ON-DEMAND SECURITY AND QoS OPTIMIZATION

IN

MOBILE AD HOC NETWORKS

BY

ZHENGMING SHEN

Master of Science

Oklahoma State University

Tulsa, Oklahoma

2003

Submitted to the Faculty
of the Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December 2006

ON-DEMAND SECURITY AND QoS OPTIMIZATION

IN

MOBILE AD HOC NETWORKS

Dissertation Approved:

Dr. Johnson Thomas

_____

Dissertation Advisor

Dr. G. E. Hedrick

_____

Dr. Venkatesh Sarangan

_____

Dr. Debao Chen

_____

Dr. Mark Weiser

_____

Dr. A. Gordon Emslie

_____

Dean of the Graduate College

PREFACE

Until recently, Security and QoS were considered as separate entities, especially in a mobile ad hoc network environment. Most widely used security mechanisms create heavy overhead and delay to communications. Research in wireless networks indicate more security will create more overhead, which will impact overall network QoS.

This dissertation suggests policy based plug-in security framework to provide more flexible security support, and a multi-layer QoS guided routing algorithm to provide better QoS performance, specifically for ad hoc network environments. In addition, we propose an on-demand security and QoS optimization algorithm which can balance security and QoS to optimize network performance.

By using the proportional integral derivative (PID) feedback control, the proposed optimization algorithm constantly monitors the ad hoc network resource status, if there are enough resources available to handle current QoS requirements, it will implement more security policies dynamically to make the network less vulnerable. This results in significant increase of network resource utilization, better QoS performance and more secure ad hoc networks.

How can we determine that a new routing protocol is more secure than any existing protocol? In this dissertation, we propose an attack tree and state machine based security evaluation mechanism for ad hoc networks. This is a new security measurement

metric to compare the relative security of two routing protocols on the same Ad Hoc network model.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Dr. Johnson Thomas for his guidance and assistance at Oklahoma State University. I would also like to thank my committee members, Dr. G.E. Hedrick, Dr. Venkatesh Saragan, Dr. Debao Chen, Dr. Martin Crossland and Dr. Mark Weiser, for their helpful contributions and advice.

Heart-felt thanks goes to my wife and my parents for their unending encouragement and emotional support throughout the years.

Finally I would like to thank all my friends who stood beside me with their unfailing and indispensable support.

TABLE OF CONTENTS

Page

Chapter 1 Introduction

1.1 Background.....1
1.2 Security.....3
1.3 QoS.....4
1.4 Security and QoS Optimization.....5
1.5 Security Measurement.....6

Chapter 2 Review of the Literature

2.1 Overview.....7
2.2 Security.....9
2.3 QoS.....11
2.4 Security and QoS Optimization.....15
2.5 Security Measurement.....18

Chapter 3 Objectives and Assumptions

3.1 Dissertation Objectives.....22
3.2 Design Assumptions.....23
3.2 Simulation Tool.....24

Chapter 4 Policy Based Security

4.1 Introduction.....25
4.2 Domain.....27
4.3 Policy.....28
4.4 Domain Join.....29
4.5 Resource Accessing.....31
4.6 Domain Leaving.....33
4.7 Policy Management Architecture.....34
4.8 Label Based Security Policy Algorithm.....37

4.8.1 Label Component Definitions and Valid Characters.....38

4.8.2 How Resource Label and User Label Work Together……...39
4.8.3 Access Mediation……...42
4.8.4 How Labels Are Evaluated for Access Mediation……42

4.8.4.1 Example of Read/Write Authorizations on Groups 43
4.8.4.2 Label Security Algorithm for Read Access……..……44
4.8.4.3 Label Security Algorithm for Write Access……..……...46

4.9 Policy Management Language…….…….…..…..49
4.10 Performance Analysis..…….…….…..…53

4.10.1 Simulation Model……...53
4.10.2 Simulation Assumptions……54
4.10.3 Traffic and Mobility models……..……...54
4.10.4 Metrics……55
4.10.5 Simulation Results……..56
4.10.5 Conclusion……..…..58

Chapter 5 Multi-Layer QoS Interface Guided Routing

5.1 Introduction…….....…….……..…59
5.2 Multi-Layer QoS Interface Guided Routing…….……..…61
5.3 Path Generation..……65
5.4 Path Selection.....…….…….……..…..69
5.5 QoS Interface.....…….…….……..….72
5.6 Performance Analysis…….…….……..…...74

5.6.1 Simulation Model……..74
5.6.2 Traffic and Mobility models……..…..75
5.6.3 Simulation Results……...…76
5.6.4 Conclusion……81

Chapter 6 Security and QoS Optimization

6.1 Introduction.....……...……..…….82
6.2 Feedback Control Theory…….…….……..…..85

6.2.1 Proportional Control……..……...87
6.2.2 Proportional-Integral Control……88
6.2.3 Proportional-Integral-Derivative Control……..……...89

6.3 Security and QoS Feedback Control Loop…….……...…..……91
6.4 Measure Network Resource Availability…….…….…….....…...93
6.5 Security Plug-in Architecture..…….…….....…..…..95

6.6 Optimization Algorithm..……..…….……..…97

6.6.1 Greedy Algorithm……..……97

6.7 Policy Depolyment Post Validation…….…….....…..…..…101
6.8 Performance Analysis…….……..…..104

6.7.1 Simulation Model…….104
6.7.2 Traffic and Mobility models……...…….105
6.7.3 Security Policies……...106
6.7.4 Simulation Results……...……107
6.7.5 Conclusion……...……110

Chapter 7 Network Security Measurement

7.1 Introduction.....…….....……..……111
7.2 Fundamentals of Security and Attack..…….....……..…113

7.2.1 Security and Dependability……..……..…113
7.2.2 Faults and Errors……..……...115
7.2.3 Threats……..……...116
7.2.4 Security Principals and Policies……..……....117

7.3 Attack Surface…….....……..……..……..…..120
7.4 Proposed Measurement Technique……...……....……..122

7.4.1 Vulnerability Assessment and Security Measurement..….....…..…..122
7.4.2 State Machine.....……...……..……..……...123
7.4.3 Model Threat Agents……...125
7.4.4 Model Attack Tree……...128

7.5 Security Measurement Metric.....……...……..…...131

7.5.1 Dimensions of a Threat Agent…….131
7.5.2 Attack Goal and Attack Path……...132
7.5.3 Critical Path…….……132
7.5.4 Access Rights……...133
7.5.5 Examples…….....134

7.6 Security Measurement.....……...……..……..137
7.7 Example of Security Measurement Metric.....……...……..…...144
7.8 Conclusion.....……...……..……146

Chapter 8 Conclusions and Future Works

8.1 Overall conclusion...……147
8.2 Policy based security……149

8.2.1 Conclusion……...149
8.2.2 Future Work...……...149

8.3 Multi-layer QoS interface guided routing……150

8.3.1 Conclusion……...150
8.3.2 Future Work...……...150

8.4 Security and QoS optimization……151

8.4.1 Conclusion……...151
8.4.2 Future Work...……...151

8.5 Security Measurement…….153

8.5.1 Conclusion……...153
8.5.2 Future Work...……...153

REFERENCES……...……...154

APPENDEX...……...……..……..159

Appendix A - Glossary……...…….…….……...…….....160

LIST OF FIGURES

Figure Page

4.1 Network Policy Domain.……..……..……...……...27

4.2 Domain Joining Process…….30

4.3 Resource Accessing Validation Process……32

4.4 Domain Leaving Process……...……33

4.5 Policy Management Architecture…..…..…….…..……..……...…...……..34

4.6 Network Layer Structure with Security Policy Management…..…....….…….…..36

4.7 Resource Categorizations with Levels, Compartments, and Groups ..…….…….…39

4.8 Example: Resource Labels and user Labels ……..…….……..……...…..……40

4.9 How Label Components Interrelate……...……...……..…….……..41

4.10 Relationships between Users, Resource, and Labels ……...…….……..42

4.11 Subgroup Inheritance of Read/Write Access…….…...…….…..….44

4.12 Label Evaluation Process for Read Access..…….…….……..……..…..……..45

4.13 Label Evaluation Process for Write Access....…...…….……..……47

4.14 A sample policy in XACML format…….…..……..52

4.15 Packet Delivery Ratios.....…….……..…….....56

4.16 Routing Performance……..…..…….…...57

5.1 Network Layer Structure and QoS Metrics Mapping……..…..……...…62

5.2 Throughput for v = 5 m/s..…….....77

5.3 Average packets delay for v = 5 m/s....……...……...77

5.4 Throughput for v = 10 m/s..….....……..……..97

5.5 Average packet delay for v = 10 m/s.....80

6.1 Feedback Control System.....85

6.2 Derivative Controller.....88

6.3 Integral Controller.....89

6.4 PID Controller 90

6.5 QoS and Security PID Feedback Control Loop.....92

6.6 Network Security Policy Plug-in Architecture 95

6.7 Greedy Algorithm 98

6.8 Acceptable Utilization and Target Utilization 99

6.9 Need More Policy.....100

6.10 Policy Deployment Post Validation Process Flow 102

6.11 Policy Deployment Post Validation Algorithm.....103

6.12 Throughput for v = 10 m/s.....107

6.13 Average packets delay for v = 10 m/s 108

6.14 Security policies are used for v = 10 m/s.....109

7.1 Fault Path.....116

7.2 Network security measurement metric.....123

7.3 Aspects of a threat agent.....127

7.4 Attack tree.....129

7.5 Critical path of attack tree.....133

7.6 Security metric of AODV under sniffing attack.....140

7.7 Measure security among different networks.....142

7.8 Measure security among different threats.....143

LIST OF TABLES

Table Page

4.1 Three Dimensions of Label Security Policy ……..……...…….....37

4.2 Sensitivity Label Components …..…..…….…..…….……..…….38

5.1 QoS metrics mapping table ……..……...……..69

5.2 Interfaces mapping table ……....72

6.1 Proportional, integral and derivative controller.…….…….....86

6.2 QoS metric parameter mapping ……..……...…….....93

6.3 Security policy priority …….…….…….....…….....106

7.1 Dependability Property of a System …….……..…….....114

7.2 Security Property of a System …….……...…….....114

7.3 AODV under sniffing attack…….……...…….....135

7.4 AODV under message alternation attack …….....136

7.5 10 most common attacks in ad hoc network …….....144

7.6 Attack measurement of AODV and DSDV…….....144

LIST OF SYMBOLS

ABR	-	Associatively Based Routing.
AODV	-	Ad Hoc On-Demand Distance Vector Routing.
BER	-	Bit Error Rate.
BSAR	-	Bootstrapping and Routing.
CGSR	-	Clusterhead-Gateway Switch Routing.
CPU	-	Central Process Unit.
CSER	-	Cooperative Security-Enforcement Routing.
CSMA/CD	-	Carrier Sense Multiple Access with Collision Detection.
CTS	-	Clear To Send.
DARPA	-	Defense Advanced Research Projects Agency.
DCF	-	Distributed Coordination Function.
DSDV	-	Destination-Sequenced Distance-Vector Routing.
EWMA	-	Exponentially Weighted Moving Average.
FIFO	-	First In First Out.
LAN	-	Local Area Network.
LAR	-	Location-Aided Routing.
MAC	-	Medium Access Control.
MANET	-	Mobile Ad Hoc Network.
PDA	-	Personal digital assistants.

PID	-	Proportional, Integral and Derivative.
QoS	-	Quality of Service.
OSI	-	Open System Interconnection.
OTCL	-	Object-oriented Tool Control Language.
RTS	-	Request To Send.
SAAR	-	Security Aware Ad Hoc Routing.
SAODV	-	Secure Ad Hoc On-Demand Distance Vector Routing.
SBRP	-	Secure Bootstrapping and Routing.
SEAD	-	Secure Efficient Ad hoc Distance vector routing.
SINR	-	Signal to Interference and Noise Ratio.
SRP	-	Secure Routing Protocol.
SSR	-	Signal Stability Routing.
TORA	-	Temporary Ordered Routing Algorithm.
WLAN	-	Wireless Local Area Network.
WRP	-	Wireless Routing Protocol.
XACML	-	Extensible Access Control Markup Language.

Chapter 1

Introduction

1.1 Background

Mobile Ad Hoc Networks (MANETs) consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure [1]. They have potential applications in disaster relief, conference, and battlefield environments, and have received significant attention in recent years.

In a MANET, a message sent by a node reaches all its neighboring nodes that are located at distances up to the transmission radius. Because of the limited transmission radius, the routes between nodes are normally created through several hops in such multi-hop wireless networks [1]. Host mobility can cause frequent unpredictable topology changes [2].

In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad hoc network routing protocol is correct and efficient route establishment between a pair of nodes so that messages may be delivered in a timely manner [2]. Route construction should be done with a minimum of overhead and bandwidth consumption.

Many protocols have been proposed for MANETs, with the goal of achieving efficient routing [1]. The MANET routing methods can be categorized as two primary classes: table-driven and demand-driven.

Table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols include: DSDV, CGSR, WRP [1][3]. The major disadvantages of table-driven routing protocols are each node needs to send messages to its neighborhoods consistently to keep their routing tables update. This can cause network traffic overhead.

Demand-driven (Source-Initiated) routing protocols create routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. The demand-driven routing protocols include: AODV, DSR, TORA, ABR, SSR [1][4][5]. The demand-driven routing protocols do not need maintain routing tables, but have the overhead of route discovery.

The simulation results reported in several papers [1] [2] [4] show that normally demand-driven routing protocols have higher packet delivery ratio and need less routing messages than table-driven routing protocols.

In this dissertation, we will discuss four aspects of MANETS: security, QoS, security and QoS optimization, and security measurement.

1.2 Security

Research on securing ad hoc networks has concentrated on secure routing, intrusion detection and key management. Although these techniques will deliver the message securely to the destination or authenticate nodes, all sources have the same access rights to resources at the destination. Given the increasing sophistication of computers, cell phones, PDAs etc., that form ad hoc networks, as well as the increasing complexity of the services such networks provide, there is a need for an additional level of security for resource protection. In this dissertation we propose a distributed policy based architecture for mobile ad hoc networks, the implementation of the policy is also presented. Simulations indicate that the routing overheads associated with the proposed system make this a feasible approach for enhancing the security of mobile ad hoc networks.

1.3 QoS

Quality-of-service (QoS) routing in an Ad Hoc network is difficult because network topology may change constantly, and the available state information for routing is inherently imprecise. Existing QoS routing approaches concentrate on QoS management at the network layer. In this dissertation, we propose a holistic multi-layer QoS surface guided routing, which separates metrics at the different layers, MAC layer metrics, network layer metrics, and application layer metrics. In our model, each layer manages its own QoS and communicates with other layers through its QoS surface. Due to link failure caused by a lack of network resources and nodes' mobility on a path, the quality should not only reflect the available resources on a path but also the stability of that path. Therefore, MAC layer metrics, network layer metrics and application layer metrics are used as additional constraints to determine the quality of paths between a source and destination. Network layer metrics determine the quality of links in order to generate the paths with good quality. On the other hand, application layer metrics select exactly one path out of the paths with a good quality which is more likely to meet application requirements. Our model considers not only the QoS requirement, but also the cost optimality of the routing path to improve the overall network performance. Simulation results show that the proposed approach provides better QoS than other QoS routing protocols such as QoS-AODV under high mobility conditions.

1.4 Security and QoS Optimization

Network quality-of-service and network security have been considered as separate entities and research in these areas have largely proceeded independently. However, security impacts overall QoS and it is therefore essential to consider both security and QoS together when designing protocols for ad hoc environments as one impact the other. In this dissertation we propose a mechanism for a distributed dynamic management system which will aim to maximize QoS and/or security while maintaining a minimum user acceptable level of QoS and/or security even as network resource availability change. In order to achieve this objective, we propose three basic frameworks: a policy based plug-in security framework, multi-layer QoS guided routing and a proportional integral derivative (PID) controller. Figure 1-1 demonstrates the overall optimization system flow. Simulation results indicate the proposed PID optimized security and QoS algorithm produce similar performance as non-secure QoS routing protocols under various traffic loads.

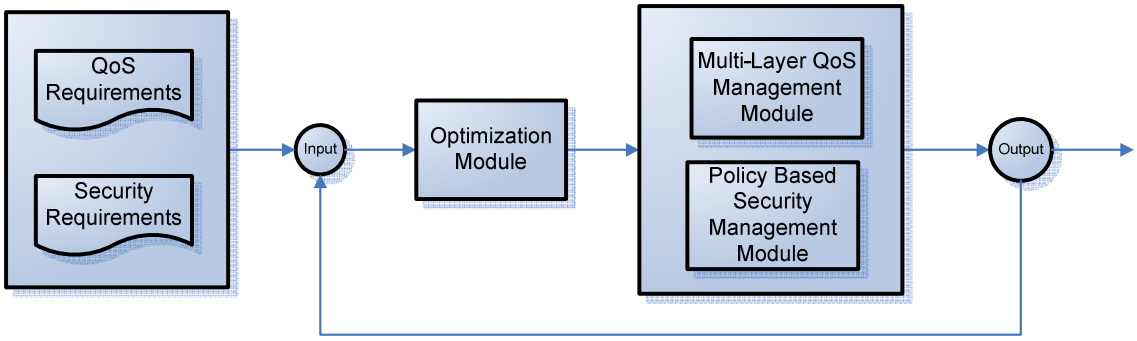


Figure 1-1 QoS and Security Optimization System

1.5 Security Measurement

Although, numerous secure and insecure ad hoc routing protocols have been proposed, it is a very difficult to evaluate the effectiveness of these protocols from a security perspective due to the absence of absolute security metrics for ad hoc networks. Not much research as been done in this area, because it is very difficult, if not impossible to define absolute security metrics for ad hoc networks.

We propose a metric to determine whether one routing protocol of an Ad Hoc network is relatively more secure than another. Rather than count bugs at the protocol code level or count vulnerability reports at the network level, we count the network's attack opportunities. We use this count as an indication of the network's security risk, likelihood that it will be successfully attacked. We describe a network's measurement metric along four abstract dimensions: attack goal, attack path, attack tree, and access rights. Intuitively, the more exposed the security risk, the more likely the network could be successfully attacked, and hence the more insecure it is. Thus, one way to improve network security is to reduce its security risk. We demonstrate and validate our method by measuring the relative security risk of different routing protocols.

Chapter 2

Review of the Literature

2.1 Overview

Mobile Ad Hoc Networks (MANETs) consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure [1]. They have potential applications in disaster relief, conference, and battlefield environments, and have received significant attention in recent years.

In a MANET, a message sent by a node reaches all its neighboring nodes that are located at distances up to the transmission radius. Because of the limited transmission radius, the routes between nodes are normally created through several hops in such multi-hop wireless networks [1]. Host mobility can cause frequent unpredictable topology changes [2].

In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad hoc network routing protocol is correct and efficient route establishment between a pair of node so that messages may be delivered in a timely manner [2]. Route construction should be done with a minimum of overhead and bandwidth consumption.

Many protocols have been proposed for MANETs, with the goal of achieving efficient routing [1]. The MANET routing methods can be categorized as two primary classes: table-driven and demand-driven.

Table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols include: DSDV, CGSR, WRP [1][3]. The major disadvantages of table-driven routing protocols are each node needs to send messages to its neighborhoods consistently to keep their routing tables update. This can cause network traffic overhead.

Demand-driven (Source-Initiated) routing protocols create routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. The demand-driven routing protocols include: AODV, DSR, TORA, ABR, SSR [1][4][5]. The demand-driven routing protocols do not need maintain routing tables, but have the overhead of route discovery.

The simulation results reported in several papers [1] [2] [4] show that normally demand-driven routing protocols have higher packet delivery ratio and need less routing messages than table-driven routing protocols.

However, all the previous routing solutions only deal with the best-effort data traffic. Connections with QoS requirements, such as video broadcasting with delay and bandwidth constraints, are not supported.

2.2 Security

Mobile ad-hoc networks operate in the absence of fixed infrastructure, which makes them easy to deploy at any place and at any time. The absence of any fixed infrastructure in mobile ad-hoc networks makes it difficult to utilize the existing techniques for network services, and this poses a number of various challenges including routing, bandwidth constraints, security and power. The diversity of nodes range from powerful lap top computers to resource constrained devices such as PDAs and cell phones. Such diversity makes it more difficult to manage and secure these networks.

Various routing solutions have been proposed for mobile ad-hoc networks, and most of these solutions can be categorized as table-driven and demand-driven. These solutions mainly focus on routing and do not concentrate much on other related issues, such as security.

Depending on the application, users within the network may want their communication to be secure. Research on securing ad hoc networks has concentrated on secure routing, intrusion detection and key management. With the increasing proliferation of powerful nodes which can now form part of an ad hoc network, existing mechanisms are not sufficient. In the current state of the art ad hoc network systems security, all nodes in the network have equal security rights. In other words, although secure routing will deliver the message securely to the destination, all sources have the same access rights to resources at the destination. With existing approaches, although a message may be delivered securely, the message itself may be trying to access or modify resources for malicious purposes. The absence of any strict security policy, could lead active attackers to easily exploit or possibly disable the mobile ad-hoc network. The consequences of this

are serious as the more powerful nodes can be attacked by smaller resource constrained nodes and the disabling of one or more powerful nodes could have a serious impact on the network. Although secure routing with intrusion detection can guarantee a certain level of security, higher level security is needed to secure the network. Furthermore, secure routing and real-time intrusion detection carry extensive overheads.

Several secure routing protocols have been proposed recently: These include SAODV [6], Ariadne [7], SEAD [8], CSER [9], SRP [10], SAAR [11], BSAR [12], and SBRP [13]. The main idea behind these protocols is to encrypt the messages using different schemes so that the message delivered correctly. Depending upon the scheme used, these secure routing protocols bind one or two security methods into the specific routing protocol.

The policy-based security management system [14] uses responsive strategy to react when network under attack. Each node has an attack monitoring agent, and when a victim node is under attack, it activates correspondent policies. It also sends a warning message to neighboring nodes. When it recovers from an attack, it sends a warning release message. To the best of our knowledge, a policy management framework has not been proposed in the literature.

2.3 QoS

The provision of QoS relies on resource reservation. Hence, the data packets of QoS connection are likely to flow along the same network path on which the required resources are reserved. The goal of QoS routing is twofold: 1) selecting a network path that has sufficient resources to meet the QoS requirements of all admitted connections and 2) achieving global efficiency in resource utilization.

QoS routing has been receiving increasingly intensive attention in the wired network domain [15]. The recent work can be divided into three broad categories: source routing, distributed routing, and hierarchical routing. In source routing [16] – [18], each node maintains an image of the global network state, which is based on a routing path that is centrally computed at the source node. The global network state is typically updated periodically by a link-state algorithm [19]. In distributed routing [20] – [23], the path is computed by a distributed computation during which control messages are exchanged among the nodes, and the state information kept at each node is collectively used in order to find a path. In hierarchical routing [24], nodes are clustered into groups, creating a multilevel hierarchy. In every level of the hierarchy, source or distributed routing algorithms are used.

The QoS routing algorithms for wired networks cannot be applied directly to Ad Hoc networks. First, the performance of most wired routing algorithms relies on the availability of precise state information. However, the dynamic nature of an Ad Hoc network makes the available state information inherently imprecise. Second, nodes may join, leave, and rejoin an Ad Hoc network at any time and any location; existing links may disappear, and new links may be formed as the nodes move. Hence, the established

paths can be broken at any time, which raises new problems of maintaining and dynamically reestablishing the routing paths in the course of data transmission.

Though some recent algorithms [18][25] were proposed to work with imprecise information (e.g., the probability distribution of link delay), they require precise information about the network topology, which is not available in an Ad Hoc network.

QoS based routing in networks with inaccurate information [18] investigated the problem of routing connections with QoS requirements across one or more networks, when the information available for making routing decisions is inaccurate and expressed in some probabilistic manner. It reviewed the uncertainty about the actual state of a node or network that arises naturally in a number of different environments, and proposed an algorithm to determine the impact of such inaccuracies on the path selection process, whose goal is then to identify the path that is most likely to satisfy the QoS requirements.

QoS routing in networks with uncertain parameters [25] discussed the multicast routing problem with multiple QoS constraints in networks with uncertain parameters. It proposed an algorithm QMRGA, a multicast routing policy for Internet, mobile network or other high-performance networks, which is based on the genetic algorithm, and can provide QoS-sensitive paths in a scalable and flexible way, in a network environment with uncertain parameters. The QMRGA can also optimize network resources such as bandwidth and delay, and can converge to the optimal or near-optimal solution within little iteration, even for a network environment with uncertain parameters. The incremental rate of computational cost is close to polynomial and is less than exponential rate. The results show that QMRGA provides a reasonable approach to QoS Multicast routing in networks environment with uncertain parameters.

Recently, cross-layer design approach [26] - [29] has been introduced into ad hoc wireless network to resolve the above issues.

A Simulator Based on a Cross-Layer Protocol between MAC and PHY Layers in WiBro Compatible IEEE 802.16e OFDMA System [28] proposed a cross-layer design frameworks for 802.16e OFDMA systems that are compatible with WiBro based on various kinds of cross-layer protocols for performance improvement: a cross-layer adaptation framework and a design example of primitives for cross-layer operation between its MAC and PHY layers. It provided a simulation framework for cross-layer analysis between the MAC and PHY layers in 802.16e systems, which shows that average cell throughput can be improved by 25-60 percent by applying careful cross-layer adaptation schemes.

