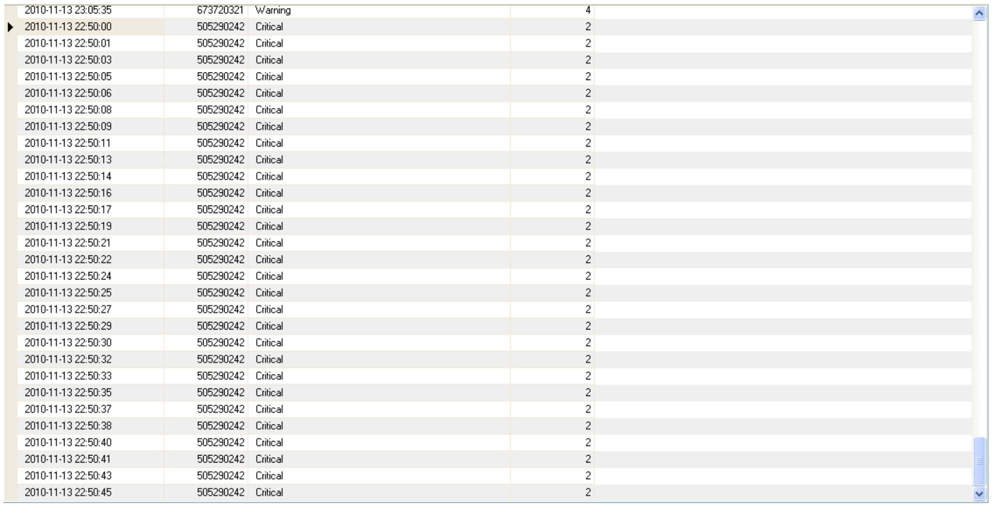
Figure 8.10: TSONE Interface – Trust Level decrease for 30.30.30.2

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
O       10.10.10.0 [110/782] via 30.30.30.2, 00:25:35, Serial0/1/0
     20.0.0.0/24 is subnetted, 1 subnets
C       20.20.20.0 is directly connected, FastEthernet0/0.3
     30.0.0.0/24 is subnetted, 1 subnets
C       30.30.30.0 is directly connected, Serial0/1/0
     40.0.0.0/24 is subnetted, 1 subnets
C       40.40.40.0 is directly connected, FastEthernet0/0.2

Router30.30.30.2#show ip route
```



2010-11-13 23:05:35	673720321	Warning	4
2010-11-13 22:50:00	505290242	Critical	2
2010-11-13 22:50:01	505290242	Critical	2
2010-11-13 22:50:03	505290242	Critical	2
2010-11-13 22:50:05	505290242	Critical	2
2010-11-13 22:50:06	505290242	Critical	2
2010-11-13 22:50:08	505290242	Critical	2
2010-11-13 22:50:09	505290242	Critical	2
2010-11-13 22:50:11	505290242	Critical	2
2010-11-13 22:50:13	505290242	Critical	2
2010-11-13 22:50:14	505290242	Critical	2
2010-11-13 22:50:16	505290242	Critical	2
2010-11-13 22:50:17	505290242	Critical	2
2010-11-13 22:50:19	505290242	Critical	2
2010-11-13 22:50:21	505290242	Critical	2
2010-11-13 22:50:22	505290242	Critical	2
2010-11-13 22:50:24	505290242	Critical	2
2010-11-13 22:50:25	505290242	Critical	2
2010-11-13 22:50:27	505290242	Critical	2
2010-11-13 22:50:29	505290242	Critical	2
2010-11-13 22:50:30	505290242	Critical	2
2010-11-13 22:50:32	505290242	Critical	2
2010-11-13 22:50:33	505290242	Critical	2
2010-11-13 22:50:35	505290242	Critical	2
2010-11-13 22:50:37	505290242	Critical	2
2010-11-13 22:50:38	505290242	Critical	2
2010-11-13 22:50:40	505290242	Critical	2
2010-11-13 22:50:41	505290242	Critical	2
2010-11-13 22:50:43	505290242	Critical	2
2010-11-13 22:50:45	505290242	Critical	2

Figure 8.11: Database for *trustcalhistory*

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
     20.0.0.0/24 is subnetted, 1 subnets
O       20.20.20.0 [110/782] via 30.30.30.1, 00:25:42, Serial0/1/0
     30.0.0.0/24 is subnetted, 1 subnets
C       30.30.30.0 is directly connected, Serial0/1/0
```

```
     40.0.0.0/24 is subnetted, 1 subnets
O       40.40.40.0 [110/782] via 30.30.30.1, 00:25:42, Serial0/1/0
```