Topology-Aided Cross-Layer Fast Handoff Designs for IEEE 802.11/Mobile IP Environments [29] reviewed state-of-the-art fast handoff techniques for IEEE 802.11 or Mobile IP networks. Based on that review, topology-aided cross-layer fast handoff designs are proposed for Mobile IP over IEEE 802.1.1 networks. Time-sensitive applications, such as voice over IP (VoIP), cannot tolerate the long layer-2 plus layer-3 handoff delays that arise in IEEE 802.11/Mobile IP environments. Cross-layer designs are increasingly adopted to shorten the handoff latency time. Handoff-related layer-2 triggers may reduce the delay between layer-2 handoff completion and the associated layer-3 handoff activation. Cross-layer topology information, such as the association between 802.11 access points and Mobile IP mobility agents, together with layer-2 triggers, can be utilized by a mobile node to start layer-3 handoff-related activities, such as agent discovery, address configuration, and registration, in parallel with or prior to

those of layer-2 handoff. Experimental results indicate that the whole handoff delay can meet the delay requirement of VoIP applications when layer-3 handoff activities occur prior to layer-2 handoffs.

The cross-layer protocols are designed by violating the seven-layer open systems interconnect (OSI) model to provide overall better efficiency and performance in ad hoc wireless environment. Here the functionality of multiple layers is condensed into fewer layers with the view to improving performance. The cross-layer designs involve a complex process and are still at a very early research stage with lots of studies yet to be done.

2.4 Security and QoS Optimization

Network quality-of-service and network security have been considered as separate entities and research in these areas have largely proceeded independently with few exceptions. However, security impacts overall network QoS as more security usually means more message overheads for authentication and other security functions as well as additional delays imposed due to overheads caused by encryption etc. This is especially true in an ad hoc network environment where security mechanisms such as authentication services are proposed to protect the communication on open mediums in wireless networks, thus introducing overheads that affect the QoS of communications significantly. It is therefore essential to consider both security and QoS together when designing protocols for ad hoc environments as one impacts the other.

Very little work has been done in the interaction between security and QoS in networks. What little has been done is limited to wireless networks. [30] - [33] study the impact of challenge/response authentication in wireless LANs.

An Analytical Study on the Impact of Authentication Local Area Networks [30] introduced a system model for the analysis of challenge/response authentication in wireless networks, and evaluated authentication cost, delay, and call dropping probability for different security levels. By considering traffic and mobility patterns, the numerical results indicate the impact of authentication on security and system performance.

A Quantitative Study of Authentication Networks [31] and Performance Analysis of Challenge/Authentication in Wireless Networks [32] analyzed the impact of authentication on security and QoS quantitatively, and proposed a concept of security level to describe the protection of communications according to the nature of security,

i.e., information secrecy, data integrity, and resource availability. By taking traffic and mobility patterns into account, the proposed approach establishes a direct and quantitative connection between security and QoS through the authentication. Numerical results are provided to demonstrate the impact of security levels, mobility and traffic patterns on overall system performance in terms of authentication delay and call dropping probability.

Integration of Authentication Management in Third Generation and WLAN Data Networks [33] introduced new authentication architecture for fast authentication during inter-networking handoff and large-scale heterogeneous networks to solve authentication of roaming users crossing different networks problem in WLAN. The simulation results show that the new architecture can reduce authentication latency significantly and be adaptive to user mobility and traffic.

In summary, the emphasis of [30] is on a framework to model the effect of authentication on security and QoS in one-hop wireless networks. In [31] and [32] the authors investigate the impact of security levels, mobility and traffic patterns on overall system performance in terms of authentication cost, delay, and call dropping probability. [33] introduces an authentication scheme for inter-domain roaming for 3G/WLAN systems. The emphasis here is on authentication architecture and a new authentication scheme.

Although the above research provided an analysis of the performance degradation caused by authentication and proposed an authentication scheme for inter-domain roaming for 3G/WLAN systems, none of them propose an optimized solution between security and QoS. In other words, given the network resources and traffic, can an

optimum QoS and security be achieved? This calls for a dynamic management system which will aim to maximize QoS and security while maintaining a minimum user acceptable level of QoS and security even as network resource availability change. In all the previous work the security feature (authentication specifically) is fixed and is permanent and is integrated with a QoS routing protocol. However, no solution has been provided when changing available network resources due to traffic, mobility etc. results in security features producing too much overhead such that it significantly impacts routing QoS performance. Furthermore, security is not limited to authentication. Other security features such as access rights for example have not been considered at all.

2.5 Security Measurement

Current Ad Hoc protocols assume that the mobile host will behave properly and will not introduce malicious information into the system. However, considering the application environments of Ad Hoc networks (battlefields, disaster rescue, etc.); the routing topology is prone to attack coming from both external and internal. Research has been carried out to apply security methods in wired networks to mobile Ad Hoc environments. The mechanisms that have been examined include information encryption and user authentication. But these methods face the following difficulties:

- The restriction on power consumption and the limited computational capability of mobile devices prevent the usage of complex encryption algorithms.
- The constantly changing network topology increases the difficulty and overhead of authentication. The dynamic membership put challenges on the key distribution and management.
- Most importantly, these methods can only guard against external attacks. But the attacks coming from compromised hosts have more severe impacts on performance and network connectivity.

The security and safety properties of Ad Hoc routing protocols are different from those in wired networks. Therefore, research is required on the vulnerabilities of the protocols, the attacks introduced by them, and their practical impacts on the network performance.

An attack tree and attack graph is a succinct representation of all paths through a system that end in state where an attacker has successfully achieved his goal [46]. It has been used for attack detection, defense and forensics in security analysis. The attach tree cannot only clearly define all the sub-goals along each attack path, but also the relationship between each attach paths, in order for an attacker to successfully achieve his ultimate attack goal.

It is very difficult, if not impossible to define security metrics for ad hoc networks. The concept of Attack Surface model introduced in [47] proposes a metric to determine whether one version of a system is more secure than another. Rather than measure the absolute security of a system, the proposed technique measures the relative security: Given two versions, A and B, of a system, it measures whether version A is more secure than version B with respect to their attack surface. The proposed technique does not use the attack surface metric to determine whether a version of a system is absolutely good or bad, rather to determine whether one version of a system is relatively better or worse than another. Intuitively, a system's attack surface is the ways in which the system can be successfully attacked. The attack surface of a system can be defined in terms of the system's resources. An attacker uses a system's resources to attack a system; hence a system's resources contribute to the system's attack surface. Intuitively, the more resources available to the attacker, the more exposed the attack surface. The more exposed the attack surface, the more ways the system can be attacked, and hence the more insecure it is. Given two versions, A and B, of a system, the proposed technique compares their attack surface to determine whether one is more secure than another. The attack surface measurements might be incomparable because of the way we define attack

surface along multiple dimensions, it can, however, use the attack surface measurements along with the knowledge of the usage scenario of the system to determine whether version A is more secure than version B.

Measuring relative attack surface [48] proposed a technique to measure computer Operating System vulnerability and attack ability by using attack surface metric. Every system action can potentially be part of an attack, and hence contributes to attack surface. Similarly, every system resource also contributes to attack surface. Intuitively, the more actions available to a user or the more resources accessible through these actions, the more exposed the attack surface. Rather than consider all possible system resources, the proposed measurement technique narrows its focus on a relevant subset of resource types. Attacks carried out over the years show that certain system resources are more likely to be used in an attack than others. Hence all system resources should not be treated equally. The attack surface categorizes the system resources into attack classes based on a given set of properties associated with the resources. These properties reflect the attackability of a type of resource, i.e., some types of resources are more likely to be attacked than other types. The notion of attack class is used to distinguish between resources with different attackability. These attack classes together constitute the attack surface of a system. The proposed security measurement technique measured the attack surface of four different versions of the Linux operating system and the attack surface of seven different versions of the Windows operating system. The results of both the Linux and Windows measurements confirm perceived beliefs about the relative security of the different versions. It uses the entry point and exit point framework to identify the relevant subset of resources that contribute to the attackability of a system, then determines the

attackability of each resource using a cost-benefit ratio to the attacker. By grouping the resources into attack classes based on their attackability, the attackability of these attack classes constitutes the attack surface of a system.

In, summary, the attack surface model uses state machines to represent all potential system resources that can be used by an attacker to achieve an attack goal, and compare security with respect to a given number of yardsticks, called *dimensions*. In this approach, rather than saying “System A is secure” or “System A has a measured security number N” the attack surface model says “System A is more secure than System B with respect to a fixed set of dimensions.”

The attack surface model uses all system resources as one single level, equal weight metric. However, the hierarchies of the attack tree and the dependence between each attack paths have not been considered in this model. For example, system A exposures both user name and password should be more vulnerable than system B exposures both employee salary and password, although all of the above information are been classified as sensitive data. Because an attacker can create much more damages to system A than system B by using a stolen identity to successfully login into system A. The attack surface model measures same vulnerability level for both system A and system B in this scenario.

Chapter 3

Objectives and Assumptions

3.1 Dissertation Objectives

Our overall goal is to provide a security and QoS optimization architecture and algorithm that will have better resource utilization, ultimately provide more security and better QoS solution for ad hoc networks. In the dissertation, we:

1. Propose a policy based plug-in security framework to adapt network security level on demand;
2. Propose a multi-layer QoS guided routing algorithm to achieve more efficient and reliable QoS;
3. Propose an on-demand security and QoS optimization architecture to provide better network resource utilization and optimize network performance, so to provide more secure and efficient QoS networks.
4. Propose a new security measurement metric to compare the relative security of two Ad Hoc routing protocols that is a state machine based security evaluation mechanism.

3.2 Design Assumptions

We make the following assumptions about the security and QoS optimization system:

- Each node has same signal coverage area radius R.
- Each node has adequate cache memory to hold the state information, including routing data, security data, QoS data, and optimization data.
- Each node has sufficient CPU power to handle required computations, including security authentications, QoS calculations and optimization calculations.
- Nodes are randomly moving in a pre-defined two-dimension area.

3.3 Simulation Tool

We use a detailed simulation model based on ns-2 in our evaluation. The Monarch research group at CMU developed support for simulation of multi-hop wireless networks complete with physical, data link and Medium Access Control (MAC) layer models on ns-2 [36]. The Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer protocol. The 802.11 DCF uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for unicast data transmission to a neighboring node. The RTS/CTS exchange precedes the data packet transmission and implements a form of virtual carrier sensing and channel reservation to reduce the impact of the well-know hidden terminal problem. Data packet transmission is followed by an ACK. Broadcast data packets and the RTS control packets are sent using physical carrier sensing. An unslotted CSMA technique with collision avoidance (CSMA/CA) is used to transmit these packets. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN. WaveLAN is modeled as shared-mediaradio with a nominal bit rate of 2Mb/sec and normal radio range of 250 meters.

Chapter 4

Policy Based Security

4.1 Introduction

Mobile ad-hoc networks are highly dynamic; topology changes and link breakages happen quite frequently. Therefore, we need a security solution which is dynamic. Any malicious or misbehaving nodes can generate hostile attacks. These types of attacks can seriously damage basic aspects of security, such as integrity, confidentiality and privacy of the node.

In this chapter we propose a policy based architecture for mobile ad hoc networks. Centralized policy based security has been implemented in fixed infrastructure networks, but little (if any) research has focused on ad hoc networks. The policy architecture described here is distributed and dynamic as new policies can be added and removed as nodes enter and leave the network. This policy based security may not be applicable to all nodes in the network and may be implemented only on nodes as needed. Interactions between devices need to be controlled in order to prevent unauthorized access to system resources and services. The framework also needs to be able to bind loosely with any existing or future routing protocols. To the best of our knowledge no one has proposed a policy based secure architecture for mobile ad hoc networks.

Security policies are written definitions of expectations and principles for the protection of critical information from various threats and vulnerabilities. Security policies define how the confidentiality, integrity, and availability of information are maintained. Policies typically mandate a risk assessment and data classification process for information and systems resources. Security policies also spell out responsibilities for maintaining security. They empower security personnel to control access, to monitor and maintain security, and to investigate and handle incidents. A policy based approach is flexible, scalable and permits adaptation to changes in security requirements and context of the ad hoc network by dynamically loading and removing policies from the system without interrupting its functioning. In this chapter, we propose a policy based security framework and a set of security rules to an ad hoc network, manage its membership, and control access to the services provided by the participants. We also show the proposed solution is robust to changes in the network topology.

In sections 4.2 to 4.7 we describe the proposed policy based security system. The implementation of the policy is presented in sections 4.8 and 4.9. The routing overheads and performance analysis associated with the proposed system are presented in section 4.10.

4.2 Domain

The term domain refers to a cluster of nodes in the Ad Hoc network with common attributes and properties, managed by a set of security policies, and those nodes communicate with each other.

Definition: an ad hoc domain interconnects a group of devices, maintains membership and ensures that only entities, i.e., users or computing services which possess certain credentials, attribute information and characteristics can join the domain. The members of the domain rely upon each other to provide services and share resources. These interactions are regulated through a set of well-defined rules and policies that govern the access to the services and resources in the domain.

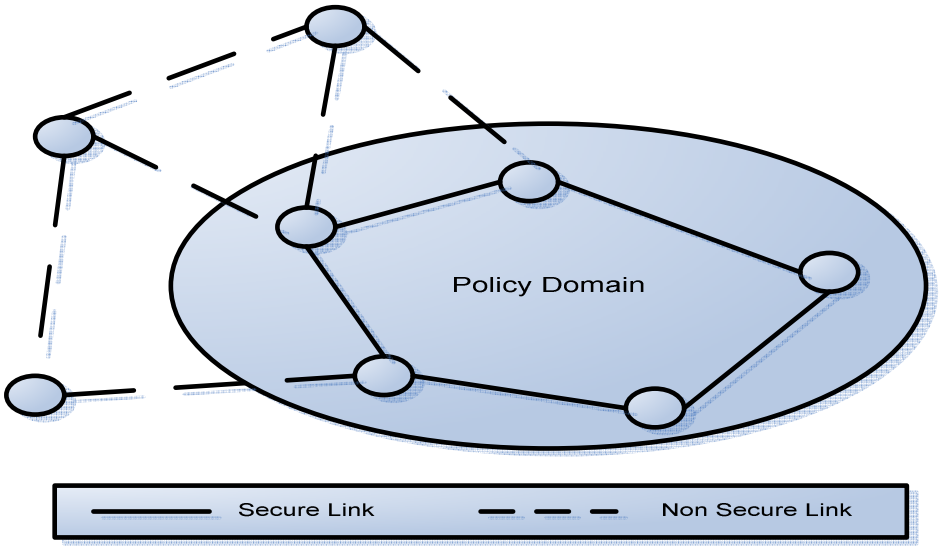


Figure 4-1 Network policy domain

4.3 Policy

Since the purpose of an ad hoc domain is to enable interactions between its members, it is thus important to ensure that these interactions are governed by well-defined policies that define the rules for accessing services and resources in the domain. Policies are explicitly specified and known to all the members.

The rationale of explicitly specifying the rules or security policies is to build trust between the members. Trust in this context derives from the fact that members' behavior is expected to be consistent with both the characteristics dictated by the admission criteria and the policies governing the behavior within the domain. Typically, the members that form the domain have to rely on each other to provide the services that they do not have on their own and usually, they do not have any a priori knowledge about each other. As a result, collaborations among them cannot be set up because they do not trust each other to use their respective services and resources. Therefore, there is a need for explicit specification of policies for each domain. By knowing the policies, a node is aware of the potential nodes that it might trust to interact with, the services and resources that it has access to, and the policies it must enforce in order to protect its resources and services.

4.4 Domain Joining

A new node N(new) periodically discovers a new domain in the neighborhood. It automatically requests to join the discovered domain. This is achieved by sending a {JOIN REQUEST} to one of the members of the domain; the receiver node contains the credentials of the requester node.

Upon receipt of the join request, the receiver node checks N(new)'s credentials satisfy the domain policies and checks that the admittance of N(new) does not violate the cardinality constraints. Credential verification can be realized using standard existing approaches. A node id is then assigned to the admitted node and the receiver node sends {JOIN REPLY} to N(new). Subsequently, the membership list is updated and broadcast to all domain members.

Sequence of events of domain joining:

1. A node first has to get the credential from the centralized certificate authority;
2. The centralized system admin sets up the public key that it obtains from the certificate authority on all the nodes in the domain, so that the newly arriving node's credential can be authenticated;
3. The newly arriving node sends a join request along with the credential issued by the certificate authority to any existing member in the domain;
4. The domain member verifies the newly arriving node's credential;
5. The domain member replies to the join request an ACCEPTED or DENIED message to the newly arriving node;
6. A new id will be assigned to the newly accepted node.
7. The new membership list will be broadcasted to all domain members.

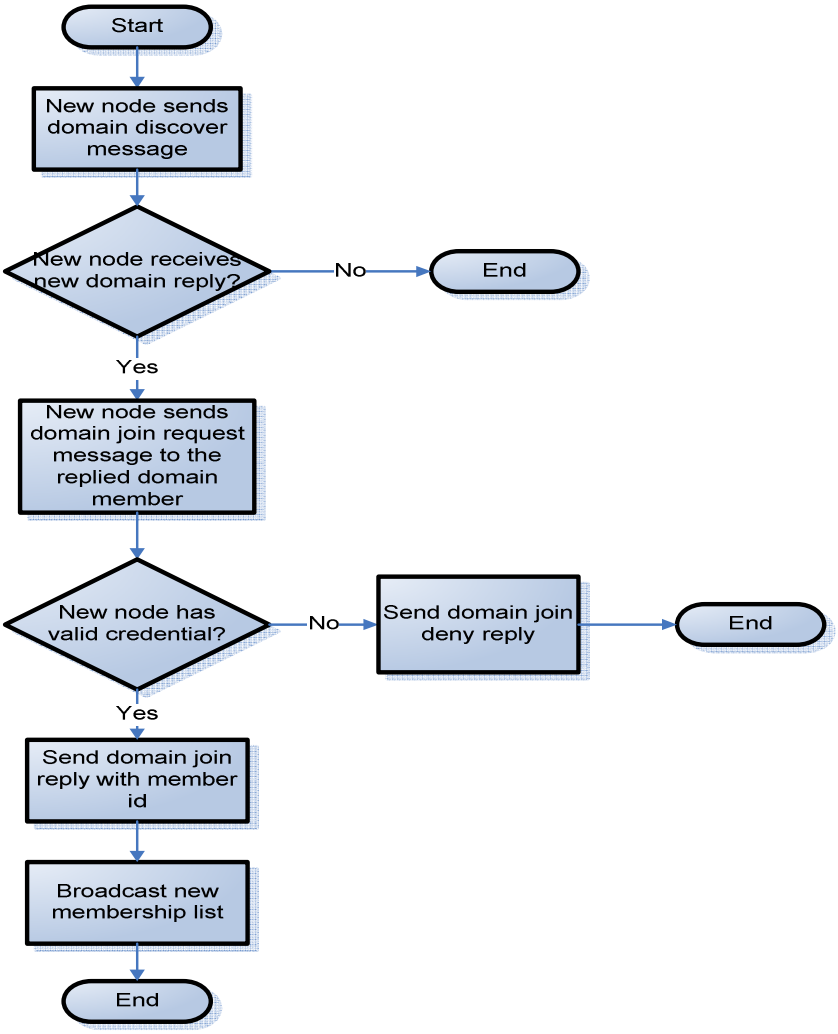


Figure 4-2 Domain joining process

4.5 Resource Accessing

Access to resources provided by domain members is regulated by domain policies. When a service provider receives a request, it first checks the membership list in order to determine the validity of the requestor's node id assignment. Then, it grants the requestor the permissions to use the resource if the authorization policies allow it.

If a violation is detected, other domain members will be notified and if needed the domain can be reconstructed.

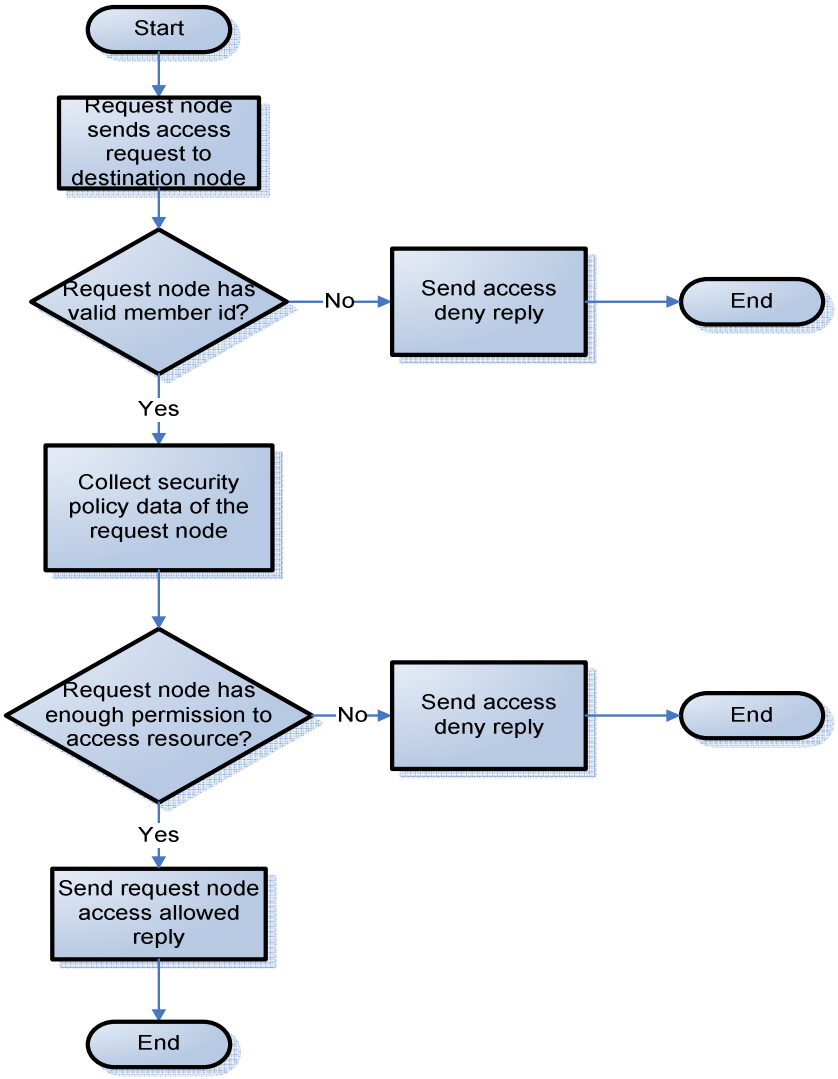


Figure 4-3 Resource accessing validation process

4.6 Domain Leaving

Two scenarios can occur: either the node notifies the neighbors that it is going to leave the community or its unexpected absence is detected by others. If it is temporarily absent (e.g., node moves out of range) but its absence is not detected by other members, no changes are necessary.

The first scenario is straightforward as the neighbors can remove the node from the membership list, which can then be broadcast to all members. In the second scenario, we rely on the other members detecting its absence, typically through a communication failure. When a communication failure occurs, a node will retry for up to x times. If the failure is confirmed, the node will remove the failure node from the membership list and broadcast the revised membership list.

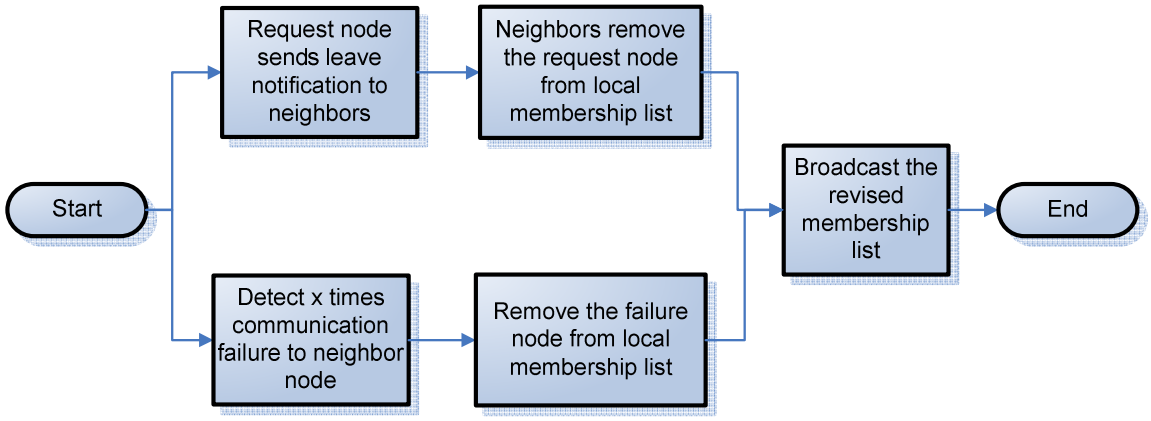


Figure 4-4 Domain leaving process

4.7 Policy Management Architecture

Figure 4-5 shows the overall architecture of the proposed framework. It is composed of four components: profile manager, membership manager, security rule manager, and policy manager. The framework runs on every node in the network.

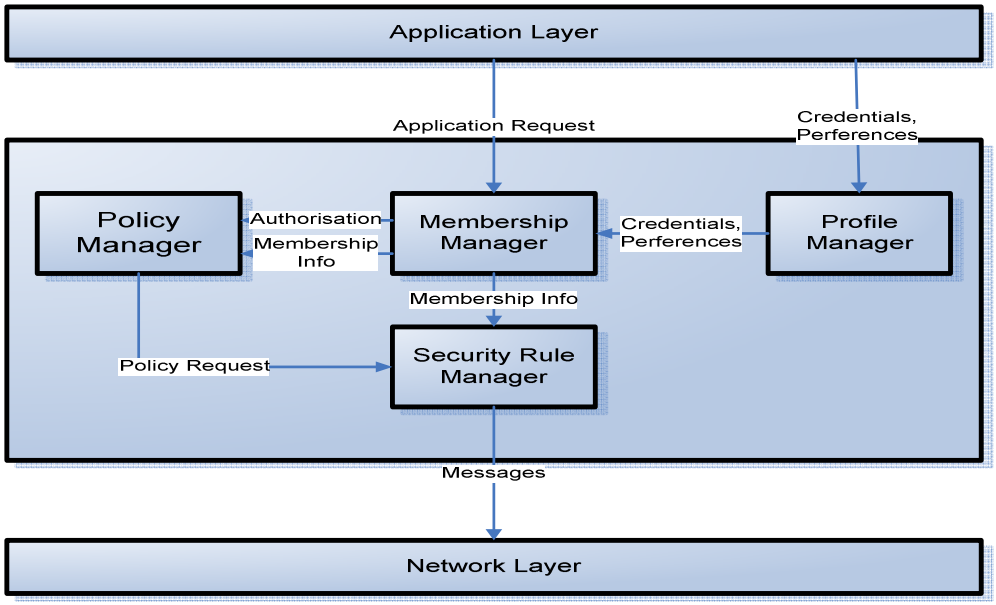


Figure 4-5 Policy management architecture

Domain management interface is the interface channel to allow the communications between application layer and the domain security module. It's the combination functions of the profile manager, membership manager and policy manager in Figure 4-5.

The profile manager component maintains the node's credentials, such as public-key certificates, private-key stores and attribute certificates. Nodes can manage their credentials and device settings through the domain management interface. In addition,

this component also maintains the node's preference on which domains the node should automatically join.

The membership manager component exposes the domain management interface to the application level, so that applications can initiate the establishment of a new domain, search for domains, as well as joining particular domains. Through this interface, the node can register the services that it is providing to other participants. The membership manager component is also responsible for verifying the newly arriving nodes' authenticity of the credentials and enforcing them by extracting and distributing the policy instances to the enforcement component, namely the security rule manager.

The security rule manager component executes various security rules for establishment (enforce security policy execution on all existing nodes within the domain), evolution (enforce security policy execution on all newly joined nodes) and management (adapt the security policy execution based on different scenario) of domains. The security rule manager component enforces both the authorization and obligation policies. Authorization policies specify what activities a node is permitted or forbidden to do to a set of target resources, obligation policies specify what activities a node must or must not do to a set of target resources. Access requests are intercepted and then verified against the policies to determine if they are permitted, obligation policies are enforced by subscribing to the specified event and executing the actions specified in the policies when the events occur. They are both enforced by 'label based policy algorithm' that is discussed in section 4.8.

The policy manager component contains all configured security policies. It provides an interface for security admin to manage the security polices. The policy

manager receives the policy request from security rule manager, responds back with the corresponding security policy based on the node's membership information, which it gets from the membership manager.

Figure 4-6 shows where the security policy management module fits in the network layer structure. It is transparent to the application layer, and independent of the specific network routing protocol.

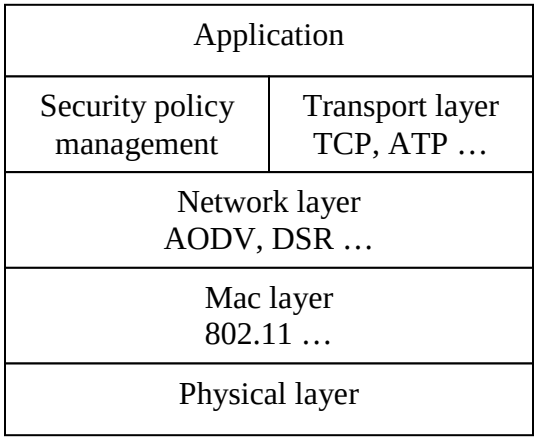


Figure 4-6 Network layer structure with security policy management

4.8 Label Based Security Policy Algorithm

Label-based security polity provides a flexible way of controlling access to sensitive resources. Label security controls resource access based on the identity and label of the user, and the sensitivity or label of the resource. The label based security approach doesn't require the entire security data to be stored in one place. Each resource carries its own security label information, and each resource might join or leave the ad hoc network at anytime. The authorization process will use access mediation with different parties to determine the actual access rights. Therefore, it is more suitable for the mobile ad hoc network scenario. With a label security policy, access to resource is controlled in three dimensions:

Resource Labels	Resources are labeled to indicate the level and nature of their sensitivity. A label on a resource specifies the sensitivity of the information and explicitly defines the criteria that must be met for a user to access it.
User Labels	Users are assigned a range of levels, compartments, and groups which indicate their label authorizations. A label assigned to a user determines the user's access to labeled resource.
Policy Privileges	Certain users may be given rights to perform special operations, and to access resource beyond their label authorizations.

Table 4-1 Three Dimensions of Label Security Policy

- A label on a resource specifies the sensitivity of the information about the resource and explicitly defines the criteria that must be met for a user to access the resource.
- Label authorizations assigned to a user determine the user's access to labeled resource.

4.8.1 Label Component Definitions and Valid Characters

A sensitivity label is a single attribute, with multiple components. All resource labels must contain a level component; compartment and group components are optional. The administrator must define the label components before create labels.

Component	Description	Examples
Level	A single specification of the labeled resource's ordered sensitivity ranking	CONFIDENTIAL (1), SENSITIVE (2), HIGHLY SENSITIVE (3)
Compartments	Zero or more categories associated with the labeled resource	FINANCIAL, STRATEGIC, NUCLEAR
Groups	Zero or more identifiers of organizations owning or accessing the resource	REGION_1, REGION_2

Table 4-2 Sensitivity Label Components

Figure 4-7 illustrates the three dimensions in which resource can be logically classified, using levels, compartments, and groups.

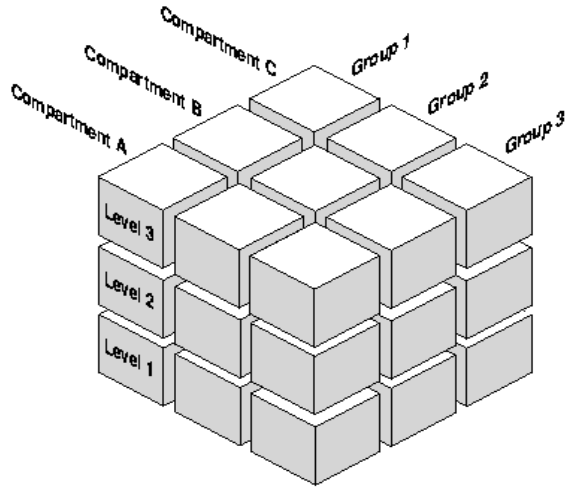


Figure 4-7 Resource Categorizations with Levels, Compartments, and Groups

4.8.2 How Resource Label and User Label Work Together

A user can only access resource within the range of his or his own label authorizations. A user has:

- Maximum and minimum levels
- A set of authorized compartments
- A set of authorized groups (and, implicitly, authorization for any subgroups)

For example, if a user is assigned a maximum level of SENSITIVE, then the user potentially has access to SENSITIVE, CONFIDENTIAL, and UNCLASSIFIED resource. The user has no access to HIGHLY_SENSITIVE resource.

Figure 4-8 shows how resource labels and user labels work together, to provide access control in Label Security. Whereas resource labels are discrete, user labels are inclusive. Depending upon authorized compartments and groups, a user can potentially access resource corresponding to all levels within his range.

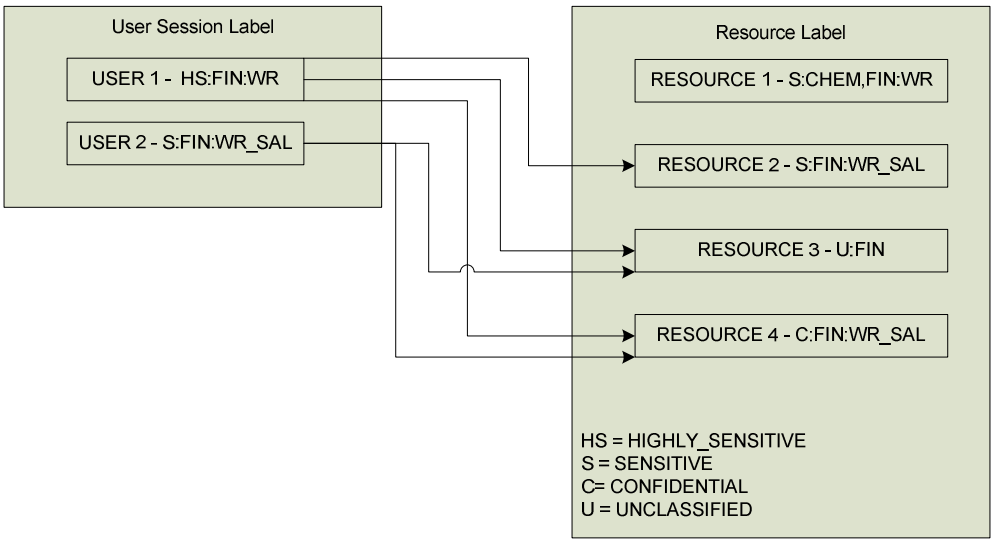


Figure 4-8 Example: Resource Labels and user Labels

As shown in the Figure 4-8, User 1 can access resources 2, 3, and 4 because his maximum level is HS; he has access to the FIN compartment; and his access to group WR hierarchically includes group WR_SAL. He cannot access resource 1 because he does not have the CHEM compartment. (A user must have authorization for *all* compartments in a resource label, to access that row.)

User 2 can access resource 3 and 4. His maximum level is S, which is less than HS in resource 2. Although he has access to the FIN compartment, he only has authorization for group WR_SAL. He cannot, therefore, access resource 1.

Figure 4-9 shows how resource pertaining to an organizational hierarchy fits in to resource levels and compartments.

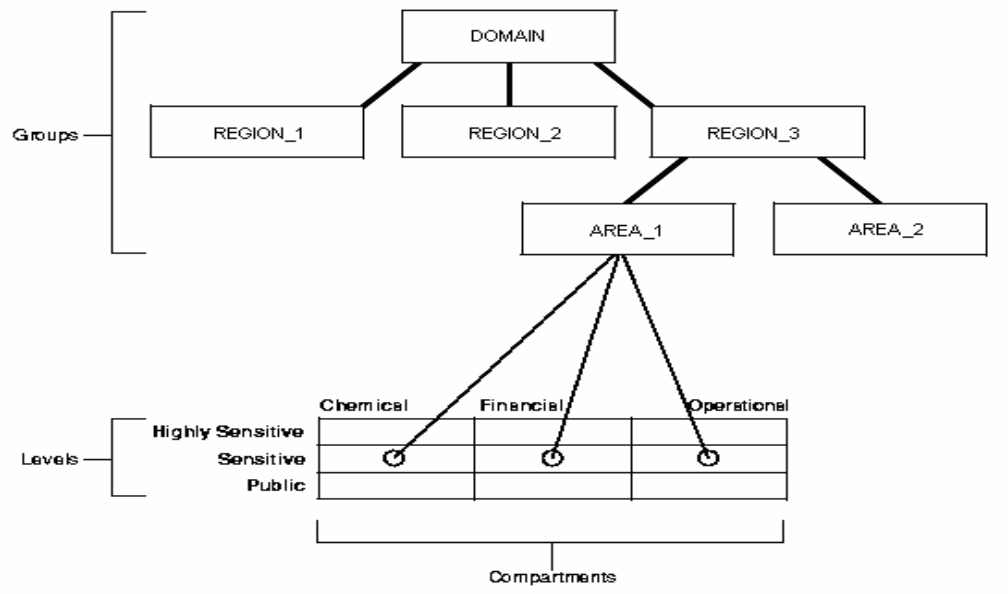


Figure 4-9 How Label Components Interrelate

For example, the DOMAIN group includes three subgroups: REGION_1, REGION_2, and REGION_1. The REGION_3 subgroup is further subdivided into AREA_1 and AREA_2. For each group and subgroup, there may be resource belonging to some of the valid compartments and levels within the network. Thus there may be SENSITIVE resource which is FINANCIAL, within the AREA_1 subgroup.

Note that a resource is generally labeled with a single group, whereas users' labels form a hierarchy. If users have a particular group, that group may implicitly include child groups. Thus a user associated with the REGION_3 group has access to all resource; but a user associated with AREA_1 would only have access to resource pertaining to that subgroup.

4.8.3 Access Mediation

To access a resource protected by a label security policy, a user must have authorizations based on the labels defined for the policy. Figure 4-10 illustrates the relationships between users, resource, and labels.

- Resource labels specify the sensitivity of resources.
- User labels provide the appropriate authorizations to users.
- Access mediation between users and resource depends upon their labels.

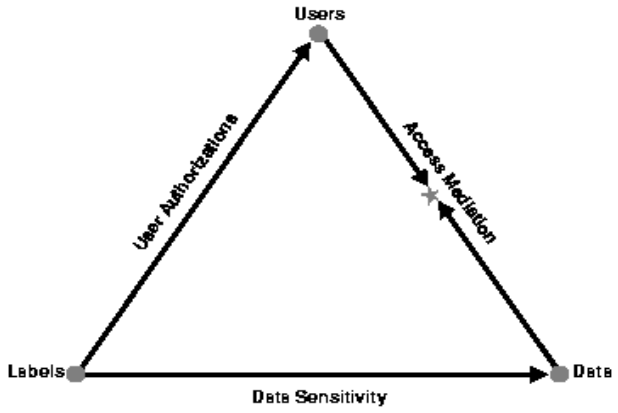


Figure 4-10 Relationships between Users, Resource, and Labels

4.8.4 How Labels Are Evaluated for Access Mediation

When a resource is protected by a label security policy, the user's label components are compared to the resource label components to determine whether the user can have access. In this way, security policy manager evaluates whether the user is

authorized to perform the requested operation on the resource. This section explains the rules and options by which user access is mediated.

4.8.4.1 Example of Read/Write Authorizations on Groups

When groups are organized hierarchically, a user's assigned groups include all subgroups that are subordinate to the group to which she belongs. In this case, the user's read/write authorizations on a parent group flow down to all the subgroups.

Consider the parent group REGION_1, with three subgroups as illustrated in Figure 4-11. If the user has read access to REGION_1, he also has read access to the three subgroups. The administrator can give the user write access to subgroup WR_FINANCE, without granting his write access to the REGION_1 parent group (or to the other subgroups). On the other hand, if the user has read/write access on REGION_1, then he also has read/write access on all of the subgroups subordinate to it in the tree.

Write authorization on a group does not give a user write authorization on the parent group. If a user has read-only access to REGION_1 and WR_FINANCE, the administrator can grant his write access to WR_ACCOUNTS_RECEIVABLE, without affecting his read-only access to the higher-level groups.

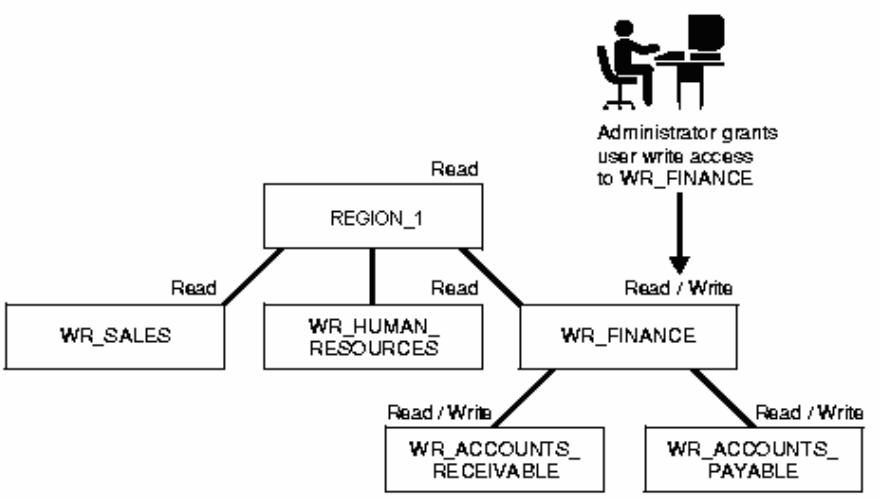


Figure 4-11 Subgroup Inheritance of Read/Write Access

4.8.4.2 Label Security Algorithm for Read Access

READ_CONTROL enforcement determines the ability to read a resource. The following rules are used, in the sequence listed, to determine a user's read access to a resource:

1. The user's level must be *greater than or equal to* the level of the resource.

2. The user's label must include *at least one of the groups* which belong to the resource (or the parent group of one such subgroup).

3. The user's label must include *all the compartments* which belong to the resource.

If the user's label passes these tests, it is said to "dominate" the resource's label.

Note that there is no notion of read or write access connected with levels. This is because the administrator specifies a range of levels (minimum to maximum) within

which a user can potentially read and write. At any time, the user can read all resources equal to or less than his current session level. No privileges (other than FULL) allow the user to write below his minimum authorized level.

The label evaluation process proceeds from levels to groups to compartments, as illustrated in Figure 4-12. Note that if the resource label is null or invalid, then the user is denied access.

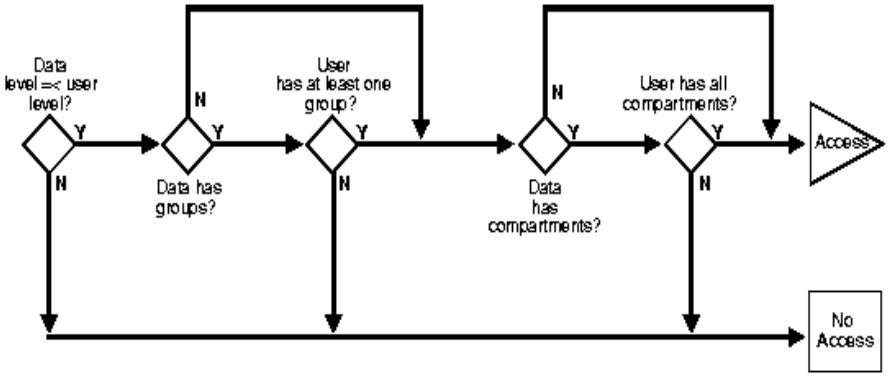


Figure 4-12 Label Evaluation Process for Read Access

As a read access request comes in, security policy manager evaluates each resource determine:

1. Is the user's level equal to, or greater than, the level of the resource?

2. If so, does the user have access to at least one of the groups present in the resource label?

3. If so, does the user have access to all the compartments present in the resource label?

If the answer is no at any stage in this evaluation process, then security policy manager denies access to the resource, and moves on to evaluate the next resource.

4.8.4.3 Label Security Algorithm for Write Access

WRITE_CONTROL enables network admin control resource access with ever finer granularity. Granularity increases when compartments are added to levels; it increases again when groups are added to compartments. Access control becomes even finer grained when network admin can manage the user's ability to write the resource which he can read.

To determine whether a user can write a particular resource, security policy manager evaluates the following rules, in the order given:

1. The level in the resource label must be greater than or equal to the user's minimum level and less than or equal to the user's session level.

2. When groups are present, the user's label must include *at least one of the groups with write access* which appear in the resource label (or the parent of one such subgroup). In addition, the user's label must include *all the compartments* in the resource label.

3. When no groups are present, the user's label must have write access on *all of the compartments* in the resource label.

Just as with read operations, the label evaluation process proceeds from levels to groups to compartments. Note that the user cannot write any resource below his authorized minimum level, nor above his current session level. The user can always read below his minimum level.

Figure 4-13 illustrates how the process works with write operation. Note that if the resource label is null or invalid, then the user is denied access.

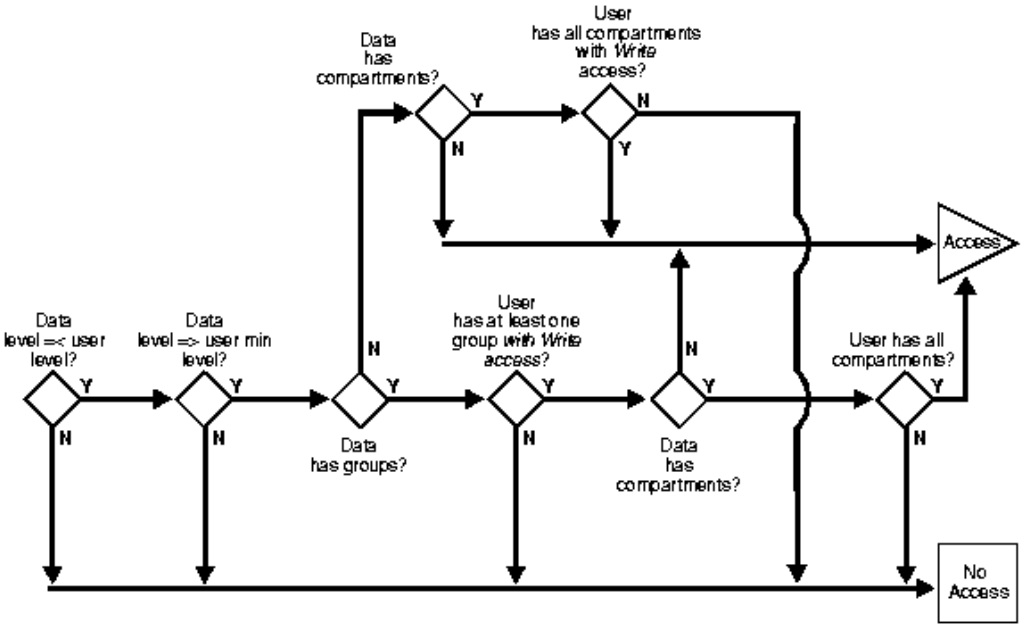


Figure 4-13 Label Evaluation Process for Write Access

As an access request comes in, security policy manager evaluates each resource to determine:

1. Is the resource's level equal to, or less than, the level of the user?

2. Is the resource's level equal to, or greater than, the user's minimum level?

3. If the resource's level falls within the foregoing bounds, does the user have write access to at least one of the groups present in the resource label?

4. If so, does the user have access to all the compartments with at least read access which are present in the resource label?

5. If there are no groups, but there are compartments, then does the user have write access to all of the compartments?

If the answer is no at any stage in this evaluation process, then security policy manager denies access to the resource, and moves on to evaluate the next resource.

In addition, each user may have an associated minimum level below which he cannot write. He cannot write any resource labeled with levels below his minimum, nor can he write any resource with a resource label containing a level less than his minimum.

4.9 Policy Management Language

All the policy defined above need to be stored in the network. Flexibility and scalability are necessary for the format to be chosen. XACML (the Extensible Access Control Markup Language) can be used to meet the requirements. In general, XACML describes two key areas for security -- an access control policy language and a request/response language for two-way communications [35].

At the root of XACML is a concern with access policies -- what XACML refers to as a Policy or a Policy Set. When XACML refers to "policy," it specifically means Authorization (AuthN) Policy.

Each XACML policy document contains exactly one Policy or Policy Set root XML tag. A Policy Set is a container that can hold other Policies or Policy Sets, as well as references to policies found in remote locations. A Policy represents a single access-control policy, expressed through a set of Rules.

XACML defines and describes "layering" between XML entities to clearly distinguish between security technologies that:

1. Create policy;
2. Collect the data required for policy evaluation;
3. Evaluate policy; and
4. Enforce policy.

Because a generic Policy or Policy Set may contain multiple policies or Rules, each of which may evaluate to different access control decisions, XACML needs some way of reconciling the decisions each makes. In XACML, this is done through a collection of Combining Algorithms. Each algorithm represents a different way of combining multiple decisions into a single decision. XACML utilizes Policy Combining Algorithms (used by Policy Set) and Rule Combining Algorithms (used by Policy).

The Deny Overrides Algorithm is an example of these indicating that no matter what, if any evaluation returns Deny, or no evaluation permits, the final result is also Deny. These Combining Algorithms are used to build up increasingly complex policies

For Policy creation/enforcement, XACML brings several features, including:

- The ability to include almost any property of any of the participants (or component) of the environment, not just the attributes of the requester;
- The ability to use data manipulation and Boolean operators (in combination) to calculate the policy effect. This is especially useful in complex policies with time, location, dollar amount or other multiple dependencies; and
- The ability to protect any sort of resource, with special handling for the important cases of hierarchical namespaces and portions of XML documents.

For scalability, XACML brings:

- The ability to independently administer multiple policies controlling access to the same resources;

- The ability to select (or define) algorithms for reconciling conflicting policies; and
- The ability to efficiently locate all the policies that are potentially applicable to a given decision without sacrificing the flexibility described above.