The routing table shows the routes discovered through a physical connection and routes discovered through the OSPF routing protocol. The later routes (routes discovered via OSPF) use a routing metric in square brackets to determine the shortest path to a destination. If the metric is too high the routing protocol will use another route. The route metric for the routes discovered above is set at 782 for the OSPF interfaces. The route metric for `30.30.30.2` is increased to 1000 in another instance of the routing table below:

```
Router30.30.30.2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
     20.0.0.0/24 is subnetted, 1 subnets
O       20.20.20.0 [110/1000] via 30.30.30.1, 00:02:04, Serial0/1/0
     30.0.0.0/24 is subnetted, 1 subnets
C       30.30.30.0 is directly connected, Serial0/1/0
     40.0.0.0/24 is subnetted, 1 subnets
```

```
O       40.40.40.0 [110/1000] via 30.30.30.1, 00:02:04, Serial0/1/0
```

The above explanation and example are given as a proposed action that could be taken on a routing device with a low trust level. Thus, the solution proposed requires the network administrator to increase the route metric cost for a routing device with a low trust level. As observed in this and previous chapters, events in a Syslog file could happen as a result of different network related problems. Event messages are generated from different components within a routing device. These events could be related to hardware or software failure and malicious intrusion of any kind. The above solution will enable the network administrator to isolate the routing device from an active part of the network.

8.5 Conclusion

This chapter has demonstrated how trust can be incorporated and implemented in a service-oriented network environment as described in the TSONE model. The prototype also illustrated how event logs of routing devices can be used to determine a central tendency and ultimately a trust level. The use of trust levels for routing devices is presented as a viable solution to the problem of reliability of routing devices where forwarding and switching of critical data traffic is essential. The next chapter concludes the dissertation and contains final comments about the TSONE model, the TSONE implementation and future work.

Chapter 9

Conclusion

It's more fun to arrive at a conclusion than to justify it.

– *Malcolm Forbes*

In my beginning is my end.

– *T. S. Eliot*

9.1 Introduction

This dissertation aimed to present a model for trust in a service-oriented network environment. The model presented followed the problem statement and study of the subject area. Chapter 1 introduced the research and gave a general overview, which led to a number of research questions to be considered. The study of the various facets of this environment enabled the adaptation of trust models to derive a trust level for routing devices.

In this chapter the objectives of the current research are revisited by reviewing the problem statement and the research questions that were posed. Chapter 9 concludes with a summary of the main contribution made by the research and suggestions for future research.

9.2 The Problem Statement

The main focus of this research was to determine a routing device's trust level. Since this is related to the service provided by the routing device, the functionalities of a routing device as part of the OSI network layer were also important. The question that arose was: **how do we assess the trustworthiness of a component, such as a routing device, of the network layer?**. With regard to the trustworthiness of a routing device, we did not pay attention to the routing protocol or the transport protocol functionality on the routing device. Our assessment of the trustworthiness of a routing device focussed explicitly on the device, the routing system and its components.

Hence the following research questions were put forward and investigated:

What services are provided by routing devices?

The answer to this question was given in Chapter 2 and began with an overview of the service-oriented architecture (SOA). The characteristics and requirements of this architecture were identified with the aim of adapting this architecture to the service-oriented network environment (SONE). The characteristics subsequently defined the components of a service-oriented architecture – service requester, service provider and a service registry. The requirements of this architecture included: loose coupling, implementation neutrality, flexible configurability, persistence, granularity and teams. These requirements are met by the current TSONE implementation.

Next, a SONE was defined based on the definition of an SOA. Service provision was linked to layer 3 of the ISO/OSI model (i.e. the network layer), and the functions of this layer were identified as error control, path determination and switching. These were the services provided by routing devices since routing devices provided services on this layer.

What are the trust/security requirements for routing devices in a network environment?

Later on in Chapter 2, the trust/security requirements for routing devices were described. These included trustworthiness of routing devices (i.e. reliability and availability of these devices), the need for a security mechanism for routing protocols implemented on routing devices because routing protocols were vulnerable to attacks, and their routing tables. These requirements necessitated a further investigation into how trust could be implemented in a SONE.

How can trust be represented in a network environment?

The above question was answered partly in Chapter 3 by means of a background study of trust implementation in different contexts. The representation of trust in different environments allowed trust implementations to be classified into two environments – virtual and human. Different views of trust by different researchers were also examined. This led to the conclusion that trust is context-dependent.

This question was answered in its entirety in Chapter 4 with a brief overview of the network layer security and why trust was an effective alternative. The requirements for trust in a SONE (TSONE) were presented as a combination of attributes of trust and reputation systems and the SOA requirements presented in Chapter 2. The requirements for a TSONE were described as: persistent and longevity, having an implementation-independent and flexible configuration, ability to capture activity and current events in the network, use of activity to guide trust decisions and scalability.

Which trust model is possibly suited for practical implementation on routing devices in a network environment?

A literature study of trust models was presented in Chapter 3 to answer this question. This study allowed for an in-depth examination of trust models and reputation systems available. Trust models that were examined included: Network trust opinions (Ntropi), eBay's feedback system and the computational model of trust and reputation. Ntropi's trust attributes provided a basis for implementing a TSONE. A motivation for Ntropi was provided in Chapter 3.

9.3 Does the model meet the TSONE requirements?

The requirements of an SOA and of a trust model were described in Chapters 2 and 3. They were subsequently used to derive the TSONE requirements and are briefly discussed next so as to evaluate the extent to which they met the requirements of the TSONE.

Persistent and long-lived

Routing devices in the TSONE environment (like in any other network environment) have consistent connectivity with and accessibility to each other. The TSONE model was found to allow the network administrator to manage the routing device as long as it was operational.

Implementation-independent and flexible configuration

The TSONE model was found to allow for any routing device to join the environment regardless of the routing protocol on the routing device. An instance of the routing device was created in the TSONE database as long as it sent its event logs to the network management server (NMS). Hence, routing devices' configuration depended on merely pointing its log events to the NMS IP address.

Capturing activity and current events in the network

Activities on the network were captured via the NMS and stored in the TSONE database. As soon as the routing device reported an event, either on the network or on the routing system to the NMS, the event was initially stored in the Syslog file and then moved to the TSONE database at the network administrator's prompt.

Using activity to guide trust decisions

The log events stored in the TSONE database were used to compute the trust level of routing devices. These events indicated the priority and severity of the event and the trust level was determined based on the severity of these events.

Scalability

The network environment can span across several autonomous systems with several devices. The TSONE model was found to accommodate as many routing devices as possible. The only limitation concerned the capacity of the NMS.

9.4 Main Contribution

The main contribution of this dissertation can be summarised as follows:

- The TSONE model that was presented is a solution to the problem of determining the trustworthiness of a routing device. Trustworthiness of routing devices indicates whether the device is always available and comments on the device's reliability. Hence, the network administrator can evaluate the routing device as a service provider.
- A Syslog file of events occurring on a router can be used to infer trust level based on severity code of events generated in the Syslog file.
- The research makes an important contribution by using a statistical function to determine the central tendency on severity code of log events. And ultimately arrive at a trust level.

9.5 Future Research

The limitations implicit in the TSONE model constitute the basis for future research that can be conducted in this field.

- The trust levels computed for a routing device over a period of time are not used again. That is, the trust level computed is used once only, before another trust level is computed. Over a period of time the trust level computed (whether good or bad), could be used in favour of or against the routing device. Time could be an extra attribute when trust level is computed.
- The full implementation of the Ntropi trust model includes a phase value that depicts the relationship between entities. Future work in this area could involve introducing a relationship attribute to determine how

routing devices perceive other other. This could serve as an extension of the routing table. In other words, a relationship attribute or trust attribute could be part of a route shown in the routing table.