Figure 4-14 demonstrates a sample policy presented in XACML format.

```
<Policy PolicyId="SamplePolicy"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">

   <!-- This Policy only applies to requests on the SampleServer -->
   <Target>
     <Subjects>
       <AnySubject/>
     </Subjects>
     <Resources>
       <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SampleServer</AttributeValue>
         <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
       </ResourceMatch>
     </Resources>
     <Actions>
       <AnyAction/>
     </Actions>
   </Target>

   <!-- Rule to see if we should allow the Subject to login -->
   <Rule RuleId="LoginRule" Effect="Permit">

     <!-- Only use this Rule if the action is login -->
     <Target>
       <Subjects>
         <AnySubject/>
       </Subjects>
       <Resources>
         <AnyResource/>
       </Resources>
       <Actions>
         <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">login</AttributeValue>
           <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                              AttributeId="ServerAction"/>
         </ActionMatch>
       </Actions>
     </Target>

     <!-- Only allow logins from 9am to 5pm -->
     <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal"
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
           <EnvironmentAttributeSelector DataType="http://www.w3.org/2001/XMLSchema#time"
                                AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
         </Apply>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">09:00:00</AttributeValue>
       </Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal"
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
           <EnvironmentAttributeSelector DataType="http://www.w3.org/2001/XMLSchema#time"
                                AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
         </Apply>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">17:00:00</AttributeValue>
       </Apply>
     </Condition>

   </Rule>

   <!-- We could include other Rules for different actions here -->

   <!-- A final, "fall-through" Rule that always Denies -->
   <Rule RuleId="FinalRule" Effect="Deny"/>

 </Policy>
```

Figure 4-14 A sample policy in XACML format

4.10 Performance Analysis

Dynamic Source Routing protocol (DSR) and Ad Hoc On-Demand Distance Vector protocol (AODV) are two of the most commonly used protocols in Ad Hoc network routing. We are using DSR and AODV as base protocols, and apply proposed Policy Based Security module as plug-in to evaluate the routing overhead generated by the extra security layer.

4.10.1 Simulation Model

The implementations of AODV and DSR in our simulation environment closely match their specifications. The routing protocol model detects all data packets transmitted or forwarded, and responds by invoking routing activities as appropriate. The RREQ packets are treated as broadcast packets in the MAC. RREP and data packets are all unicast packets with a specified neighbor as the MAC destination. RERR packets are treated differently in the two protocols. They are broadcast in AODV and use unicast transmissions in DSR. Both protocols detect link breaks using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop.

Both protocols maintain a send buffer of 64 packets. It contains all data packets waiting for a route. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 seconds. All packets sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has maximum size of 50 packets and is maintained as a priority queue

with three priorities each served in FIFO order. Routing packets get higher priority than security packets, and security packets get higher priority than data packets.

The security management module is created by Object-oriented Tool Control Language (OTcl) as a plug-in implemented above the network layer. It generates and acknowledges all security related requests which have been discussed in the previous section.

4.10.2 Simulation Assumptions

We are using following assumptions in our simulations:

- Nodes are randomly moving in a pre-defined two-dimension area.
- Nodes have adequate memory to store require security policy data.
- Nodes have adequate CPU power to handle security authentications.
- Security policy of each node is pre-defined during the entire simulation.
- Only security authentications overhead will be simulated, the overhead of security policy synchronization among each node will not be considered.
- Security packets are considered as part of routing packets vs. data packets to calculate routing overhead.

4.10.3 Traffic and Mobility models

We use traffic and mobility models similar to those previously reported using the same simulator. Traffic sources are CBR (continuous bit-rate). The source-destination pairs are spread randomly over the network. Only 512 byte data packets are used. The

number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network.

The mobility model uses the random waypoint model in a rectangular field. We use 1500m x 300m field with 50 nodes. Each node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0-20 m/sec). Simulation period is 900 seconds. Each data point represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios. Identical mobility and traffic scenarios are used across protocols.

4.10.4 Metrics

In comparing the protocols, we chose to evaluate them according to the following two metrics:

Packet delivery ratio: the ratio between the number of packets originated by the application layer CBR sources and the number of packets received by the CBR sink at the final destination.

Routing overhead: the total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet (each hop) counts as one transmission.

Packet delivery ratio is important as it describes the loss rate that will be seen by the transport protocols, which in turn affects the maximum throughput that the network can support. This metric characterizes both the completeness and the correctness of the routing protocol.

Routing overhead is an important metric for comparing these protocols, as it measure the scalability of a protocol, the degree to which it will function in congested or low bandwidth environments, and its efficiency in terms of consuming node battery power. Protocols that send large numbers of routing packets can also increase the probability of packet collisions and may delay data packets in network interface transmission queues.

4.10.5 Simulation Results

Figure 4-15 and 4-16 highlight the relative performance of the four routing protocols on our traffic loads of 20 sources.

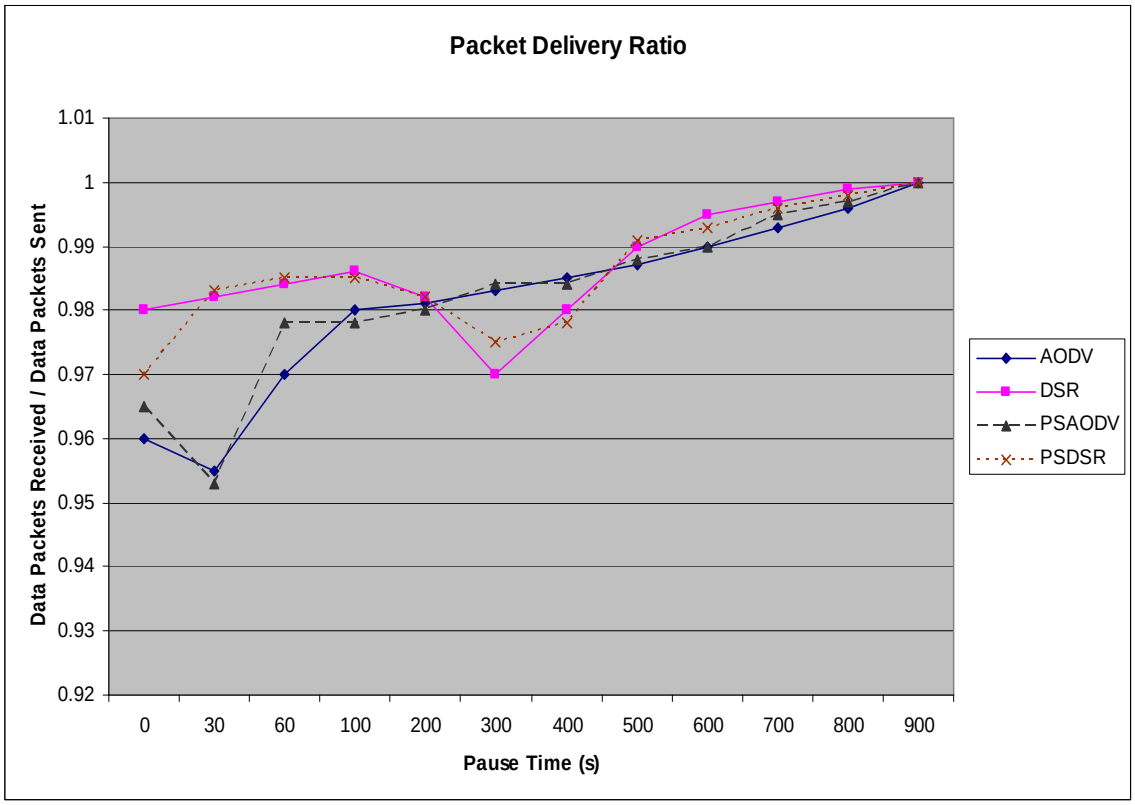


Figure 4-15 Packet Delivery Ratios

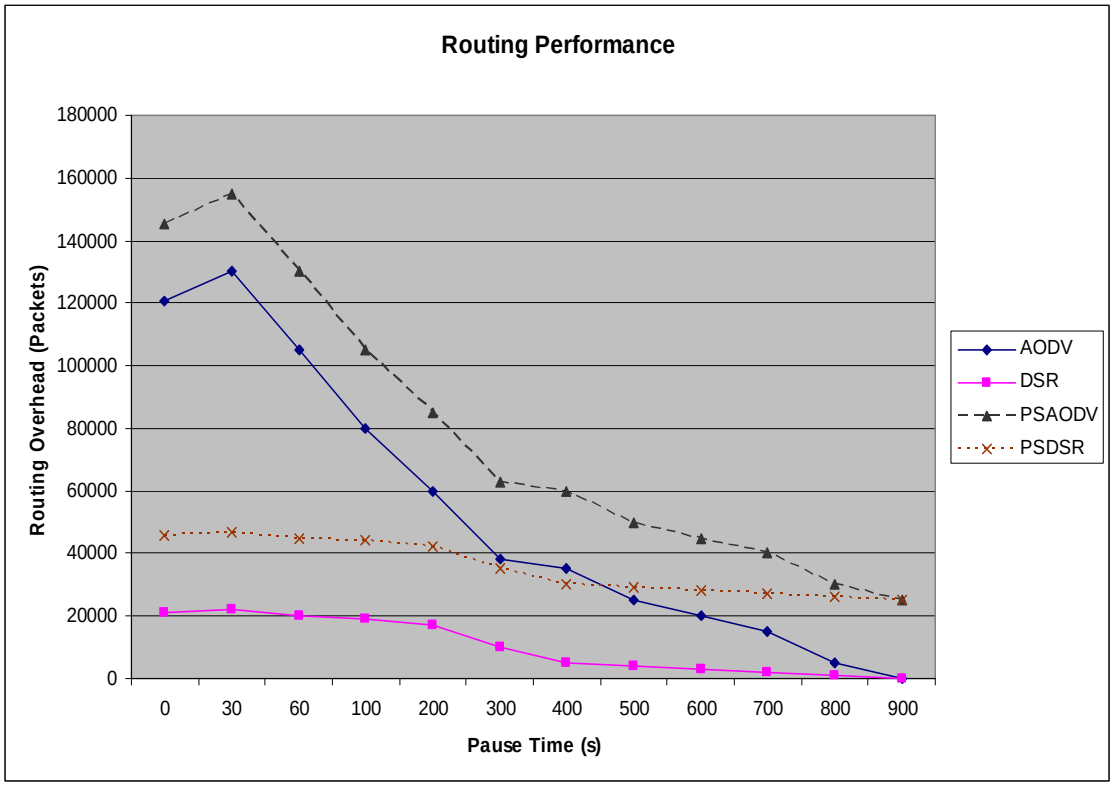


Figure 4-16 Routing Performance

All of the protocols deliver a great percentage of the originated data packets when there is little node mobility (at large pause time); converging to 100% delivery when there is no node motion. The regular DSR and AODV can deliver over 95% of the data packets regardless of mobility rate. The proposed the policy based secure DSR (PSDSR) and policy based secure AODV (PSAODV) can achieve very close delivery ratio compared with the original protocols.

The four routing protocols impose vastly different amount of overhead. DSR has overall better performance than AODV. The proposed secure protocols apparently

generate more overhead than original protocols, the routing packets increase about 10-15% in secure protocols.

The actual memory usage for the proposed security policy management in our simulation is not high, it remains below 1 megabyte. The distributed security policy synchronization process is not considered as overhead in our simulation. Because we believe in the real world scenario, the security policies don't change often, and it usually can be done at off peak.

4.10.5 Conclusion

The simulation results indicate the proposed Policy Based Security approach has almost no impact on the packet delivery ratio of the original routing protocol, but it does result in an increase of an average of 10-15% of routing overhead regardless of mobility. This is because the proposed approach generates extra security packets which we consider as part of the routing packets as opposed to data packets.

Chapter 5

Multi-Layer QoS Interface Guided Routing

5.1 Introduction

The QoS routing algorithms for wired networks cannot be applied directly to Ad Hoc networks. First, the performance of most wired routing algorithms relies on the availability of precise state information. However, the dynamic nature of an Ad Hoc network makes the available state information inherently imprecise. Though some recent algorithms [28][25] were proposed to work with imprecise information (e.g., the probability distribution of link delay), they require precise information about the network topology, which is not available in an Ad Hoc network. Second, nodes may join, leave, and rejoin an Ad Hoc network at any time and any location; existing links may disappear, and new links may be formed as the nodes move. Hence, the established paths can be broken at any time, which raises new problems of maintaining and dynamically reestablishing the routing paths in the course of data transmission.

Recently, cross-layer design approaches [26] - [29] has been introduced into ad hoc wireless network to resolve the above issues. The cross-layer protocols are designed by violating the seven-layer open systems interconnect (OSI) model to provide overall better efficiency and performance in ad hoc wireless environment. Here the functionality of multiple layers are condensed into fewer layers with the view to improving

performance. The cross-layer designs are still at a very early research stage with lots of studies yet to be done.

Instead of a cross-layer design approach, we choose a “hybrid” approach, which will still retain the seven-layer OSI model, but define an extra QoS interface for each layer to provide better “hand shake”. Compared with the cross-layer designs which require radical and complex changes to the protocol architecture, our approach is much easier to accomplish and can be implemented on existing systems while providing improved QoS management and performance. Furthermore, the proposed holistic approach is novel as it considers the different factors that contribute to QoS at the different layers in contrast to traditional QoS routing protocols which work primarily on ensuring that the QoS requirements are satisfied at a specific level.

5.2 Multi-Layer QoS Interface Guided Routing

We propose a multi-layer QoS interface guided routing, which separates metrics at the different layers, MAC layer metrics, network layer metrics, and application layer metrics. We believe the QoS requirements of an application is different from the QoS requirements of the network, but depends on the quality of the network. In our model, each layer manages its own QoS and communicates with other layers through its QoS interface.

At the application layer, we propose to classify the QoS requirements into a set of QoS priority levels with their corresponding application layer metrics. For example, we classify application requirements into three QoS level services. Level I guaranteed service corresponds to applications that have strong delay constraints, such as voice. Level II controlled load service is suitable for applications requiring high throughput such as video broadcasting applications. Level III best effort service has no specific constraints.

At the network layer, we recommend using nodes' hop count state, buffer state, and stability state to characterize the quality of network, and we call them network layer metrics. The hop count represents the number of hops required to a packet to reach its destination. The buffer state stands for the available unallocated buffer. The stability indicates the connectivity variance of a node with respect to its neighboring nodes over time. In our algorithm, we use this metric of each node in one specific route to calculate the path quality.

At the MAC layer, the quality of network could mean line signal to noise and interference ratio (SINR), and we call it MAC layer metrics. Link SINR determines the communication performance of the link: the data rate and associated probability of packet

error rate or bit error rate (BER) that can be supported by the link. Links with low SINR are not typically used due their poor performance, leading to partial connectivity among all nodes in the network. Moreover, it is essential to minimize the volume of traffic being transmitted over the wireless interface because of the lack of wireless resources. This can be achieved via our interface mapping algorithm.

In each layer, the layer specific QoS interface accepts requirements from higher layer, and translates into layer metrics. For example, network layer QoS interface accepts throughput service requirements from application layer, and translates into network layer metric such as buffer, power, and stability requirements.

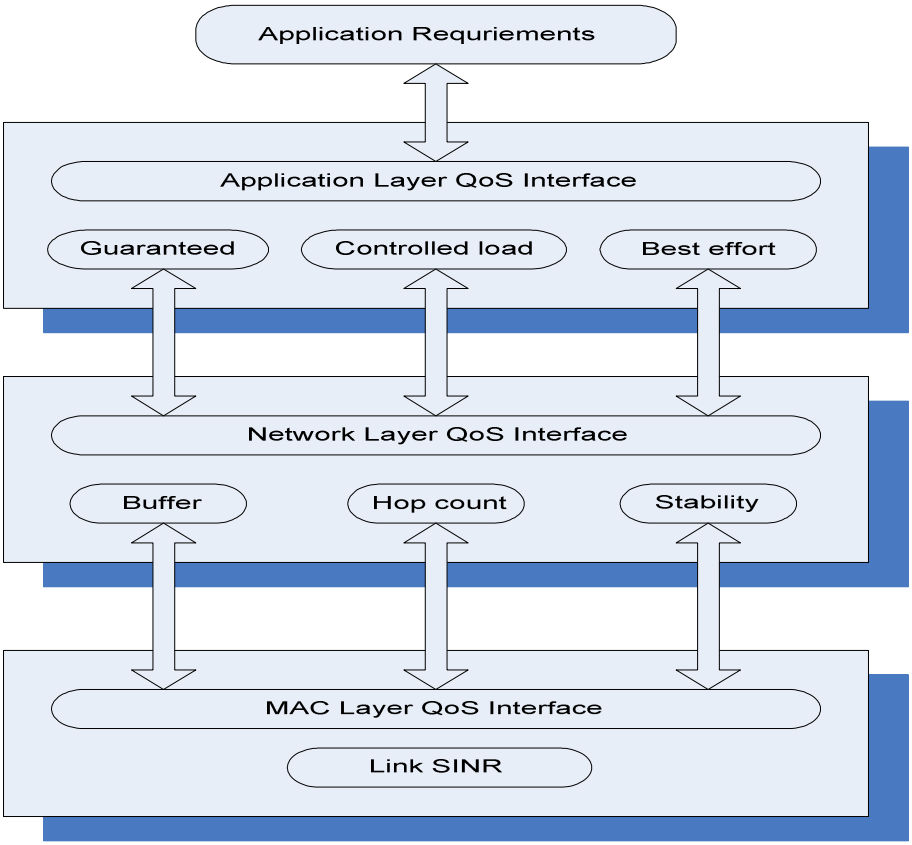


Figure 5-1 Network Layer Structure and QoS Metrics Mapping

We utilize the QoS interface metrics defined above to guide the routing process,

which includes:

- Path generation – generates paths according to the assembled and distributed state information of the network and application.
- Path selection – selects appropriate paths based on the network and application state information.
- Data forwarding – forwards user traffic along the selected path.

Path generation is a process in which the quality of a path to route the data traffic is computed suing the quality of individual node in the path. The quality should not only reflect the available resources residing both in the wireless medium and in each node but also the stability of these resources. We use buffer level and stability level to characterize the quality of a node, and call them network layer metrics. With the knowledge of the quality of paths, an application selects the most suitable path according to the desired QoS level. For this purpose, application requirements are classified into three QoS level including guaranteed service, throughput service, and best effort service. In order to be able to compute these metrics, a reasonable combination of network layer metrics is mapped into the application layer metrics, which we defined as a QoS interface. Figure 5-1 shows the mapping between QoS layers.

In order to keep the routing overhead low and support fast routing decisions in QoS routing, we associate a state to the available network resources. In the path generation phase, the nodes use the state information to generate paths according to the available network resources. Then in the path selection phase, this state is used in conjunction with the desired QoS level to select the most suitable path according to the application requirements. The model differentiates services and provides soft guarantees

to network resources for an admitted application by using a class-based weighted fair queuing (CB-WFQ) at intermediate nodes.

5.3 Path generation

Unlike fixed networks such as the internet, QoS support in Ad Hoc networks depends not only on the available network resources but also on the nodes' mobility rate. This is because mobility may result in link failure which in turn may result in a broken path. Furthermore, Ad Hoc networks potentially have fewer resources than fixed networks. Therefore, more criteria are required in order to capture the quality of the links between nodes. We propose to measure the quality of network and use it in the path generation process. We define network layer metrics to determine the quality of network in order to generate the good quality path.

The main objective of network layer metric is to provide a trade-off between load balancing and resource conservation. We define three network layer metrics: buffer level, hop count, and stability level. We assume that a node periodically broadcasts its network layer metrics to its neighbors, indicating its presence and its QoS state.

Hop Count – corresponds to the number of hops required to a packet to reach its destination. Note that the hop count metric is related to resource conservation, since a path with fewer hops is preferable.

Buffer level – indicates the available unallocated buffer. This metric is related to load balancing. If the buffer level of a node is low, it implies that a large number of packets are queued up for forwarding, which implies that a packet routed through this node would have to experience high queuing delays. We use high, medium, low QoS states to represent the buffer level. A high QoS state indicates that the corresponding node has no packets queued up for forwarding. Since there is a delay between the broadcast of this metric and its use, instantaneous buffer level may be misleading. Hence, a node

should maintain the average buffer level. Exponentially weighted moving average (EWMA) may be used.

Stability level – we define the connectivity variance of a node with respect to its neighboring nodes over time as the stability of that node. The metric is used to avoid an unstable node to relay packets. We calculate the stability S of a node n as:

$$S(n) = \frac{| N_{t_i} \bigcap N_{t_{i+1}} |}{| N_{t_i} \bigcup N_{t_{i+1}} |}$$

Where N_{t_i} and $N_{t_{i+1}}$ represent the nodes as neighbor of n at time t_i and t_{i+1} respectively. A node is unstable if a large number of its neighbors change. On the other hand, if most of the neighbors remain the same at the two times t_i and t_{i+1}, then we call this node stable. A node has high stability if none of its neighbors change ($N_{t_i} = N_{t_{i+1}}$), in this case we have S(n) = 1. A node is unstable if all its neighbors change ($N_{t_i} \bigcap N_{t_{i+1}} = \phi$), in this case we have S(n) = 0. We define a node stability level as below:

LOW $0 <= S(n) < \alpha$

MEDIUM $\alpha <= S(n) < \beta$

HIGH $\beta <= S(n) <= 1$

Where:

$0 < \alpha < \beta < 1$

In the path generation phase, network layer metrics are propagated through the nodes of the generated path. Suppose *P* is a path between source node *s* and destination node *d*, where *P* is a sequence of nodes, $P = \{s, n_1, n_2, ..., n_i, d\}$. The formulas to calculate

the value of metrics of P are:

$$P.hop = \sum_{n \in P} 1$$

$$P.buffer = \min_{n \in P}(n.buffer)$$

$$P.stability = \min_{n \in P}(n.stability)$$

The buffer level of *P* is represented by the node with the least buffer in *P*. This is appropriate for the route generation process, since a route is rendered broken even if one intermediate node has no buffer. Similarly, the stability level of *P* can also be calculated by the node with least stability on *P*. However, the buffer level of *P* can also be calculated as the average over the buffer levels of the all the nodes in *P*:

$$P.buffer = (\sum_{n \in P} n.buffer) / P.hop$$

At the MAC layer, the quality of network could mean line signal to noise and interference ratio (SINR), and we call it MAC layer metrics. Link SINR determines the communication performance of the link: the data rate and associated probability of packet error rate or bit error rate (BER) that can be supported by the link. Links with low SINR are not typically used due their poor performance, leading to partial connectivity among all nodes in the network. Moreover, it is essential to minimize the volume of traffic being transmitted over the wireless interface because of the lack of wireless resources. This can be achieved via our network layer to MAC layer interface mapping algorithm.

Our algorithm will be greedy in that the information will be transmitted to the node which has the highest SINR, which means no matter what network layer QoS

requirements are, the algorithm always tries to choose the highest SINR nodes available to generate the path, unless the node buffer is full. On the other hand, as soon as one node buffer reaches full condition, the algorithm will suggest a lower QoS level path to use lower SINR node to protect the high QoS level path and thereby perform load balancing.

5.4 Path selection

In order to incorporate application requirements in the path selection process, they need to be translated into QoS metrics that specify the application QoS constrains. Then a reasonable combination of network layer metrics is mapped into each QoS metrics. Furthermore, the MAC layer metrics are mapped into each network metrics.

We define three QoS levels at the application layer, namely, guaranteed service, controlled load service, and best effort service. Guaranteed service corresponds to applications that have strong delay constraints, for example applications with real-time traffic such voice. The network layer QoS interface will translate this requirement into network QoS metric, which will select a path that has minimum delay based on the average buffer level and hop count. Controlled load service is suitable for applications requiring high throughput such as video broadcasting. The network QoS interface needs to pick the highest buffer size path in this case to meet the application layer QoS requirements. Best effort service has no specific constraints. The network QoS interface will need select the most stable path and the shortest path. In fact, it selects the most stable path when the network mobility is high and the shortest path when the network mobility is low. Table 5-1 shows the mapping between each layer QoS metric.

Application layer	Network layer	MAC layer
Guaranteed	Buffer & hop count	SINR
Controlled load	Buffer & hop count	SINR
Best effort	Stability & hop count	SINR

Table 5-1 QoS metrics mapping table

Guaranteed – guaranteed service defines the maximum latency required by the application. The total latency is experienced by a packet to traverse the network from the source to the destination. At the network layer, the end-to-end packet latency is the sum of processing delay, transmission delay, queuing delay, and propagation delay. Queuing delay contributes most significantly to the total latency and all other delays are negligible. Therefore, the packet latency can be calculated as:

$$P.latency = \sum_{n \in P} [(n.bufferSize - n.buffer) / n.throughput]$$

Where n.buffer is the node unallocated buffer level, (n.bufferSize – n.buffer) denotes the node buffer occupancy. The formula can also be represented as:

$$P.latency = P.hop \times (P.bufferSize - P.buffer) / P.throughput$$

Where P.buffer denotes the path unallocated buffer level, and (P.bufferSize – P.buffer) denotes the path buffer occupancy.

Controlled load – controlled load service define the minimum throughput required by the application. The throughput is the defined as the rate at which packets are transmitted in the network. The throughput for an end-to-end connection can be estimated as:

$$P.throughput = \frac{MIN(n.throughput)}{\sum_{n \in P} (n.bufferSize - n.buffer)} = \frac{MIN(n.throughput)}{P.hop \times (P.bufferSize - P.buffer)}$$

Best Effort – best effort service provides no service guarantees for the applications. It selects between the most stable path in the high mobility case and the

shortest path in the low mobility path case. In our model, it uses P.stability to determine which path to choose.

5.5 QoS interface

We use QoS interfaces to translate high layer QoS metrics to lower layer metrics. For instance, the QoS interface between the application layer and the network layer (AN Interface) will translate guaranteed service requirements into network layer buffer level and hop count requirements. Furthermore, the QoS interface between network layer metrics and the MAC layer (NM Interface) will use network buffer level; hop count and stability number to determine the path SINR requirements at the MAC layer.

Application layer	AN Interface	Network layer	NM Interface	MAC layer
Guaranteed	Path latency	Buffer & hop count	Greedy algorithm	SINR
Controlled load	Path throughput	Buffer & hop count	Greedy algorithm	SINR
Best Effort	Path stability	Stability & hop count	Greedy algorithm	SINR

Table 5-2 QoS interfaces mapping table

For guaranteed service, the AN interface translates the QoS requirements to the maximum path latency, and pass to network layer as application layer QoS requirements. During the path selection process, the network lay will choose the qualified path by using the calculations defined in the last section, and using the network layer metrics as input parameter.