- Distrust was not addressed in this work. However, a very untrustworthy routing device could be placed in a state of distrust after having been rated at a "very untrustworthy" trust level.
- Human involvement in administrating this environment is essential. However, it could be interesting to investigate an automated process whereby the trust level is determined and a script is executed to the routing device to update the routing metrics could be interesting.

Bibliography

[1] Amazon auction. http://auctions.amazon.com.

[2] Facebook. http://www.facebook.com.

[3] Hotmail. http://www.hotmail.com.

[4] Hyperterminal. http://www.hilgraeve.com.

[5] MySpace. http://www.myspace.com.

[6] Mysql. http://www.mysql.com.

[7] SoftRecipe. http://www.softrecipe.com.

[8] ebay, 2008. http://www.ebay.com.

[9] Kiwi enterprises, 2008. http://www.kiwisyslog.com.

[10] Oxford English Dictionary, Online Edition 2008. Oxford University Press.

[11] YouTube, 2008. http://www.youtube.com.

[12] Alfarez Abdul-Rahman. *A Framework for Decentralised Trust Reasoning*. PhD thesis, University of London, 2005.

[13] Alfarez Abdul-Rahman and Stephen Hailes. A distributed trust model. In *NSPW '97: Proceedings of the 1997 workshop on New security paradigms*, pages 48–60, New York, NY, USA, 1997. ACM.

[14] K.A. Abuosba and A.A. El-Sheikh. Formalizing service-oriented architectures. *IT Professional*, 10(4):34–38, 2008.

[15] Emmanuel A. Adigun and J. H. P. Eloff. Defining a trusted services-oriented network environment. In *Proceedings of the Second International Conference on Availability, Reliability and Security (AReS)*, April 2007.

[16] Jonathan E. Adler. Testimony, trust, knowing. *The Journal of Philosophy*, 91(5):264 – 275, May 1994.

[17] Algirdas Avižienis, Jean-Claude Laprie, and Brian Randell. Fundamental concepts of dependability. Technical Report UCLA CSD Report no. 010028, University of California, Los Angeles, 2001. http://www.cs.ncl.ac.uk/research/pubs/trs/papers/739.pdf.

[18] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[19] Annette Baier. Trust and antitrust. *Ethics*, 96(2):231 – 260, January 1986.

[20] Madalina Baltatu and Antonio Lioy. Ip security. In *NATO Workshop on Advanced Security Technologies in Networking*, May 2000.

[21] Bernard Barber. *The Logic and Limits of Trust*. Rutgers University Press, 1983.

[22] BBCNews. Details emerge on YouTube block, February 2008. http://news.bbc.co.uk/go/pr/fr/1/hi/technology/726600.stm.

[23] S. M. Bellovin. Security problems in the tcp/ip protocol suite. *SIGCOMM Comput. Commun. Rev.*, 19(2):32–48, 1989.

[24] M. Bichler and K-J. Lin. Service-oriented computing. *Computer*, 39(3):99 – 101, March 2006.

[25] Susan D. Boon and John G. Holmes. The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. In Robert A. Hinde and Jo Groebel, editors, *Cooperation and Prosocial Behaviour*. University Press, Cambridge, 1991.

[26] M. Castro, M. Costa, and A. Rowstron. Performance and dependability of structured peer-to-peer overlays. In *International Conference on Dependable Systems and Networks*, pages 9 – 18. IEEE, 2004.

[27] R. Cellan-Jones. YouTube back in pakistan, February 2008. `http://www.bbc.co.uk/blogs/technology/2008/02/youtube_back_in_pakistan.html`.

[28] Anibran Chakrabarti and G. Manimaran. Internet infrastructure security: A taxonomy. *IEEE Network*, 16(6):13 – 21, November/December 2002.

[29] Elizabeth Chang, Tharam Dillon, and Farookh K. Hussain. *Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence*. John Wiley & Sons, England, 2006.