For controlled load service, the AN interface translates the QoS requirements to the minimum path throughput, and pass to network layer as application layer QoS requirements. Network layer will choose the qualified path by calculate the path buffer level and hop count.

For best effort service, the AN interface compromises between the most stable path in high mobility case and shortest path in low mobility path case. In the former case, it applies the stability metric to establish the most stable path from the source to the destination in order to improve delay performance due to path failure caused by the node mobility. In the latter case, it use hop count metric in order to minimize network resource utilization since the more hops a flow traverses, the more resources it consumes.

Our NM interface uses a greedy method to insure that the information will be transmitted to the node which has the highest SINR, which means no matter what network layer QoS requirements are, the algorithm always tries to choose the highest SINR nodes available to generate the path, unless the node buffer is full. On the other hand, as soon as one node buffer reaches full condition, the algorithm will suggest the lower QoS level path to use lower SINR node to protect the high QoS level path and perform load balancing.

5.6 Performance Analysis

The performance of the proposed QoS routing protocol is studied with simulations.

5.6.1 Simulation Model

The QoS routing protocol has been implemented with ns2 [36]. The implementation is based on AODV module contributed by the MONARCH group from CMU, and the QoS routing functions are added. In additional to building QoS routes, the protocol also builds a best-effort route when it learns such a route. The best-effort route is used when a QoS route is not available. The Evolutionary-TDMA scheduling protocol (E-TDMA) [37] developed by the same authors is used at the MAC layer. It is distributed protocol which dynamically generates and updates TDMA transmission schedules among the nodes. Transmission rate is 1 Mbps. There are 40 slots in a frame, and a slot carries 32 bytes of information. A packet needs to be transmitted in multiple slots if it cannot fit in one slot. Limited contention is used for nodes to make their time slot reservations, hence E-TDMA a mainly limited by nodal density rather than network size. Considering the overhead of making reservation, an information slot is equivalent to 18 kbps. Details of E-TDMA can be found in [37]. In the simulations, Route_setup_time = 1000 ms and Route_life_time = 200 ms.

The implementations of AODV and QoS-AODV in our simulation environment closely match their specifications. The routing protocol model detects all data packets transmitted or forwarded, and responds by invoking routing activities as appropriate. The

RREQ packets are treated as broadcast packets in the MAC. RREP and data packets are all unicast packets with a specified neighbor as the MAC destination. RERR packets are broadcast in both AODV and QoS-AODV. Both protocols detect link breaks using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop.

Both protocols maintain a send buffer of 64 packets. It contains all data packets waiting for a route. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 seconds. All packets sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has a maximum size of 50 packets and is maintained as a priority queue with three priorities each served in FIFO order. Routing packets get higher priority than security packets, and security packets get higher priority than data packets.

Our multi-layer QoS interface guided routing protocol are implemented based on the existing QoS-AODV protocol in ns2. By expanding aodv.cc module in ns2, we add four more parameters in this module: node SINR, node buffer, node stability and link hop count. The detail algorithms of these parameters have been discussed in previous sections.

5.6.2 Traffic and Mobility models

We use traffic and mobility models similar to those previously reported using the same simulator. Traffic source are CBR (continuous bit-rate). The source-destination pairs are spread randomly over the network. Only 512 byte data packets are used. The

number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network.

The mobility model uses the random waypoint model in a rectangular field. We use 1500m x 300m field with 50 nodes. Each node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0-20 m/sec). Simulation period is 900 seconds. Each data point represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios. Identical mobility and traffic scenarios are used across protocols.

5.6.3 Simulation Results

The multi-layer QoS AODV routing protocol (mQoS AODV) is compared with the QoS AODV and AODV protocols.

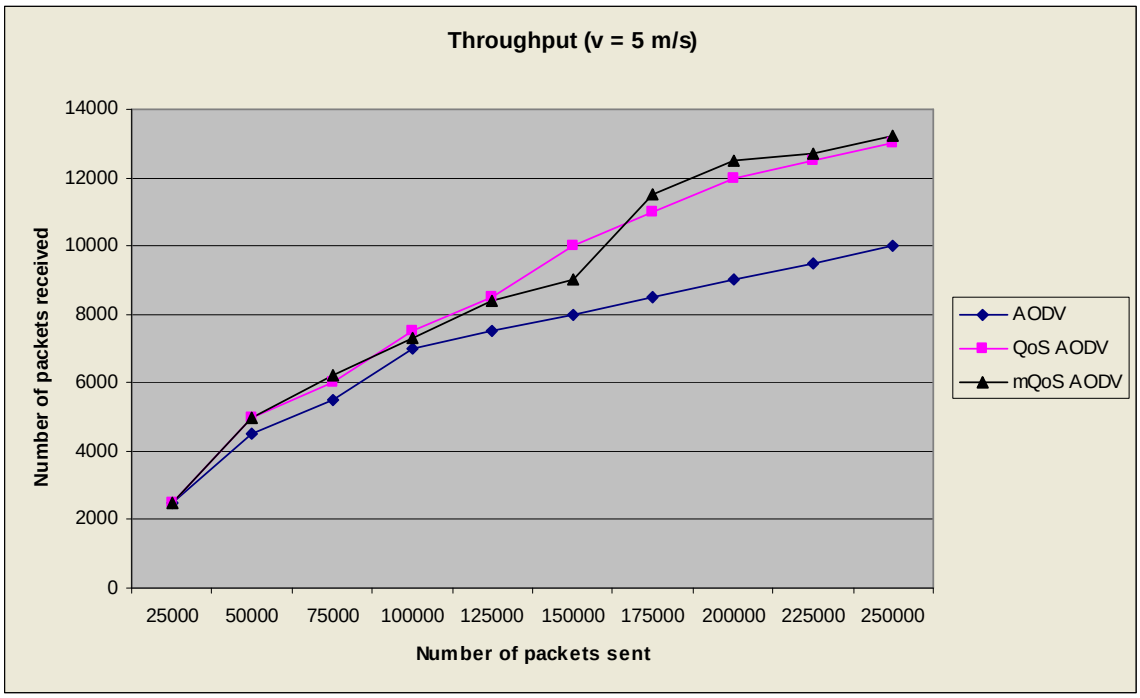


Figure 5-2 Throughput for v = 5 m/s

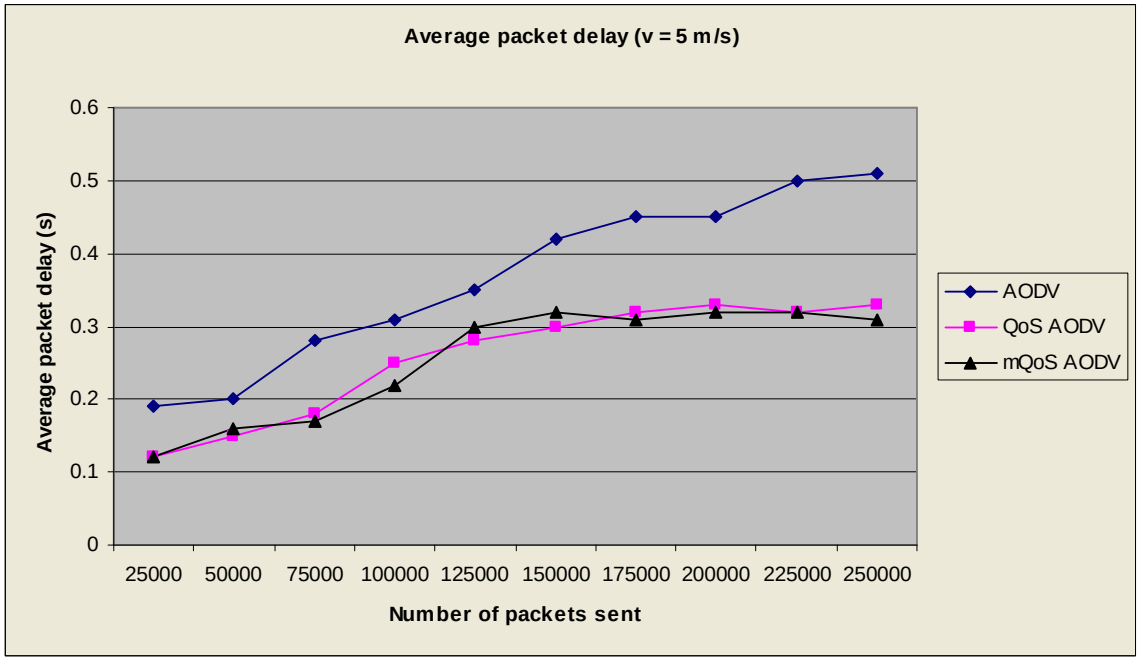


Figure 5-3 Average packets delay for v = 5 m/s

Figures 5-2 and 5-3 show the packet throughput and the average packet delay under different traffic loads in low mobility. Under light traffic, packet throughput and packet delay are very close for all three protocols, because they often use the same routes.

The advantage of QoS routing protocols become apparent when traffic gets heavy. With the AODV protocol, a node has one active route to a destination and uses it for all the packets to the destination. As the network traffic becomes heavy, this route becomes heavily loaded, causing packets to be delayed and dropped. The average packet delay increases significantly under heavy traffic. On the other hand, the QoS routing protocols try to find and use routes satisfying bandwidth constraints for different flows, even between the same pair of source and destination. Two QoS routes may share the same path, but the protocol will ensure enough bandwidths are reserved on this path to accommodate both flows. The traffic load is more balanced this way. The average packet delay increases with offered load slowly with the QoS routing protocols. There is not much difference between two QoS protocols in low mobility.

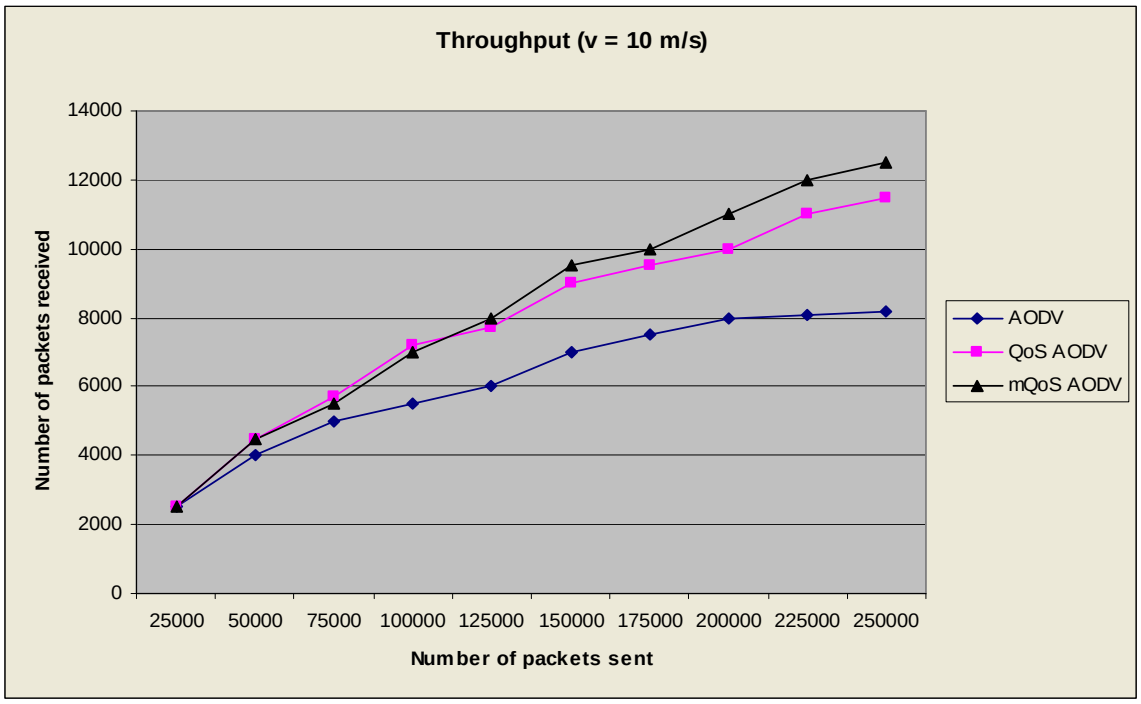


Figure 5-4 Throughput for v = 10 m/s

Figure 5-4 shows when mobility increases, the throughput of all protocols drops. Mobility affects network throughput at both the MAC layer and the routing layer. At the MAC layer, it takes time for E-TDMA to resolve the collisions caused by node movement and to reserve new slots. Essentially a protocol like E-TDMA which is based on establishing reservation has only limited capability to handle network mobility and is best for a static network. At the network layer, it takes time for the routing protocol to re-establish a route when it breaks. For the QoS routing protocols, the packet throughput drops roughly by 15% at v=10 m/s, compare with v=5 m/s.

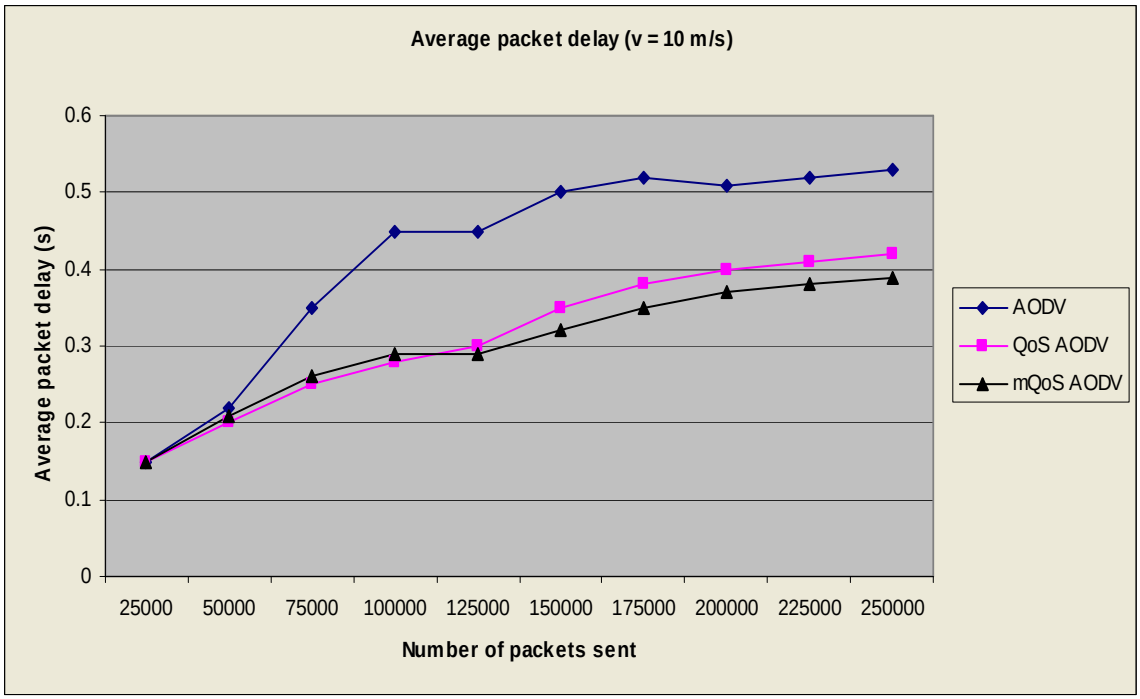


Figure 5-5 Average packet delay for v = 10 m/s

Figure 5-5 shows mobility also increases the average packet delay. The average packet delay increases roughly 50% at v=10 m/s, compare with v=5 m/s. Interestingly, when we compare the three routing protocols under mobility, the advantage of QoS routing protocols increases. Because the QoS routing protocols use different QoS routes for individual flows, when one of the QoS routes breaks, only this QoS route is repaired. Other are not affected. Packets of the flow on the broken route are temporarily forwarded using the best-effort route, which may coincide with one of the other QoS routes. There is more route redundancy with QoS routing (at the cost of increased routing table size). In the AODV protocol, when the only route to a destination breaks, all packets addressed to this destination are delayed or dropped. It can be expected that a best-effort routing protocol which finds multiple routes will be better than AODV in this aspect.

Also our proposed multi-layer QoS routing protocol performs better than

traditional QoS routing protocol during high mobility, because it's always looking for more reliable paths during the path selection phase. The trade off is each node requires more memory to store path quality data.

5.6.4 Conclusion

The simulation results indicate the proposed multi-layer QoS routing protocol can produce higher throughput and lower delay than traditional QoS routing protocols in a high mobility ad hoc network environment. There is not much improvement under low network mobility. More internal memory is required for each node.

Chapter 6

Security and QoS Optimization

6.1 Introduction

There is a need for a mechanism to dynamically manage security and QoS such that minimum user requirements are met. Although a user may specify minimum security and/or QoS requirements, the system should aim to provide the maximum security and/or QoS possible. Malicious attacks are unpredictable and unknown. Although the user may have specified a minimum requirement, the unpredictability of an attack in terms of its time, point of attack and maliciousness suggests that the maximum security possible should be implemented in the network. This is particularly needed in a mobile ad hoc environment where there are no central or other significant points that can be monitored and the medium is open. A QoS that is more than the minimum specified is always desirable from a user perspective. In this dissertation, we propose an on-demand security and QoS optimization algorithm in a mobile ad hoc network, which can automatically adapt network security level along with changes in network topology, traffic conditions, and link QoS requirements - such as Guaranteed, controlled load, best effort, etc. to keep the security and QoS within the minimum requirements whilst aiming to providing more than the minimum security and QoS. In order to achieve this objective, we proposed two basic frameworks that are described in previous chapters: a policy based plug-in security

framework and multi-layer QoS guided routing. The plug-in security framework provides a dynamic security policy management system and the multi-layer QoS guided routing mechanism is an adaptable QoS routing mechanism for ad hoc networks to ensure QoS even as network resources change.

Based on the above two fundamental frameworks, the proposed network security and QoS optimization algorithm uses proportional integral derivative (PID) feedback control to constantly monitor and adjust the network security policy to ensure that the network satisfies all existing QoS requirements while making the network the most secure possible. When network topology changes or traffic loads become heavier, causing existing QoS links to be broken, the algorithm will selectively remove some security policies to reduce overhead, until the QoS requirements can be satisfied. If a link in the path breaks, the multi-layer QoS guided routing mechanism is activated to realize a path with the desired QoS. Hence in the proposed approach two steps are taken to ensure that desired QoS and security are maintained:

- Step 1: If the QoS is below the user specified level and the security level is above the minimum level, the security level is decreased to reduce the associated overheads.
- Step 2: If there is break in the path, the multi-layer QoS guided routing mechanism is activated to obtain a path with the desired QoS.

Alternatively if more available resources are available due to reduced traffic, the security level can be increased through the plug-in security framework. The proposed approach is equally applicable to a system where the priority is security. Here the QoS can be varied such that the required security is maintained. This approach is also

appropriate to a system where both security and QoS are of importance based on some weightage mechanism.

6.2 Feedback Control Theory

We use proportional integral derivative (PID) control theory to achieve security and QoS optimization.

A typical feedback control system looks like figure 6-1:

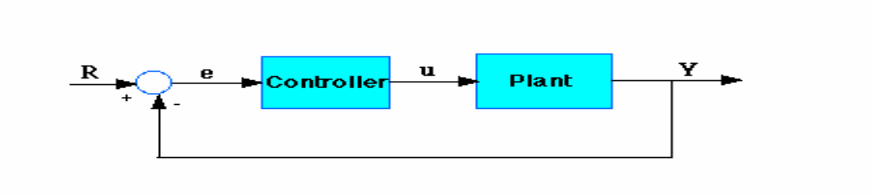


Figure 6-1 Feedback Control System

Where

Plant: A system to be controlled

Controller: Provides the excitation for the plant; Designed to control the overall system behavior

The transfer function of the PID controller looks like the follows [38]:

$$K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s}$$

Where

Kp = Proportional gain

KI = Integral gain

Kd = Derivative gain

The variable (e) represents the tracking error, the difference between the desired input value (R) and the actual output (Y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The signal (u) just past the controller is now equal to the proportional gain (Kp)

times the magnitude of the error plus the integral gain (Ki) times the integral of the error plus the derivative gain (Kd) times the derivative of the error.

$$u = K_p e + K_I \int e dt + K_D \frac{de}{dt}$$

This signal (u) will be sent to the plant, and the new output (Y) will be obtained. This new output (Y) will be sent back to the sensor again to find the new error signal (e). The controller takes this new error signal and computes its derivative and it's integral again. This is an iterative process.

A proportional controller (Kp) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (Ki) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (Kd) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Effects of each of controllers Kp, Kd, and Ki on a closed-loop system are summarized in the table shown below.

CL RESPONSE	RIST TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	Small Change

Table 6-1 Proportional, integral and derivative controller

Note that these correlations may not be exactly accurate, because Kp, Ki, and Kd are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when you

are determining the values for Ki, Kp and Kd.

For example if a modeling equation of this system is

$$Mx''+bx'+kx=F$$

Taking the Laplace transform of the modeling equation

$$Ms^2X(s)+bsX(s)+kX(s)=F(s)$$

The transfer function between the displacement X(s) and the input F(s) then becomes

$$\frac{X(s)}{F(s)}=\frac{1}{Ms^2+bx+k}$$

6.2.1 Proportional Control

From the table shown above, we see that the proportional controller (Kp) reduces the rise time, increases the overshoot, and reduces the steady-state error. The closed-loop transfer function of the above system with a proportional controller is:

$$\frac{X(s)}{F(s)}=\frac{K_p}{s^2+10s+(20+K_p)}$$

Figure 6-2 shows that the derivative controller reduced both the overshoot and the settling time, and had small effect on the rise time and the steady-state error.

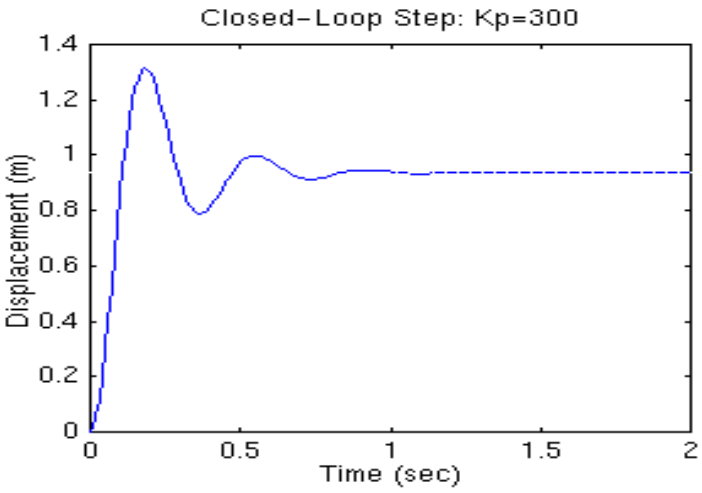


Figure 6-2 Derivative Controller

6.2.2 Proportional-Integral Control

From the table, we see that an integral controller (Ki) decreases the rise time, increases both the overshoot and the settling time, and eliminates the steady-state error. For the given system, the closed-loop transfer function with a PI control is:

$$\frac{X(s)}{F(s)} = \frac{K_P s + K_I}{s^3 + 10s^2 + (20 + K_P)s + K_I}$$

We have reduced the proportional gain (Kp) because the integral controller also reduces the rise time and increases the overshoot as the proportional controller does (double effect). Figure 6-3 shows that the integral controller eliminated the steady-state error.

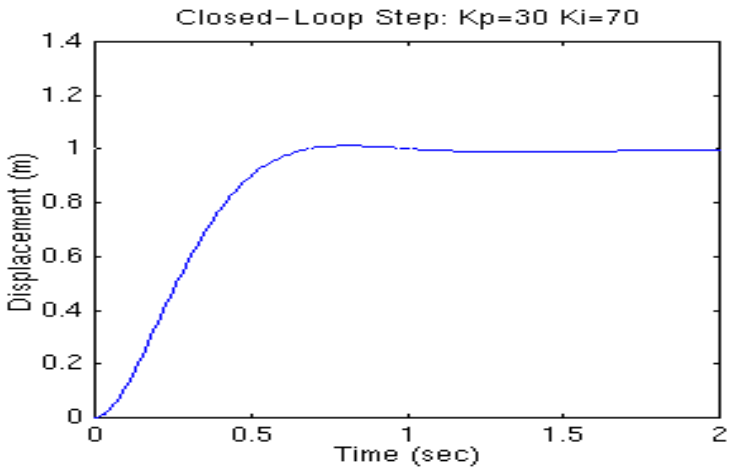


Figure 6-3 Integral Controller

6.2.3 Proportional-Integral-Derivative Control

The closed-loop transfer function of the given system with a PID controller is:

$$\frac{X(s)}{F(s)} = \frac{K_D s^2 + K_P s + K_I}{s^3 + (10 + K_D)s^2 + (20 + K_P)s + K_I}$$

Figure 6-4 shows the system with a PID controller has no overshoot, fast rise time, and no steady-state error.