[30] Elizabeth J. Chang, Farookh Khadeer Hussain, and Tharam S. Dillon. Fuzzy nature of trust and dynamic trust modeling in service oriented environments. In *SWS '05: Proceedings of the 2005 workshop on Secure web services*, pages 75–83, New York, NY, USA, 2005. ACM Press.

[31] Cisco. *System Message Guide (4.x)*, 2007. www.cisco.com/en/US/docs/switches/lan/catalyst5000/catos/4.5/system/messages/edesc.ht

[32] M. Coetzee and J. H. P. Eloff. Autonomous trust for web services. *Internet Research*, 15(5):498–507, 2005.

[33] Marijke Coetzee. *WSACT - A Model for Web Services Access Control incorporating Trust*. PhD thesis, University of Pretoria, 2006.

[34] Partha Dasgupta. Trust as a commodity. In Diego Gambetta, editor, *Trust: Making and Breaking cooperative Relations*, pages 49 – 72. Basil Blackwell, Oxford, 1988.

[35] Chrysanthos Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407 – 1424, 2003.

[36] J.B. Folkerts. A comparison of reputation-based trust systems. Master's thesis, Rochester Institute of Technology, 2007.

[37] Behrouz A. Forouzan. *TCP/IP Protocol Suite*. McGraw Hill, USA, 2000.

[38] M. Fowler and K. Scott. *UML Distilled: A Brief Guide to the Standard Object Modelling Language*. Addison-Wesley, USA, 2 edition, 2000.

[39] Diego Gambetta. Can we trust trust? In Diego Gambetta, editor, *Trust: Making and Breaking cooperative Relations*, pages 213–237. Basil Blackwell, Oxford, 1988.

[40] R. T. Golembiewski and M. McConkie. The centrality of interpersonal trust in group processes. In G. L. Cooper, editor, *Theories of group processes*, pages 131–185. John Wiley & Sons, London, 1975.

[41] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.

[42] M. Hollick, I. Martinovic, and I. Rimac. A survey on dependable routing in sensor networks, ad hoc networks and cellular networks. In *Proceedings of the 30th Euromicro Conference*, pages 495 – 502, 2004.

[43] Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, January-March 2004.

[44] Kevin J. Houle, George M. Weaver, Neil Long, and Rob Thomas. Trends in denial of service attack technology. Technical report, Computer Emergency Response Team (CERT) Coordination Center, 2001.

[45] Dijiang Huang, Qing Cao, Amit Sinha, Marc J. Schniederjans, Cory Beard, Lein Harn, and Deep Medhi. New architecture for intra-domain network security issues. *Commun. ACM*, 49(11):64–72, 2006.

[46] M.N. Huhns and M.P. Lin. Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE*, 9(1):75 – 81, January-February 2005.

[47] P. Hunter. Pakistan YouTube block exposes findamental internet security weakness: Concern that Pakistani action affected access elsewhere in the world. *Computer Fraud & Security*, 2008(4):10 – 11, April 2008.

[48] International Standard Organization. *Information Technology – Open Systems Interconnection – Basic Refrence Model: The Basic Model*, 1994. http://standards.iso.org/ittf/licence.html.

[49] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, pages 1 – 27, July 2005.

[50] Paul Kedrosky. The first disaster of the internet age. (cover story). *Newsweek (Atlantic Edition)*, 152(18):p24 – 29, 2008.

[51] S. Kent and K. Seo. Security architecture for the internet protocol. *Internet RFC 4301*, December 2005. http://rfc.net/rfc4301.html.

[52] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (s-bgp). *IEEE Journal on Selected Areas in Communications*, 18(4):582 – 592, April 2000.

[53] Stephen T. Kent. Securing the border gateway protocol. *Internet Protocol*, 6(3):2 – 14, September 2003.

[54] Gary C. Kessler. *An Overview of Cryptography*. Hill Associates, 1998. http://www.garykessler.net/library/crypto.html.

[55] D. Krafzig, K. Banke, and D. Slama. *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall, 2005.

[56] R. M. Kramer and T. R. Tyler. *Trust in organizations: Frontiers of Theory and Research*. Sage, Thousand Oaks, California, 1996.

[57] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, USA, 2001.

[58] C. Lonvick. The bsd syslog protocol. *Internet RFC 3164*, August 2001. http://rfc.net/rfc3164.html.

[59] Boris Lublinsky. Defining SOA as an Architectural Style, January 2007. `http://www.ibm.com/developerworks/architecture/library/ar-soastyle/`.

[60] Niklas Luhmann. *Trust and Power*. Wiley, 1979.

[61] Niklas Luhmann. Familiarity, confidence, trust. In Diego Gambetta, editor, *Trust: Making and Breaking cooperative Relations*, pages 94 – 107. Basil Blackwell, Oxford, 1988.

[62] Ravi Malhotra. *IP ROuting*. O'Reilly Media Inc., USA, 2002.

[63] Stephen Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computer Science and Mathematics, University of Sterling, 1994.

[64] Stephen Marsh and Mark R. Dibben. Trust, untrust, distrust and mistrust an exploration of the dark(er) side. 3477/2005:17 – 33, May 2005.

[65] Roger C. Mayer, James H. Davis, and David Schoorman. An integrative model of orgaizational trust. *The Academy of Management Review*, 20(3):709 – 734, July 1995.

[66] D.H. McKnight and N.L. Chervany. The meanings of trust. Technical Report Working Paper Series 96-04, University of Minnesota, Management Information Systems Research Center, 1996. http://misrc.umn.edu/wpaper/.

[67] D. Mcpherson. Internet routing insecurity: Pakistan nukes YouTube, February 2008.

[68] Barbara Misztal. *Trust in Modern Societies*. Polity Press, 1996.

[69] J. Moy. Ospf version 2. *Internet RFC2328*, April 1998. http://rfc.net/rfc2178.html.

[70] Lik Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games and Social Networks*. PhD thesis, Massachusetts Institute of Technology, 2002.

[71] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. A computational model of trust and reputation. In Jr. Ralph H. Sprague, editor, *Proceedings of the 35th Annual Hawaii International Conference on System*

Sciences (HICSS), pages 2431 – 2439. IEEE Computer Society, January 2002.

[72] news.com. Mark cuban: Only a 'moron' would buy youtube, 2006. http://news.zdnet.com/2100-9588_22-6121034.html.

[73] OASIS. *Reference Model for Service-Oriented Architecture 1.0*, 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.

[74] Mike P. Papazoglou. Service-oriented computing: Concepts characteristics and directions. In *Proceedings of the fourth International confrence on Web Information Systems Engineering (WISE'03)*, pages 75–83, New York, NY, USA, 2003. ACM Press.

[75] Charles P. Pfleeger and Shari Lawrence Pfleeger. *Security in Computing.* Prentice Hall, USA, 4th edition, 2007.

[76] A. Pirzada, A. Datta, and C. McDonald. Incorporating trust and reputation in the dsr protocol for dependable routing. *Computer Communications*, 29(15):2806 – 2821, 2006.

[77] Bizrate Online Business Ratings. http://www.bizrate.com/ratings_guide/guide.html.

[78] P. Resnick and R. Zeckhauser. Trust among strangers in electronic transactions: Empirical analysis of ebay's reputation system. *Advances in Applied Microeconomics*, 11, 2002.

[79] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communication of the ACM*, 43(12):45–48, 2000.

[80] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on ebay: A controlled experiment, 2003.

[81] Robert J. Samuelson. The internet and gutenberg. *Newsweek (Atlantic Edition)*, 135(4):p2 – 2, 2000.

[82] J.F. Short. The social fabric of risk. *American Sociological Review*, 49:711 – 725, December 1984.

[83] David Sprott and Lawrence Wilkes. Understanding service-oriented architecture. *The Microsoft Architecture Journal*, pages 237–241, 2004.

[84] Telecommunication Standardization Sector. *ITU-T Recommendation E.600: Terms and Definitions of Traffic Engineering.* International Telecommunications Union, 1994.