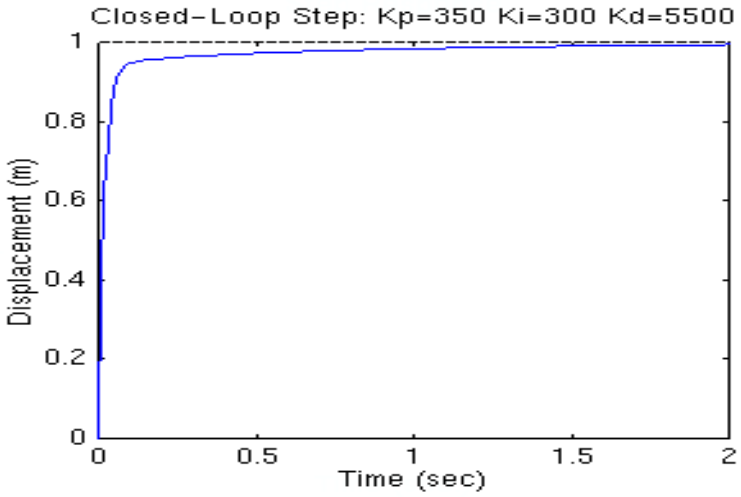


Figure 6-4 PID Controller

6.3 Security and QoS Feedback Control Loop

We use PID feedback control loop to manage network security and QoS self-optimization.

Figure 6-5 demonstrates how optimization process works. Each application has QoS requirements as input sent into the network: Guaranteed, Controlled load or Best effort. In each node of the network, the QoS plant is a module of the routing protocol to handle the QoS request. Security policies are considered as another input to the network; all the security policies are implemented by the security plant, which is a module of the routing protocol to handle all security requests. The PID controller module takes the network resource usage metrics (path latency, path throughput and path stability) as system output feedback to calculate the adjustments which will be fed into the QoS plant and security plant. This PIC control loop will constantly keep the network in the optimized state – maximize network resource usage to satisfy every QoS requests and make the network as secure as possible.

Network security is controlled by a policy based security management. This means the network security level can be adapted by the security plant module adding or removing security policies at runtime.

The PID Controller collects all actual paths' metrics from the entire network, and calculates network resource availability. If network resources are sufficient for more security policies, the PID Controller will choose more un-implemented security policies and apply to the network. Eventually, the algorithm will keep all existing paths satisfying the QoS requirements, and make the network as secure as possible.

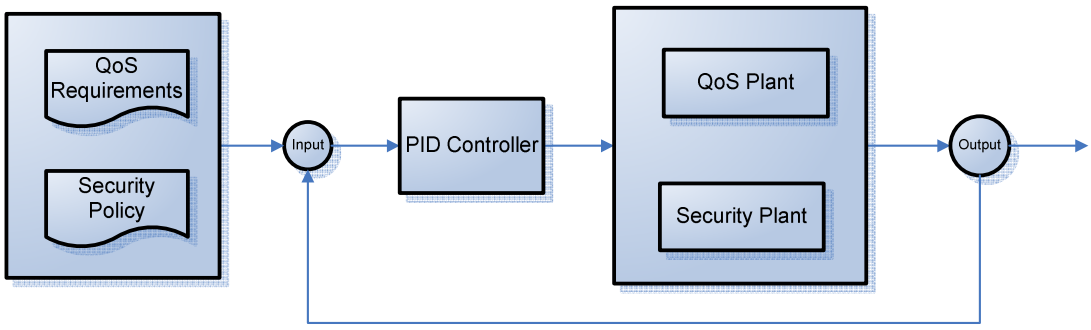


Figure 6-5 QoS and Security PID Feedback Control Loop

6.4 Measure Network Resource Availability

We use the application layer and network layer QoS metric parameter mapping to determine network resource availability.

Application layer	Metric Mapping	Network layer	Metric Mapping	MAC layer
Guaranteed	Path latency	Buffer & hop count	Greedy algorithm	SINR
Controlled load	Path throughput	Buffer & hop count	Greedy algorithm	SINR
Best Effort	Path stability	Stability & hop count	Greedy algorithm	SINR

Table 6-2 QoS metric parameter mapping

For guaranteed service, application-network layer metric mapping translates the QoS requirements to the maximum path latency. If actual path latency is less than guaranteed service target path latency, this path has sufficient resources to implement additional security policies. The target path latency can be calculated by the PID function:

$$\frac{p.latency_{target}(S)}{p.latency_{required}(S)} = \frac{K_D S^2 + K_P S + K_I}{S^3 + (10 + K_D)S^2 + (20 + K_P)S + K_I}$$

Where

$p.latency_{target}$ = target path latency at time S

$p.latency_{required}$ = required path latency at time S

Kp = Proportional gain of path latency

KI = Integral gain of path latency

Kd = Derivative gain of path latency

For controlled load service, the application-network layer metric mapping translates the QoS requirements to the minimum path throughput. If actual path throughput is more than controlled load target path throughput, this path has sufficient resource to implement additional security policies. The target path throughput can be calculated by the PID function:

$$\frac{p.throughput_{target}(S)}{p.throughput_{required}(S)} = \frac{K_D S^2 + K_P S + K_I}{S^3 + (10 + K_D)S^2 + (20 + K_P)S + K_I}$$

Where

$p.throughput_{target}$ = target path throughput

$p.throughput_{required}$ = required path throughput

Kp = Proportional gain of path throughput

KI = Integral gain of path throughput

Kd = Derivative gain of path throughput

For best effort service, the application-network layer metric mapping selects between the most stable path in the high mobility case and shortest path in the low mobility path case. There are no particular resource requirements in this case; all available security policies can be implemented.

6.5 Security Plug-in Architecture

An expansible security framework is the key to provide flexible security in an ad hoc network to achieve security and QoS optimization. We propose a policy based plug-in architecture to provide dynamic security policy management at runtime.

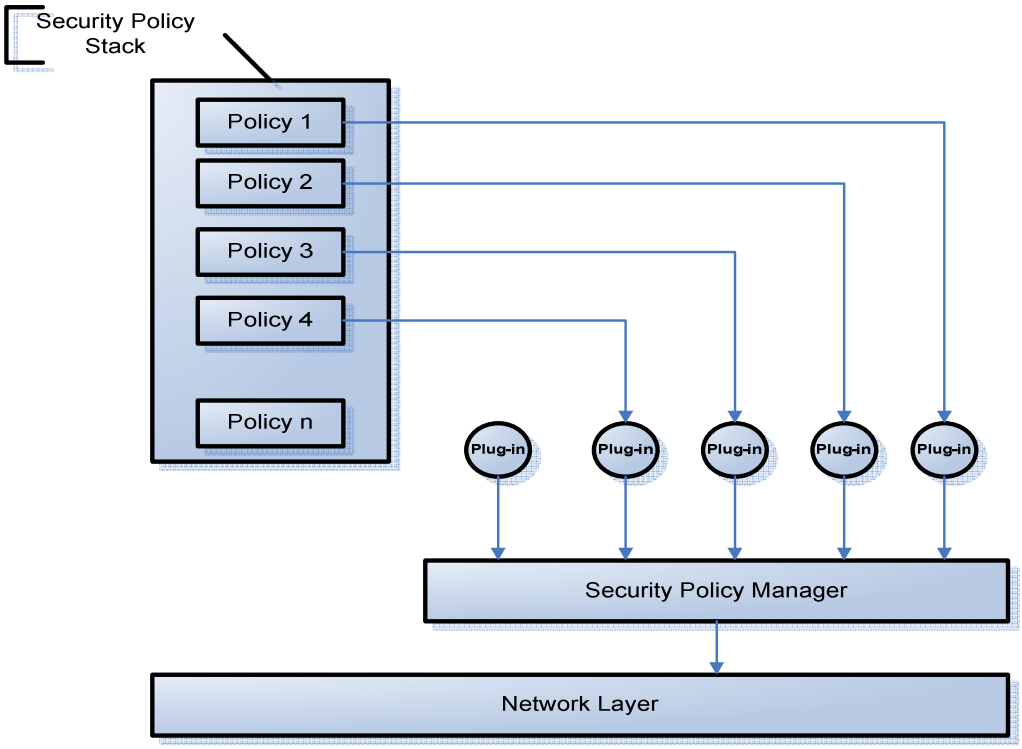


Figure 6-6 Network Security Policy Plug-in Architecture

Figure 6-6 shows the proposed security policy plug-in architecture. The security policy manager keeps monitoring the network layer. If there are more network resources available, the security policy manager will get the next available policy from the security policy stack, and activate it into the network as a plug module. If the network suffers from a lack of resources, the security policy manager will remove the least priority policy

from the network and add it back into the available security policy stack.

6.6 Optimization Algorithm

Using the path monitoring and PID feedback control loop mechanism, each communication path can determine if there are extra resources available to support more security policies until the system reaches resource utilization target. If every path in the policy domain agrees the current resource is sufficient, the domain policy manager will choose the next security policy in the available policy stack, and deploy it to every node in the domain.

6.6.1 Greedy algorithm

We use a greedy algorithm for deploying security policies to reach network resource utilization target. As long as the network does not reach its resource utilization target, the policy manager will continue deploying new security policies into the network. Figure 6-7 shows the greedy algorithm process flow.

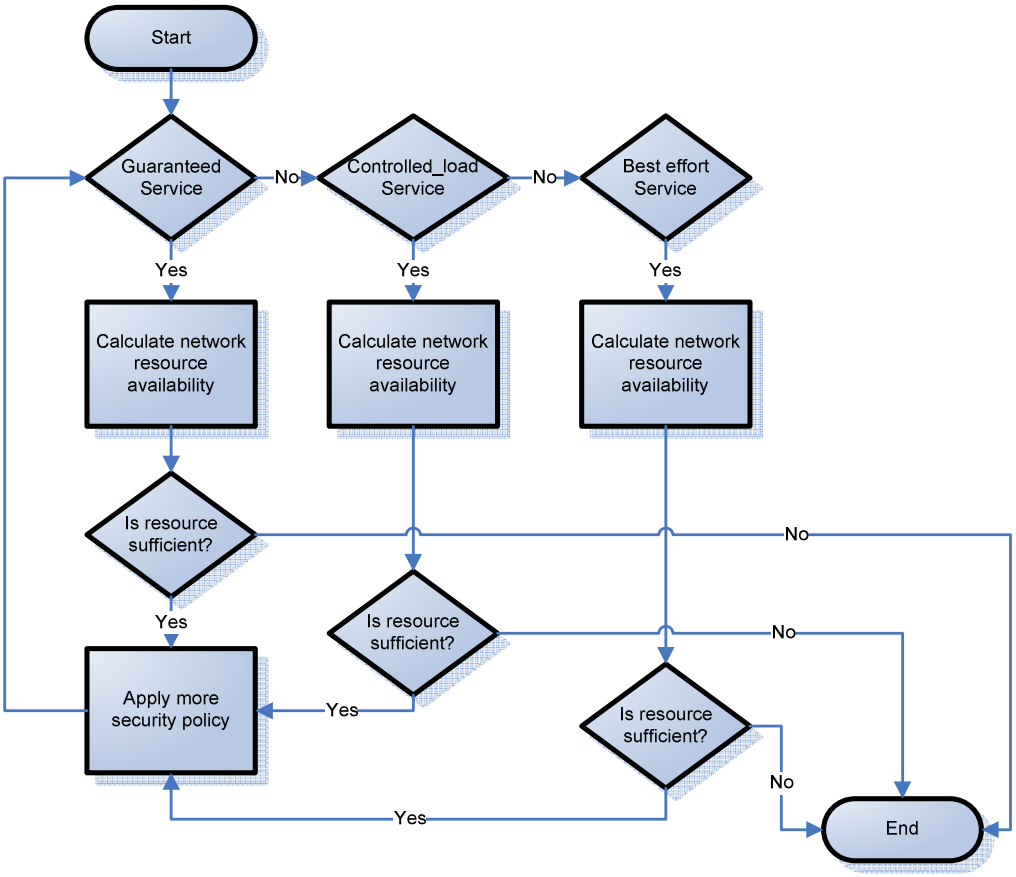


Figure 6-7 Greedy Algorithm

In real world scenario, it's impossible to keep ad hoc network at target resource utilization due to various reasons, especially mobility. Therefore, we introduce the acceptable resource utilization, where:

$$Utilization_{acceptable} = Utilization_{target} \times \delta$$

Where:

δ is the mobility factor ($0 < \delta < 1$).

Figure 6-8 shows relationship between acceptable utilization and target utilization.

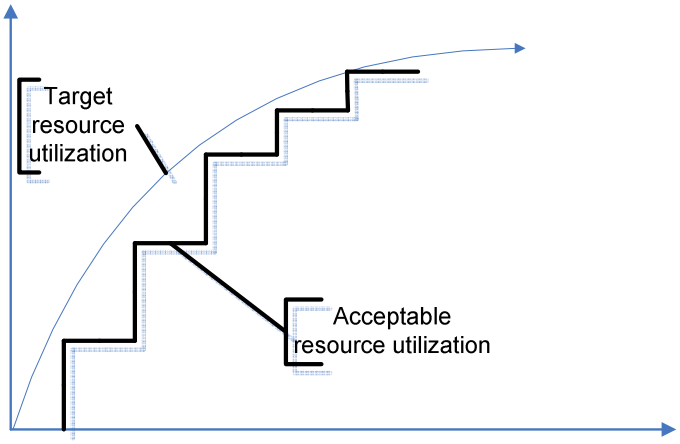


Figure 6-8 Acceptable Utilization and Target Utilization

NeedMorePolicy() routing shown in Figure 6-9 verifies if the actual resource utilization reaches acceptable utilization. It returns TRUE if actual resource utilization is below acceptable utilization, otherwise it returns FALSE. As long as the NeedMorePolicy() routing returns TRUE, the security policy manager will keep deploying the next security policy from the available stack, until the resource usage reaches the target level, at which NeedMorePolicy() returns FALSE. After that, the PID controller will take over to calculate the next resource target utilization.

```
bool NeedMorePolicy()
{
        for each PATH in DOMAIN
        {
                switch (PATH.QoS)
                {
                        case GUARANTEED:
                                if (PATH.actualLatency < PATH.targetLatency * δ)
                                {
                                        // the resource is sufficient for current path
                                }
                                else
                                {
                                        return false;
                                }
                                break;
                        case CONTROLLED_LOAD:
                                if (PATH.actualThrouthput > PATH.targetThroughput * δ)
                                {
                                        // the resource is sufficient for current path
                                }
                                else
                                {
                                        return false;
                                }
                                break;
                        case BEST_EFFORT:
                                {
                                        // no particular resource requirements
                                }
                                break;
                }
        }
        return true;
}
```

Figure 6-9 Need More Policy Algorithm

6.7 Policy Deployment Post Validation

The path monitoring and feedback control loop mechanism also need to verify that there is no existing path suffering from a lack of resources due to the new security policy deployment. If there is any path that is not able to satisfy the original QoS requirement, which means the previously deployed security policy is causing the network to suffer from resource starvation, the domain policy manager needs to remove the previously deployed security policy and log all the suffering paths. The greedy algorithm will not be called until at least one of the suffering paths change state (e.g., finish communication, change QoS requirements, etc.).

The process flow and algorithm are showing in Figure 6-10 and Figure 6-11:

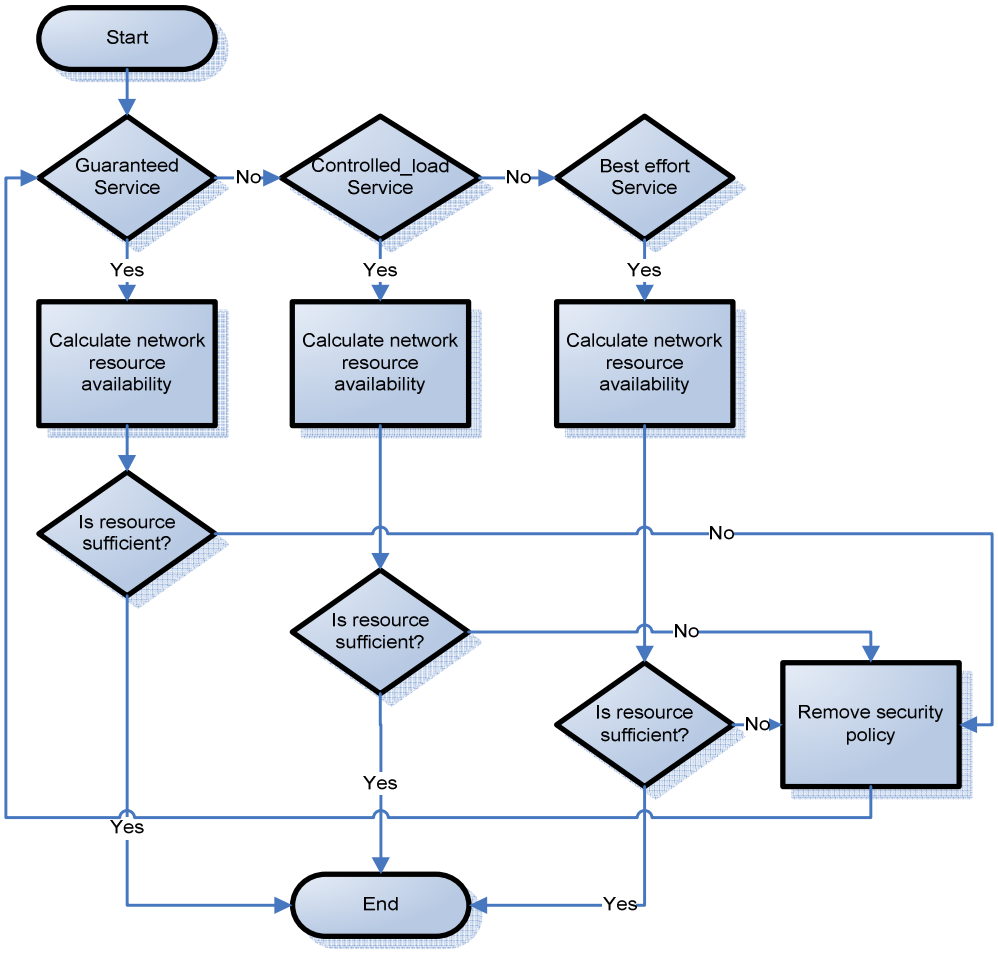


Figure 6-10 Policy Deployment Post Validation Process Flow

```
bool NeedRemovePolicy()
{
	for each PATH in DOMAIN
	{
		switch (PATH.QoS)
		{
			case GUARANTEED:
				if (PATH.actualLatency < PATH.targetLatency)
				{
					// the resource is sufficient for current path
				}
				else
				{
					return true;
				}
				break;
			case CONTROLLED_LOAD:
				if (PATH.actualThrouthput > PATH.targetThroughput)
				{
					// the resource is sufficient for current path
				}
				else
				{
					return true;
				}
				break;
			case BEST_EFFORT:
				{
					// no particular resource requirements
				}
				break;
		}
	}
	return false;
}
```

Figure 6-11 Policy Deployment Post Validation Algorithm

6.8 Performance Analysis

The performance of the proposed PID controlled security and QoS optimization algorithm is studied with simulations. Ad Hoc On-Demand Distance Vector protocol (AODV) is one of the most commonly used protocols in ad hoc. In this study, we are using AODV as our base model, to compare with QoS AODV, static policy based secure AODV (PS-AODV) and the proposed PID optimized AODV (PID-AODV).

6.7.1 Simulation Model

The QoS routing protocol has been implemented with ns2 [36]. The implementation is based on AODV module contributed by the MONARCH group from CMU, and the QoS routing functions are added. In additional to building QoS routes, the protocol also builds a best-effort route when it learns such a route. The best-effort route is used when a QoS route is not available. The Evolutionary-TDMA scheduling protocol (E-TDMA) [37] developed by the same authors is used at the MAC layer. It is distributed protocol which dynamically generates and updates TDMA transmission schedules among the nodes. Transmission rate is 1 Mbps. There are 40 slots in a frame, and a slot carries 32 bytes of information. A packet needs to be transmitted in multiple slots if it cannot fit in one slot. Limited contention is used for nodes to make their time slot reservations, hence E-TDMA a mainly limited by nodal density rather than network size. Considering the overhead of making reservation, an information slot is equivalent to 18 kbps. Details of E-TDMA can be found in [37]. In the simulations, Route_setup_time = 1000 ms and Route_life_time = 200 ms.

The implementations of AODV, PS-AODV, QoS-AODV and the proposed PID-AODV in our simulation environment closely match their specifications. The routing protocol model detects all data packets transmitted or forwarded, and responds by invoking routing activities as appropriate. The RREQ packets are treated as broadcast packets in the MAC. RREP and data packets are all unicast packets with a specified neighbor as the MAC destination. RERR packets are broadcast in both AODV and QoS-AODV. Both protocols detect link breaks using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop.

All protocols maintain a send buffer of 64 packets. It contains all data packets waiting for a route. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 seconds. All packets sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has maximum size of 50 packets and is maintained as a priority queue with three priorities each served in FIFO order. Routing packets get higher priority than security packets, and security packets get higher priority than data packets.

Our proposed PID-AODV routing protocol are implemented based on the existing QoS-AODV protocol in ns2. The PID control module is created by Object-oriented Tool Control Language (OTcl) as a plug-in implemented above the network layer. It collects path latency and throughput as network output parameters and sends security policy requests back to the network layer to perform optimization.

6.7.2 Traffic and Mobility models

We use traffic and mobility models similar to those previously reported using the same simulator. Traffic sources are CBR (continuous bit-rate). The source-destination pairs are spread randomly over the network. Only 512 byte data packets are used. The number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network.

The mobility model uses the random waypoint model in a rectangular field. We use 1500m x 300m field with 50 nodes. Each node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0-20 m/sec). Simulation period is 900 seconds. Each data point represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios. Identical mobility and traffic scenarios are used across protocols.

6.7.3 Security Policies

We use three security policies in our simulation: domain join authentication, read access authorization, write access authorization. Each security policy has been assigned a priority level. Depending on the network resource utilization ratio, the algorithm will add or remove security policies based on the priority level to maintain the QoS.

Security policy	Priority
Domain join authentication	High
Write access authorization	Median
Read access authorization	Low

Table 6-3 Security policy priority

6.7.4 Simulation Results

The proposed PID optimized AODV routing protocol (PID-AODV) is compared with the AODV, QoS AODV and static policy based secure AODV protocols.

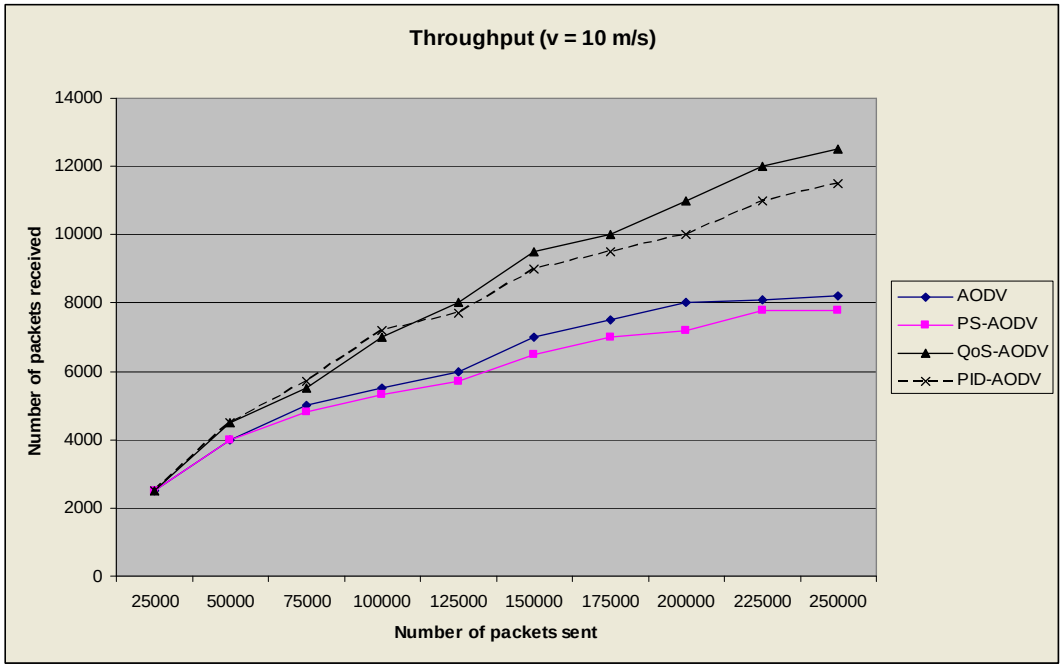


Figure 6-12 Throughput for v = 10 m/s

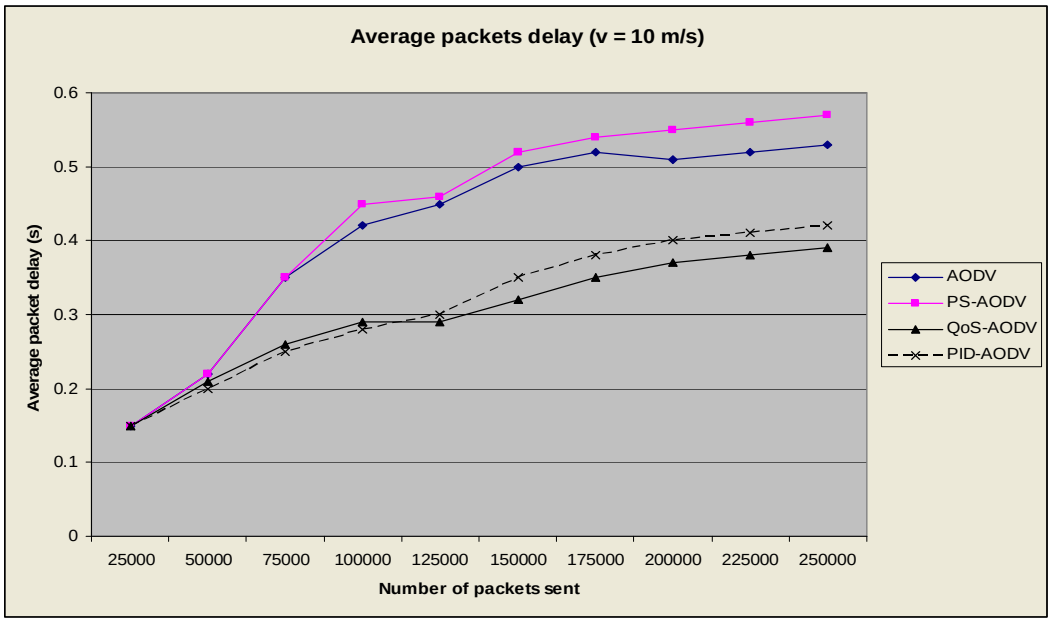


Figure 6-13 Average packets delay for v = 10 m/s

Figures 6-12 and 6-13 show the packet throughput and the average packet delay under different traffic loads in mobility v = 10 m/s. Under light traffic, packet throughput and packet delay are very close for all three protocols, because they often use the same routes.

The advantage of QoS routing protocols become apparent when traffic gets heavy. With the AODV protocol, a node has one active route to a destination and uses it for all the packets to the destination. As the network traffic becomes heavy, this route becomes heavily loaded, causing packets to be delayed and dropped. The average packet delay increases significantly under heavy traffic. On the other hand, the QoS routing protocols try to find and use routes satisfying bandwidth constraints for different flows, even between the same pair of source and destination. Two QoS routes may share the same path, but the protocol will ensure enough bandwidths are reserved on this path to accommodate both flows. The traffic load is more balanced this way. The average packet

delay increases with offered load slowly with the QoS routing protocols.

Under light traffic, PID-AODV does not have much advantage in terms of performance compared with AODV and PS-AODV. As the network traffic becomes heavy, PID-AODV performs better. It provides a same level of security as PS-AODV, but has throughput and packets delay that is very close to QoS AODV. It therefore provides the security of a secure protocol and the QoS of a QoS protocol.

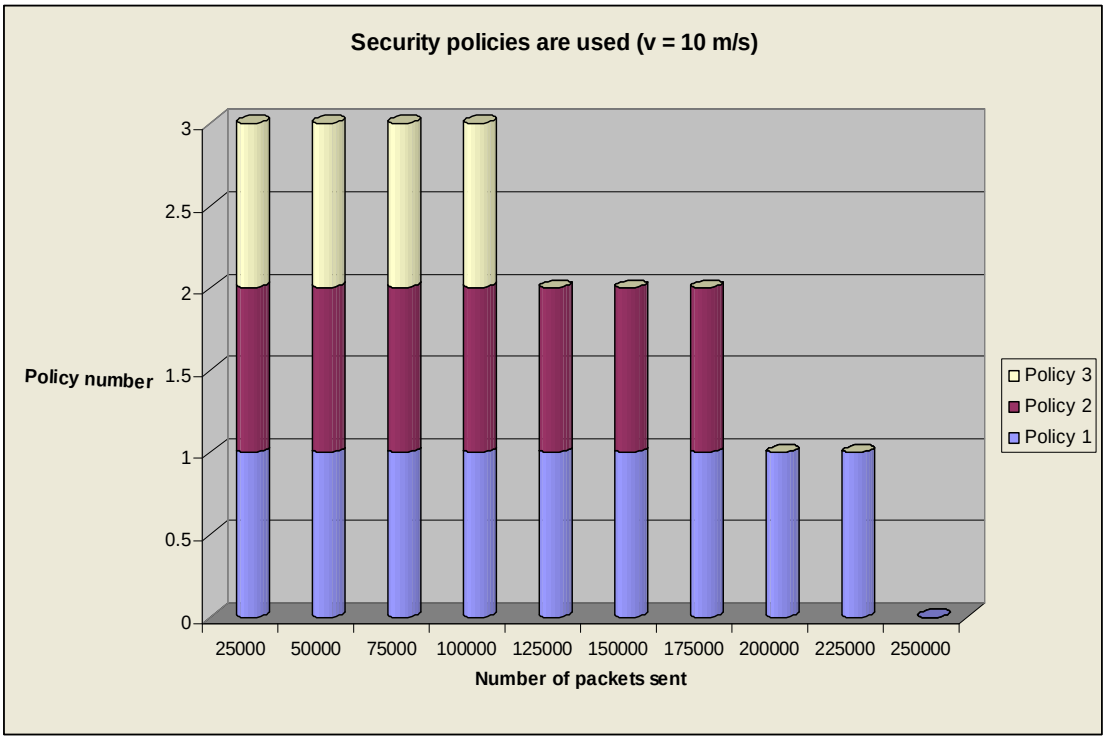


Figure 6-14 Security policies are used for v = 10 m/s

Figure 6-14 shows the number of security policies that have been used in PID-AODV protocol at mobility v = 10 m/s. Initially, three security policies have been used under light traffic, because there is enough bandwidth resource in the network. When the traffic becomes heavier, the PID controller starts reducing the number of the security

policies to keep up the same performance as QoS AODV. Eventually, when the network traffic becomes too heavy, there is no extra bandwidth to handle any security feature, the security policy number drops to 0.

6.7.5 Conclusion

The simulation results indicate the proposed PID optimized security and QoS algorithm can produce similar performance as non-secure QoS routing protocol under various traffic loads.

The level of security can be adaptable due to different traffic loads. The best case scenario is under light traffic where it can provide the same security level as any other secure protocols, but the same performance as non-secure QoS protocols; the worst case scenario is under extreme heavy traffic where it provides similar performance as QoS protocols, but with no security feature at all.

Under normal traffic condition - medium traffic load, the proposed protocol can provide more secure networks without compromising the QoS performance.

Chapter 7

Network Security Measurement

7.1 Introduction

There are many routing protocols around including secure routing protocols. However, a question that arises is, how secure are these protocols? In other words, can we define a security metric? This is difficult, if not impossible. However, as proposed by [47] [48] we can come up with a relative comparison of the security of two protocols. In [47] [48], the vulnerabilities in the system are identified and summed up to measure the security of the system. However, this approach is simplistic and does not reflect a true measure of security for a number of reasons.

- A system may have many vulnerabilities, but it may still be secure because the goal of the attack is not realizable in this system. For example, DSDV routing can be very secure from routing table overflow attack but vulnerable from routing cache poisoning attack.
- A system may have few vulnerabilities, but if there are multiple ways to exploit these vulnerabilities, the system is relatively insecure.
- A system may have vulnerabilities, which if exploited on an individual basis pose little threat. However, if these vulnerabilities are exploited one after the other as a group, may have serious consequences. For example, a vulnerability

in the system results in an attacker obtaining the user ID. This by itself is not a major threat. However, if the vulnerabilities can be exploited for the attacker to gain the user ID followed by the password, this is a very serious attack.

In this chapter we propose a new approach to measuring security based on the parameters identified above – namely, any measure of security should be based on:

- the vulnerabilities in the system, as described in [47] and [48]
- The feasibility of realizing the attack goal
- The number of different ways to exploit different or the same vulnerabilities to achieve the goal
- The exposure of the system resulting from the exploitation of multiple vulnerabilities

In the proposed approach the vulnerabilities in the system are measured using the number of potential vulnerable resources in the system, the feasibility of realizing the attack goal is represented by a threat agent, and the number of different ways to achieving the attack goal is modeled using attack paths. We also defined the concept of an attack tree to identify the overall system exposure.

7.2 Fundamentals of Security and Attack

A successful system attack usually is caused by the existing of both internal system flaws and external threats. This section briefly discussed the basic concept of the fault path; fault path is the steps that external threat attacks internal system flaw to achieve system damage.

7.2.1 Security and Dependability

Security is a property of a *system* or *service*. A *system* is an entity that has internal structure and interacts with other systems. We are interested in systems that are engineered; i.e. are developed and then operated to achieve some useful purpose. The purpose of the system is implemented as the *service* the system, acting as a provider, delivers to another system, the user system.

The user system is dependent on the provider system for the service. The delivered service usually will have many properties, depending on its type. Among these, the user system will be concerned about the *dependability* of the provider system, or, equivalently, of the provided service [49]:

1. The ability to deliver a service that can justifiably be trusted. (calls for a justification of trust)
2. The ability to avoid service failures that are more frequent and more severe than is acceptable (implies criteria for deciding whether a service is dependable)

The second definition indicates a measurement approach to dependability, based on the likelihood and severity of service failures.

A particular service can fail in a variety of ways, resulting in dependability being a composite property, covering the following more specific properties (*more* of the property is indicative of *fewer* or *absence* of the corresponding failures):

Dependability Property of a System	Associated Types of Service Failure
Availability	failures implied by the service being *incorrect*
Reliability	interruption or outage in correct service over a time interval
Safety	failures that cause catastrophic harm to users or the environment
Integrity	improper/unauthorized system alterations
Maintainability	service failures resulting from a system being difficult to successfully maintain during use

Table 7-1 Dependability Property of a System

Like dependability, *security* is a composite property of a system or service, with different sub-properties being associated with different types of service failure:

Security Property of a System	Associated Types of Service Failure
Confidentiality	unauthorized disclosure of information
Integrity	improper/unauthorized system alterations
Availability	types of failure implied by the term *correct*
Authenticity	A user not identified correctly – not who they claim to be
Non-reputability	A neutral third party is unable to decide if a particular transaction or event did or did not occur

Table 7-2 Security Property of a System

Definitions of security in the literature vary according to the types of failure that are of concern. The following are representative:

1. Preservation of confidentiality, integrity and availability of information; in addition, other properties, such as authenticity, accountability, non-repudiation, and reliability can also be involved.

2. Work that involves ensuring the confidentiality, integrity, and availability of systems, networks, and data through the planning, analysis, development, implementation, maintenance, and enhancement of information systems security programs, policies, procedures, and tools.

Dependability and security overlap in the sense that some types of failure fall under both properties. For convenience, security will be discussed as a single property in the following. It is understood that, for a particular system or service, dependability and security will be defined as some selection from the sub-properties, depending on the concerns of the user system.

The definition of dependability and security as the ability to avoid failures raises the question of how a system or service can be measured with regard to such ability. Before addressing this question, we need to define a model of how a service failure is caused.

8.2.2 Faults and Errors

A service failure implies that the provider system's external states (i.e. those states observable by the user at the provider's service interface) deviate from the external states associated with the provision of a correct service. This deviation is called an *error*. The adjudged or hypothesized cause of an error is called a *fault*. Faults may be located within the provider system and/or in its environment.

Security vulnerability is a type of internal fault that enables an external fault to cause harm. An external fault may be the result of malicious actions of a threat agent. A system may have a property that is believed to remove or mitigate a fault or set of faults.

Figure 7-1 illustrates a fault path linking three domains: a system/service environment, a system of interest that provides a service; and a user system in which service failures may cause damage.

Faults can arise in any aspect of a system. Avizienis [49] provide an extensive typology of faults, along several dimensions. It is often a question of judgment as to the root cause of a failure, i.e. where a chain of dependability and security threats begins. For example, the presence of a fault in a software component may be due to a failure in the software development process (viewed as a service provided by a project socio-technical system).

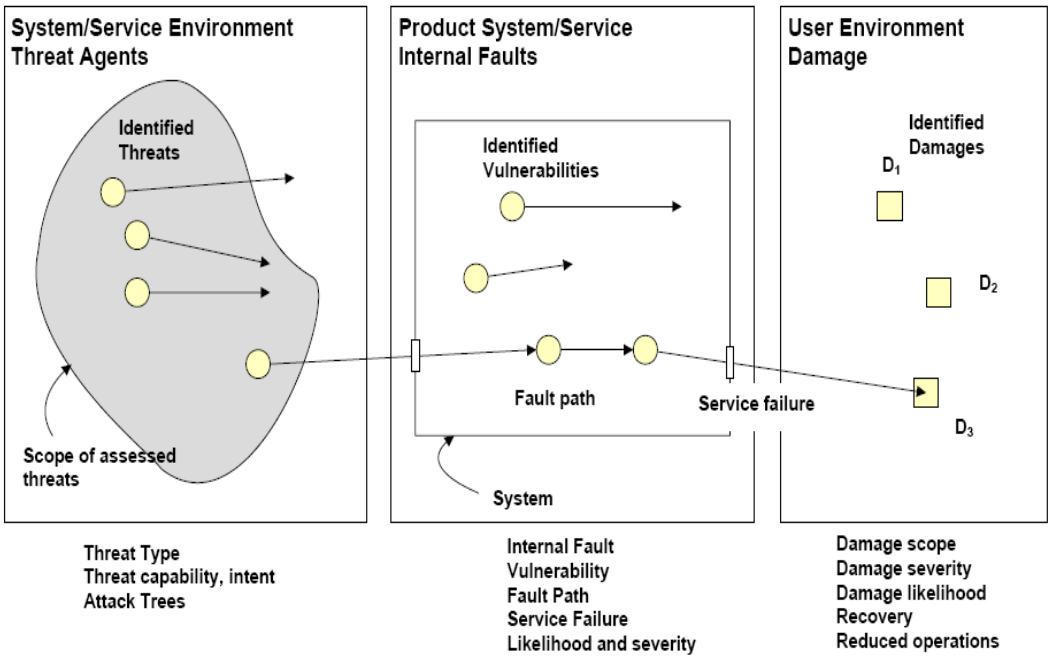


Figure 7-1 Fault Path

7.2.3 Threats

It follows from the definitions that security is a property of a system (and provided service) *in relation to* a threat environment. A given system may be acceptably

secure in one environment, but not in another; or it may be acceptably secure today but not tomorrow.

Many types of fault of concern to security engineering are similar to safety faults: i.e. events in the natural environment, accidental, non-malicious actions during development etc. However, security has an additional type of fault arising from the presence of malicious threat agents in the operational and development environment. Such agents can learn and adapt, resulting in evolving external faults.

Attack Trees are used to map the objectives of a threat agent onto vulnerabilities of the system. Alternative attack sequences represent the possible ways the agent might achieve his/her goal. Development and operational policies can be adjusted to prioritize defensive actions.

Measurement can support the decision making involved, for example in the estimation of the cost to a threat agent of different attack sequences. Under certain assumptions, an increase in attack cost would imply a lowering of the likelihood of the attack sequence occurring and an increase in security with regard to the associated service failure.

7.2.4 Security Principles and Policies

The security field is a large one – information security is perhaps the most general term (to which might be added control system security). The fields of computer security, network security and software security are more specialized areas of professional engineering practice. Each has more specialized areas of expertise. System security engineering addresses the concerns from the viewpoint of software-intensive systems,

compatible with systems engineering as defined by ISO/IEC 15288 and related standards. The overlap with system safety engineering has been addressed in recent years [50].

The long history of computer security has established several principles that are used to guide the architectural design and operation of secure computer-based systems. They can be viewed as being expressed through design policies and requirements and include [51]:

1. Accountability;
2. Least privilege;
3. Minimize the variety, size and complexity of trusted components;
4. Secure default configurations;
5. Defense in depth.

Such principles guide strategic design choices that reduce the likelihood of common types of service failure. Security principles are implemented using a selection of security mechanisms, for example [51]:

1. Defining and implementing domains, i.e. areas of stored data and applications with restricted access;
2. Linking users with domains;
3. Authorizing operations;
4. Auditing operations;
5. Cryptography.

Security mechanisms are implemented by a range of security components (i.e. components whose primary functions are security-related) forming the security architecture of the system, and operations policies. Systems and software security

engineering specialties are responsible for specifying, designing and implementing these systems, and for supporting general systems and software engineering functions in realizing the security properties of the total system product.

Measurements can be developed to (1) assess the degree to which an implemented and operated system meets the design intent and (2) the degree to which the design intent, as implemented, meets the needs of users.

7.3 Attack Surface

The concept of Attack Surface model introduced in [47] [48] measures computer Operating System vulnerability and attack ability. The attack surface model uses state machines to represent all potential system resources that can be used by an attacker to achieve an attack goal. These vulnerabilities are described as “dimensions”, and they are compared to provide a measure of relative security. In this approach, rather than saying “System A is secure” or “System A has a measured security number N” the attack surface model says “System A is more secure than System B with respect to a fixed set of dimensions.”

The attack surface models an attack as a three dimension model: target, carrier and communication channel. Target is the attack goal; carrier is the media by which an attacker passes the attack to the target, the examples of carriers include viruses, worms, Trojan horses, and email messages; communication channels are the means by which the attacker gains access to the targets on the system. The attack surface model uses the matching mechanism to identify system security exposure. If any system data and process can be identified as an attack target, carrier or communication channel, they are counted as security exposure. The overall count is summation of the dimensions from the attack surface metric for the system.

However, the state machine model of a threat used in the attack surface model does not precisely represent a real world threat; it simply lists all system resources that are utilized by the threat as one single level – without dependency, and give equal weight – same importance to all these resources. But a real world threat might have one or more

attack sequences, where some of the steps in the attack sequence can be more critical than others, or depends on the successful attack of multiple sub-sequences.

For example, a system A that exposes both user name and password is more vulnerable than a system B that exposes both employee salary and password, although all of the above information are been classified as sensitive data. An attacker can cause more serious damage to system A than system B by using a stolen identity to successfully login into system A. The attack surface model, however, measures the same vulnerability level for both system A and system B in this scenario.

In the next section, we propose a new state machine model of threat. Each threat will be associated with an attack tree; there might be one or more critical paths within an attack tree. Therefore, some system resource can be weighted more than others, depending on where it is located in the attack tree.

7.4 Proposed Measurement Technique

7.4.1 Vulnerability Assessment and Security Measurement

We want a measure at a lower abstraction level that allows us to refer to very specific states of a protocol [54]. There are certain protocol features that are more likely than others to be opportunities of attack, such as sending information to an unauthenticated node, etc. The counts of these "more likely to be attacked" protocol features determine a protocol's attack opportunity.

Suppose we are given a fixed set of dimensions and a fixed set of attack opportunities for each dimension. Then with respect to this fixed set of dimensions of attack opportunities, we can measure whether protocol A is "more secure" than protocol B.

We use state machines to model the vulnerability of protocol A and B. Our abstract model allows protocol A and B to be any two state machines. In practice, it is more useful and more meaningful to compare two protocols that have some close relationship.

The abstract dimensions along which we compare two protocols are derived directly from our state machine model: process and data resources and the actions that we can execute on these resources. For a given threat agent, which we define to be a sequence of action executions, we distinguish attack goal from attack path: attack goal are processes or data resources that an attacker aims to control, and attack path are all other processes and data resources that are used by the attacker to carry out the attack successfully. The attacker may use a set of attack goals (attack objective), attack paths

(sequence of steps to achieve an attack goal) and critical paths (the primary attack paths to cause system failure) to accomplish the ultimate attack goal. Control is subject to the constraints imposed by a protocol's set of access rights. In summary, our threat agent metric's four dimensions are: Attack Goal, Attack Path, Critical Path and Access Rights.

Figure 7-2 demonstrates the process flow of network security measurement metric generation and the four dimensions of the security measurement metric.

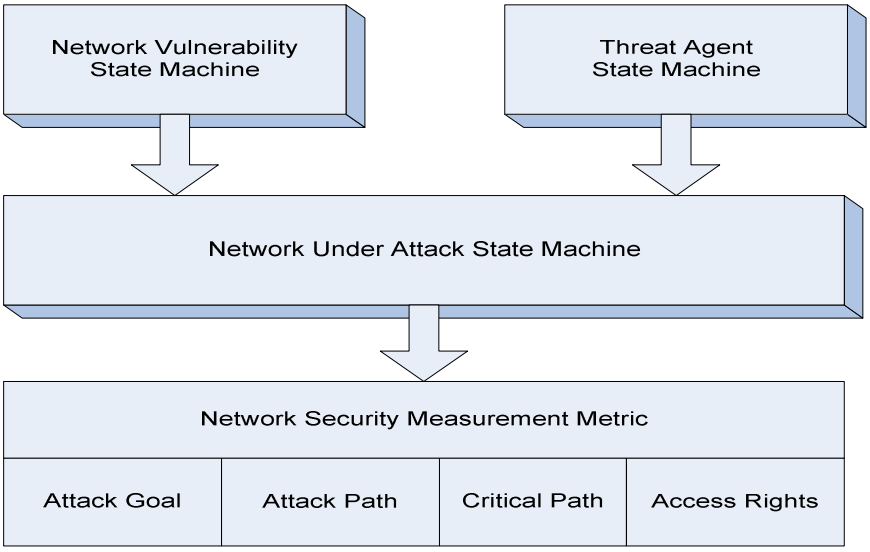


Figure 7-2 Network Security Measurement Metric

7.4.2 State Machine

We model both the protocol's vulnerability and the threat agent as state machines. A state machine has a set of states, a set of initial states, a set of actions, and a state transition relation. We model an attack as a sequence of executions of actions that ends in a state that satisfies the attacker's goal, and in which one or more of the actions executed in an attack involves vulnerability.

A state machine, M = (S, Σ, T, s, A), is a 5-tuple,

Where:

a finite set of states (S)

a finite set called the alphabet (Σ)

a transition function (T: $S \times \Sigma \rightarrow S$).

a start state ($s \in S$)

a set of accept states ($A \subseteq S$)

A state S is a mapping from typed resources to their typed values:

S: Resource(M) $\rightarrow$ Value(M)

Of interest to us are state resources that are processes and data. A state transition, (s, σ, s'), is the execution of action σ in state s resulting in state s'. A change in state means that either a new resources is added to the mapping, a resource was deleted, or a resource changes in value. We assume each state transition is atomic.

An execution of a state machine is the alternating sequence of states and action executions:

$$s_0 \times \sigma_1 \rightarrow s_1 \times \sigma_2 \rightarrow s_2 .....s_{i-1} \times \sigma_i \rightarrow s_i ...$$

An execution can be finite or infinite. If finite, it ends in a accept state A.

The behavior of a state machine, M, is the set of all its executions. We denote this set Behavior(M). A state S is reachable if either S $\in$s or there is an execution, E $\in$Behavior(M), such that S in E.

We will assume that action are specified by pre- and post-conditions. For an action, σ, if σ.pre and σ.post denote σ's pre- and post-condition specifications, we can then define the subset of the transition relation such as:

$$\sigma.T = \left\{ \left\langle s, \sigma, s' \right\rangle : S \times \Sigma \times S \mid \sigma.pre(s) \Rightarrow \sigma.post(s, s') \right\}$$

We model both the network vulnerability and the threat agent as state machines:

$$Vu\ln erability = (S_V, \Sigma_V, T_V, s_V, A_V)$$

$$ThreatAgent = (S_T, \Sigma_T, T_T, s_T, A_T)$$

We define the combination of the two state machines, Security = Vulnerability x ThreatAgent, by merging all the corresponding components:

- $S_S \subseteq 2^{\mathrm{Re}\,source_S \to \mathrm{Value}_S}$
- $s_S = s_V \cup s_T$
- $A_S = A_V \cup A_T$
- $T_S = T_V \cup T_T$

An attacker targets a network under attack to accomplish a goal:

Network-Under-Attack = (NetworkVulnerability x ThreatAgent) x Goal

Where Goal is formulated as a predicate over states in S_S.

7.4.3 Modeling Threat Agents

Factors involved in assessing the security risk posed by a particular agent have been modeled by [52] (Figure 7-3). These factors can be assessed on the basis of qualitative scales, enabling risks to be prioritized. For example, the threat capability of a group of terrorist threat agents might be assessed on the basis of [52]:

1. Group size;
2. Level of education;
3. Cultural factors;
4. Access to communications and the Internet;
5. Technical expertise;
6. History of activity;
7. Sponsoring countries;
8. Funding.

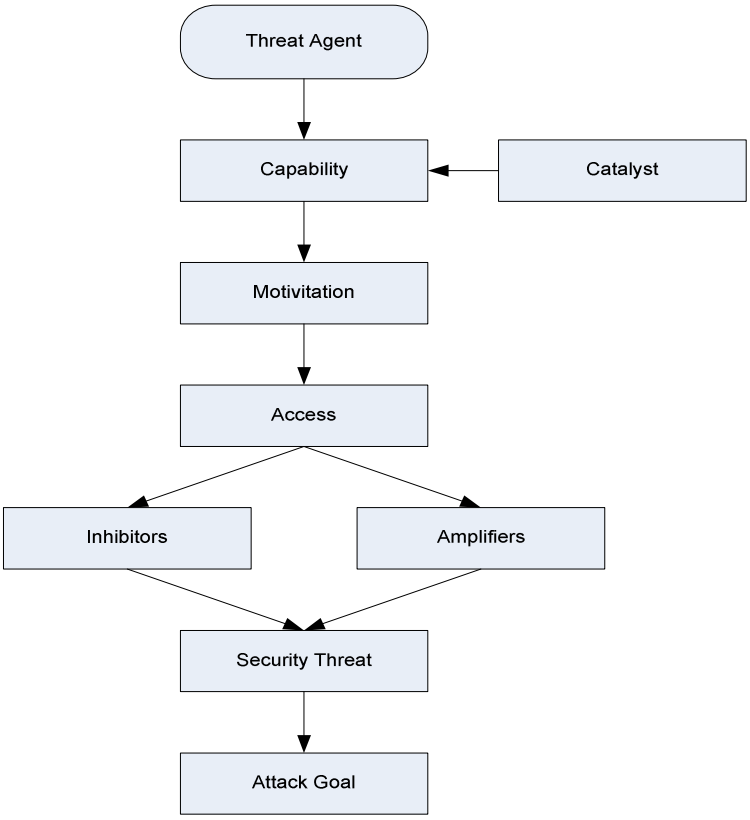


Figure 7-3 Aspects of a threat agent

Let M = (*S*, Σ, *T*, *s*, *A*) be the state machine representing the network under attack and the goal the state predicate characterizing the attacker's goal to be achieved in attacking the network.