[85] Telecommunication Standardization Sector. *ITU-T Recommendation E.800: Terms and Definitions related to Quality of Service and Network Performance including Dependability.* International Telecommunications Union, 1994.

[86] R. Thayer, N. Doraswamy, and R. Glenn. Ip security document map. *Internet RFC 2411*, November 1998. http://rfc.net/rfc2411.html.

[87] L. Vercouter, S. Casare, J. Sichman, and A. Brando. An experience on reputation models interoperability based on a functional ontology. In *IJCAI '07: Proceedings of the Twentieth International Join Conference on Artficial Inteligence*, pages 617 – 622, Hyderabad, India, 2007.

[88] Tao Wan, Evangelos Kranakis, and P.C. van Oorschot. Pretty secure bgp (psbgp), September 2004.

[89] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proceedings of the Third International Conference on Peer-to-Peer Computing*, pages 150–157, September 2003.

[90] Russ White. Securing bgp through secure origin bgp. *Internet Protocol*, 6(3):15 – 22, September 2003.

[91] Oliver E. Williamson. Calculativeness, trust and economic organization. *Journal of Law and Economics*, 36(1):453 – 486, April 1993.

[92] Marika Wojcik. Evaluation criteria for trust models with specific reference to prejuidice filters. Master's thesis, University of Pretoria, 2007.

[93] World Wide Web Consortium (W3C). *Web Service Architecture*, 2004. http://www.w3.org/TR/ws-arch/wsa.pdf.

[94] Yahoo!Auction. http://auction.yahoo.com.

[95] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(19):881 – 907, 2000.

Glossary

Application Layer	The application layer of the ISO/OSI model provides a suitable user interface for interpretation of messages sent across a network.
BGP	Border Gateway Protocol is an intra-domain routing protocol
DDoS	Distributed Denial of Service
IP	Internet Protocol
IPSec	Internet Protocol Security
ISO/OSI	International Standards Organization Open Systems Interconnection
Network	An interconnected system, consisting of devices to forward message across the system.
Network Environment	An environment that has network devices connected to each other
Network Layer	The network layer of the ISO/OSI model provides routing capability for messages or packets in a network environment.

NMS Network Management Server

Ntropi Network trust opinions is a trust model implemented for agents in a distributed network environment

OSPF Open Shortest Path First is a routing protocol.

ReadSyslog This the java class implemented to read the Syslog file

Reputation A collective measure of trustworthiness based on referrals.

Reputation Systems A system that keep a record of interactions between two interacting parties for future use by other parties.

Router A router is a network device that connects two subnetworks together. In this document, it is used interchangeably with routing devices

Routing Devices A router is a network device that forwards data packets to another part of a network. In this document, it is used interchangeably with routers.

Routing Protocol Routing protocols are used to determine the path travelled by packets in a network environment

Service(s) Functionality provided by a routing device.

Service-Oriented Architecture A service-oriented architecture is a framework that allows services to be published and discovered through a service registry by using a standard protocol

Service-Oriented Computing This is the computing paradigm that utilises services as fundamental elements for developing applications and domain

Service-oriented environment An open, collaborative, dynamic and distributed environment where services are requested and provided based on what is available in the service registry.

Service-oriented network environment A collaborative environment where network devices utilise their resources to publish and discover services available in a network environment.

Severity This is the priority of an event in a syslog file

Severity Code Numeric value assigned to the severity of an event

SOA Service-Oriented Architecture

SOC Service-Oriented Computing

SONE Service-Oriented Network Environment

Syslog System Log. This is protocol is used to report events

Trust A level of reliance placed in a service provider based on its accessibility and the expected outcome of the service provider.

Trust Model Trust models are computational efforts to represent trust relationships and trust values in a virtual environment. It is used interchangeably with reputation systems.

TrustModel Java class for the trust model implementation

TSONE Trust in Service-Oriented Network Environment.

TSONEApp This is the prototype implementation for TSONE

UML Universal Modelling Language

Web services This is a middleware technology that offers a standard communication interfaces to facilitate communication between dynamic applications over distributed network environment