A threat agent is a finite sequence of action executions $\sigma_1, \sigma_2, ...\sigma_i, ...\sigma_n$ such that:

$\forall 1 \le i \le n$;

$\sigma_i \in \Sigma$.

A threat agent includes actions from the action sets of the network, the threat. Since an attack is initiated by the threat, the sequence starts with an action of the threat. The sequence includes at least one action of the network to model the exploitation of

some network vulnerability by the threat in the attack. Finally, the attacker's goal should hold at the end of the attack path.

7.4.4 Modeling Attack Tree

The level of threat (potential to cause damage) of a threat agent is also influenced by their motivation and opportunities to access the system, among other factors.

An attack tree is defined as a tuple G = (V, E, f),

Where:

V is a finite set of vertices (attack goals);

E is a finite set of edges (attack steps);

f is a logic function that maps a vertices into an AND/OR tree.

An attack path is a finite sequence of vertices and edges from a leaf node to the root of an attack tree $v_1, e_1, v_2, e_2, ...v_i, e_i .....v_n$ such that:

$\forall 1 \le i \le n$;

$v_i \in V$;

$e_i \in E$.

Attack trees model the particular attack goals and the options for achieving them in relation to the attacked system. A top-level goal (Figure 7-4) is decomposed into sub-goals in an AND/OR tree. The path from a leaf node to the top-level root is an *attack path*. The set of all identified threats to a system from a particular threat agent is the agent's *threat profile*.

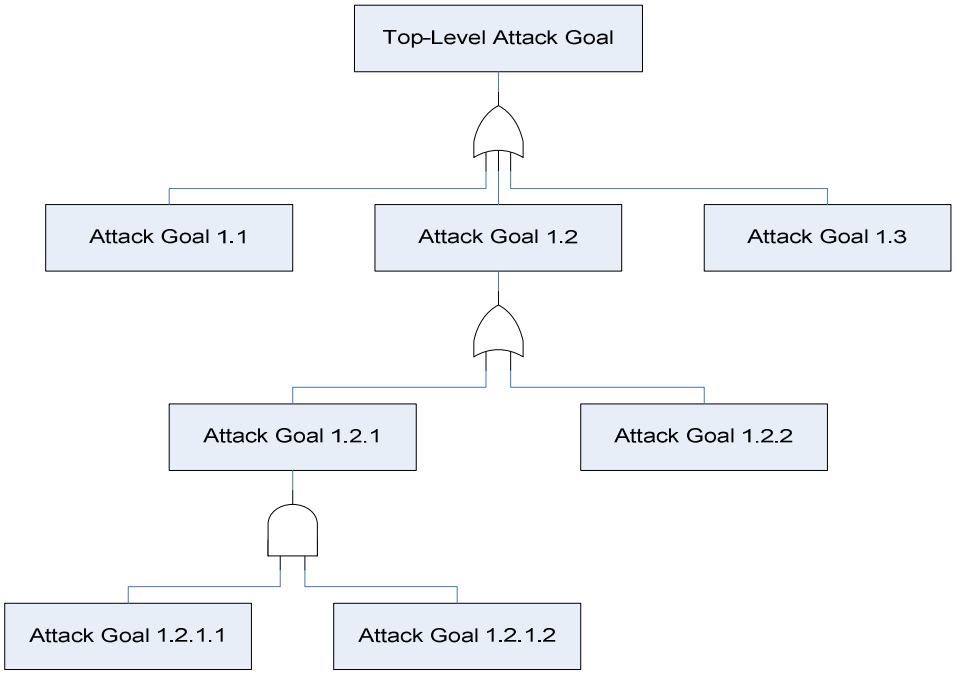


Figure 7-4 Attack tree

Attack trees may be used to integrate quantified assessments of the costs to the attacker in achieving the goal at each node. Alternatively, a probability of success may be associated with each node, based on judgments about the threat presented by the agent and the protection presented by the system. If probabilities could be assigned to nodes, the likelihood of a successful attack could be assessed from the probabilities along the complete network of potential attack paths. The security risk associated with the attack is assessed from the costs associated with the effects of the successful attack.

In addition to the probability and cost aspects, measurements can also be based on tracking identified threats and attack paths (as in a project risk register); the number of threats (top level goals) and attack paths, under selected categories, can be tracked over time. Time and costs associated with mitigation actions can be tracked.

The particular form of attack goals and sub-goal strategies will depend on the target system and assets. For example, threat effects have been classified as follows in the development of secure application software (not a complete list) [53]:

1. Spoofing;
2. Tampering;
3. Repudiation;
4. Information disclosure;
5. Denial of service;
6. Elevation of privilege.

7.5 Security Measurement Metric

We define the notion of threat agent and security measurement more formally and in terms of a different state machine model. The significant difference in our state machine model is in making the access matrix explicit and in distinguishing the system as an entity different from its principals.

We present a method of applying our metric so that others can use the notion of security measurement for any network. The method requires identifying resources that are potential goals of threat agents and identifying interesting properties of the resources to characterize their attackability. We also allow users to specify a penalty function for attacks, to help determine what attacks to use for comparing two network protocols.

7.5.1 Dimensions of a Threat Agent

- Attack Goal

  To achieve the attack goal, the attacker has one or more sub-goals on the network to attack. An attack goal is a distinguished process or data resource in network that plays a critical role in the attacker's achieving his goal.

- Attack Path

  We use the term path for any accessed process or data resource that is used as part of the means of the attack but is not signed out to be a target.

- Critical Path

  Critical path is a set of attack paths that the attacker may use to achieve the ultimate attack goal.

- Access rights

These rights are associated with each process and data resource of a state machine.

Intuitively, the more attack goals the threat agent has, the less secure the network. The more attack paths the threat agent has, the less secure the network. The smaller attack trees the threat agent has, the less secure of the network. The more generous the access rights, the less secure the network.

7.5.2 Attack Goal and Attack Path

Attack goal and attack path are resources that an attacker can use. There are two types of resources: processes and data. It's a matter of the attacker's goal that determines whether a resource is an attack goal or an attack path. In particular, an attack goal in one attack might simply be an attack path for a different threat agent, and vice versa.

Examples of process targets are message sending, routing table updating, and password changing. Example of data goals are access rights, routing tables, important files and data stored on specific nodes in network.

Part of calculating the security metric is determining the types and numbers of instances of potential process goals and data goals.

7.5.3 Critical Path

Critical path is one or multiple attack paths that an attacker can use to achieve the ultimate attack goal. Figure 7-5 uses a different color scheme to demonstrate that there

are a total of 4 different critical paths within a single attack tree. In this scenario, an attacker can use 4 different attack methods to achieve the top-level attack goal. If any one of the attack methods can be successfully executed, the attacker will successfully attack the system. Notice Attack Goal 1.2 is shared by two critical paths, which means there are certain system resources that can be used by multiple attack methods.

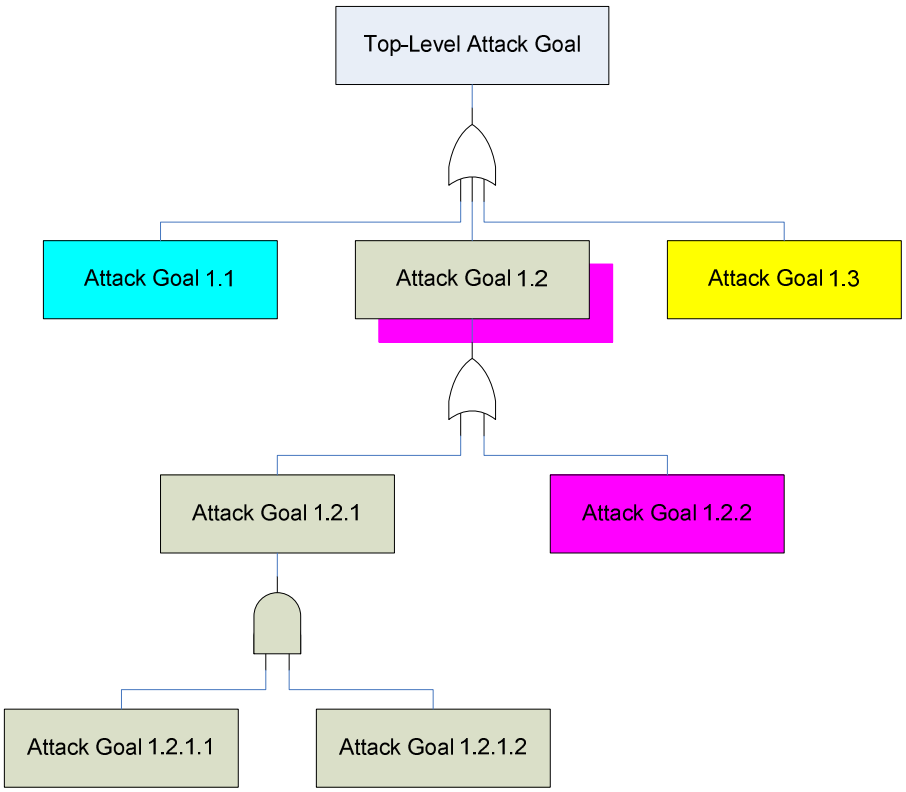


Figure 7-5 Critical path of attack tree

7.5.4 Access rights

We associate access rights with all resources. Conceptually we model these rights as a relation, suggestive of Lampson's original access control matrix [55]:

Access ⊆ Principals x Resources x Rights

Where

$$\text{Pr}\,incipals = Users \cup \text{Pr}\,ocesses$$

$$\text{Re}\,sources = \text{Pr}\,ocesses \cup Data$$

Reducing the network vulnerability with respect to access rights is a special case of abiding by Principle of Least Privilege: Grant only the relevant rights to each of the principals who are allowed access to a given resource.

7.5.5 Examples

The following are some examples of the proposed security measurement metric. Assuming our network uses the AODV protocol, for common network attacks like sniffing/snooping and message alternation, our resource table is shown:

Attack 1: sniffing/snooping

Goal: Monitoring the network for sensitive data, user name and passwords.

Resource	Attack Goal	Attack Path	Critical Path	Access right
1. Unencrypted sensitive data	Y		Y	Read
2. Unencrypted password	Y			Read
3. Unencrypted user name	Y			Read
4. Encrypted sensitive data				Read
5. Encrypted password				Read
6. Message sender		Y		Send
7. Message receiver				Receive
8. Message contain password		Y		-
9. Message contain user name		Y		-
10. Message contain sensitive data		Y		-
11. User identity	Y		Y	-

Table 7-3 AODV under sniffing attack

Pre-conditions:

- Message sends from sender node to receiver node.
- Message contains sensitive data and password.
- Message is not encrypted.

Attack sequence:

- Attacker listens to victim network.
- Sender sends message to receiver.
- Attacker capture messages before it actually reach destination receiver.

Post-conditions:

- Arbitrary, depending on the payload.

Attack 2: message alternation

Goal: Modifying a message and sending.

Resource	Attack Goal	Attack Path	Critical Path	Access right
1. Unencrypted data				Read
2. Unencrypted data	Y			Read, Write
3. Encrypted data				Read
4. Encrypted data				Read, write
5. Message sender				Send
6. Message receiver				Receive
7. Data carried by message		Y		-
8. Modified data received by receiver	Y		Y	-

Table 7-4 AODV under message alternation attack

Pre-conditions:

- Message sends from sender node to receiver node.
- Message is not encrypted.

Attack sequence:

- Attacker listens to victim network.
- Sender sends message to receiver.
- Attacker captures message before it actually reach destination receiver.
- Attacker modifies data inside of the message.
- Attacker sends message to receiver.

Post-conditions:

- Arbitrary, depending on the payload.

7.6 Security Measurement

We define the network security measurement to be a function of attack goal and attack path, the critical path associated with each type of the attack goal and attack path, and access rights that constrain the access to all resource.

$$SECURITY = f(goal, path, critical\_path, rights)$$

This equation can also be represented in the form of sum of independent resource contributions from a set of attack goal types, a set of attack path types, a set of critical path types. The attack goal types and attack path types are subject to the constraints of the access rights relation, the critical path type depends on the availability of attack goal types and attack path types:

$$SECURITY^{rights} = SECURITY_{goal}^{rights} + SECURITY_{path}^{rights} + SECURITY_{critical\_path}^{goal, path}$$

The security measurement of a network consists of the set of network actions Σ and the collective set of resources of each action σ_i. A naïve but impractical way of measuring the security is to enumerate the set of network actions of a given network and count the number of resources in each of the action's resource set. We describe below a more practical, yet meaningful way to measure the security based on the attacks of the network.

Consider a network with a fixed set, Σ, of network actions, each specified in terms of pre- and post- conditions.

Step 1: Identify the resources that are potential goals of threat agents as $\bigcup_{\sigma \in \Sigma_N} \mathrm{Re}source(\sigma)$ from the given set of network actions Σ_N. Let Type be the set of types all these resources.

Step 2: For each given threat agent, identify resource that attack is targeting as attack goal and attack path.

Step 3: Identity critical paths within the attack tree – some attack goals require attack sequences in multiple attack paths to accomplish the attack goal. Verify if all resources are available within the critical path.

Step 4: Define a penalty function P: Attack → [0, 1] to assign penalties to each resource categories identified in step 2.

Step 5: Loop through network resource set identified in step 1, determine whether each resource falls in attack resource category identified in step 2. SECURITY = SECURITY + penalty, if there is any.

Step 6: The final result indicate the overall security risk of the network. Compare the two versions of the protocol, A and B, with respect to these k threat agents to obtain their relative security risk exposure.

Figure 7-6 demonstrates the process to generate the security measurement metric on a simplified AODV under sniffing attack. This process can be interpreted as the following steps:

1. Create an attack tree of sniffing attack.
2. Create the network resources list.

3. Identify attack goals, attack paths and critical paths from the attack tree. Mark “Y” for each attack goal, attack path, and critical path in threat agent metric.

   Attack goals:

   - sensitive data
   - user identify
   - user name
   - password

   Attack paths:

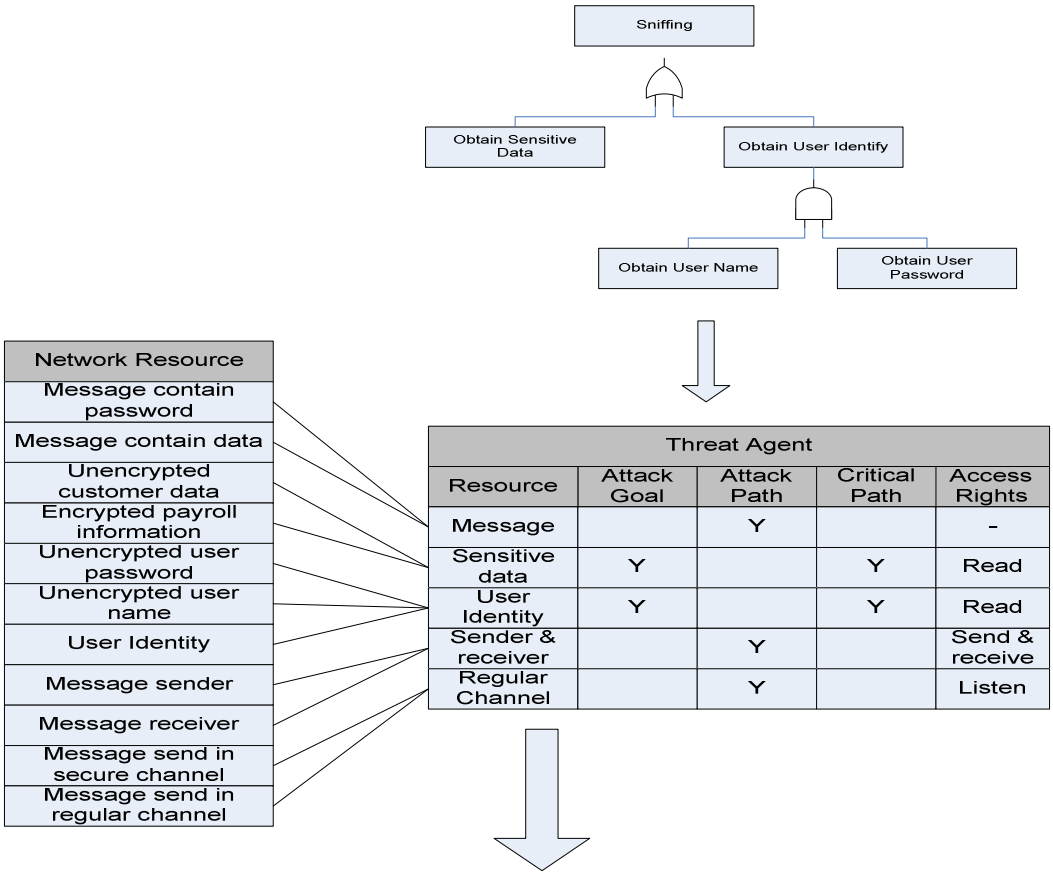
   - data message
   - message sender & receiver
   - non-secure communication channel

   Critical paths, there are two critical paths in this attack tree:

   - [obtain sensitive data]
   - [obtain user name, obtain password] -> [obtain user identity]

4. Map network resources to attack goals, attack paths, and critical paths of the threat agent.
5. Create the network under attack metric based on the network resources’ access rights.
6. Loop through each resource of network under attack metric. Penalty = Penalty + 1, if there are any “Y” marked for this resource.
7. Calculate the total of penalty column in network under attack metric.

In this case, security measurement number is 10. The higher number indicates more security risks, in other words, the network is less secure.



Threat Agent				
Resource	Attack Goal	Attack Path	Critical Path	Access Rights
Message		Y		-
Sensitive data	Y		Y	Read
User Identity	Y		Y	Read
Sender & receiver		Y		Send & receive
Regular Channel		Y		Listen

Network Under Attack					
Network Resource	Attack Goal	Attack Path	Critical Path	Access Rights	Penalty
Message contain password		Y		-	1
Message contain data		Y			1
Unencrypted customer data	Y		Y	Read	2
Encrypted payroll information				Read	0
Unencrypted user password	Y			Read	1
Unencrypted user name	Y			Read	1
User Identity	Y		Y	Read	2
Message sender		Y		Send	1
Message receiver				Receive	0
Message send in secure channel				Listen	0
Message send in regular channel		Y		Listen	1

Figure 7-6 Security metric of AODV under sniffing attack

Reducing network security risk

Our formal model and measurement method suggest ways in which we can reduce the exposure of network security risk:

- Reduce the number of network activities(for example, reduce the number of messages);
- Remove known or potential network vulnerability by strengthening the pre- and post- conditions of a network action Σ, in a way that prevents the goal of the threat from ever being achieved.
- Eliminate entire threat;
- Reduce the number of instances of threat.
- Implement more security protections.

Advantages

The use of security measurement metric has the following advantages:

- First, our metric is a relative measure of security. It is difficult to identify a yardstick for measuring a network's absolute security. Instead, we find it more practical and more useful to compare the security of two protocols of a network with a given set of attacks. Our metric can be used to determine whether a new protocol is more secure than an earlier version. Figure 7-7 demonstrates the relationship among threat agents, network vulnerabilities, and security measurement.
- Second, our metric can be used to track the security level of the network over time by measuring the threat agent and network vulnerability at regular

intervals. We can observe the change in security level as different resources are turned on and off as required.

- Third, our metric can also be used to compare the security risk of the same network protocol against different threat agents. In this case, penalty function can return different numbers depends on the resource type other than [0, 1]. Figure 7-8 demonstrates the relationship among threat agents, network vulnerabilities, and security measurement in this application.

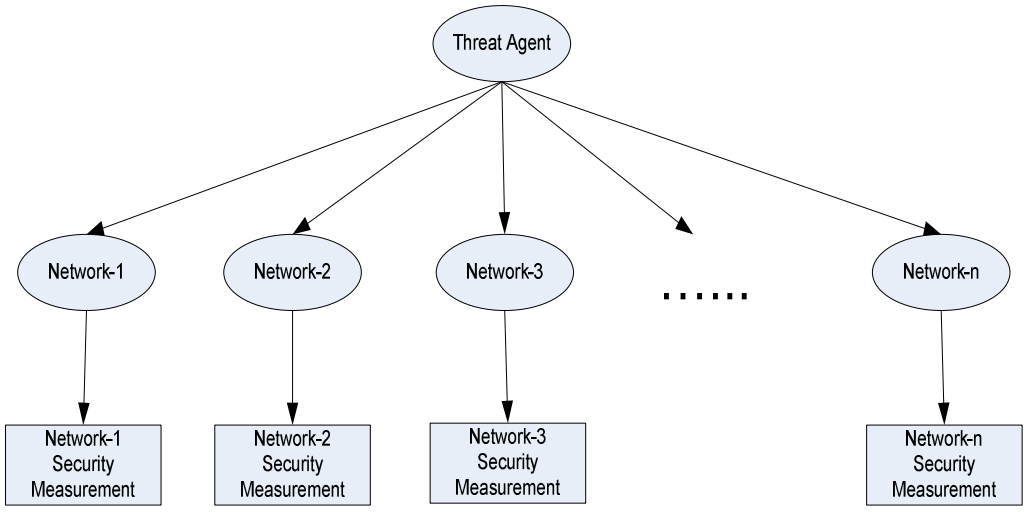


Figure 7-7 Measure security among different networks

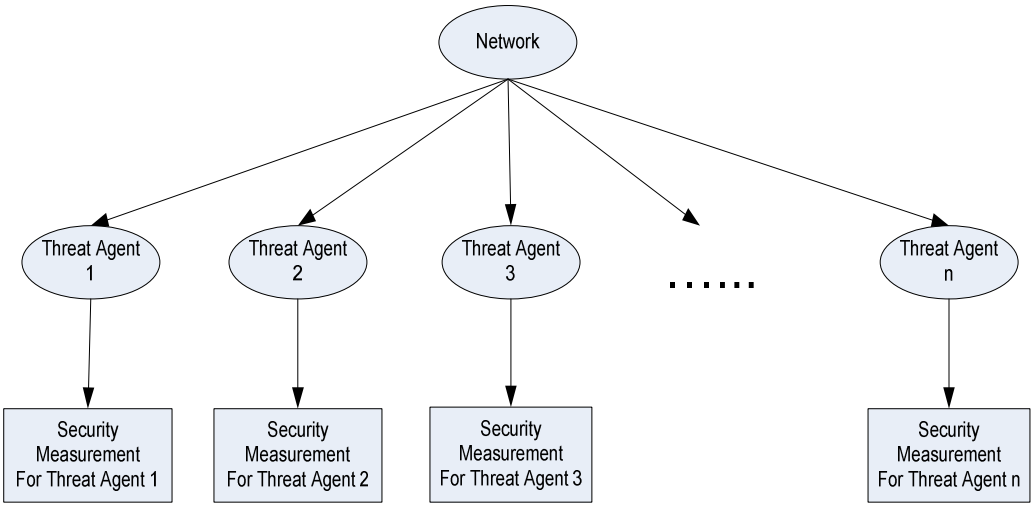


Figure 7-8 Measure security among different threats

7.7 Example of Security Measurement Metric

We used our method to measure 10 of the most common ad hoc network attacks on AODV and DSDV routing protocols, assuming both environments have the same network topology and node movement.

10 common ad hoc network attacks	Description
Sniffing/snooping	Monitoring the network for sensitive data and passwords.
Message replays	Sending a message repeatedly to a receiver (replay attack).
Message alternation	Modifying a message and sending.
Message delay and denial	Lowering or removing QoS in a network (AKA denial of service).
Spoofing	Making a packet appear to come from location other than the one from which it was sent.
Remote redirection	Remote redirection with modified route sequence number
Redirection with modified hop count	Redirection with modified hop count
Attacks using impersonation	Attacks using impersonation
Route cache poisoning	Route cache poisoning
Routing table overflow	Routing table overflow attack

Table 7-5 10 most common attacks in ad hoc network

Our calculation show DSDV has higher security risk compare with AODV, which means DSDV has higher chance to be attacked – less secure.

Attack	AODV	DSDV
Sniffing/snooping	10	10
Message replays	5	3
Message alternation	4	3
Message delay and denial	5	3
Spoofing	5	3
Remote redirection	5	3
Redirection with modified hop count	4	4
Attacks using impersonation	4	4
Route cache poisoning	0	0
Routing table overflow	6	3
Total	48	36

Table 7-6 Attack measurement of AODV and DSDV

From the above analysis we can conclude that AODV is more secure than DSDV. The table-driven protocol DSDV periodically broadcasts messages to maintain an updated routing table, whereas this is not required by AODV. The number of routing messages involved in DSDV is more than in the demand-driven protocol AODV. Since messages play a big role in facilitating a network attack path in our model, this becomes the one factor to cause DSDV to have a higher security risk than AODV. DSDV also requires each node to maintain a routing table which can be used as attack goal and even attack path for certain threat agents. Therefore, DSDV in general has higher security risk than AODV.

7.8 Conclusion

Our state machine is a general model of network behavior and attack. Our security measurement method can be applied to any network. Our application of measuring relative metric to AODV and DSDV shows the relative security of the two protocols.

Measuring security has been a long-standing challenge to the community. The need to do so has recently become more pressing. We view our work as a first step in solving this research problem. We suggest that the best way to begin is to start counting what is countable; then use the resulting numbers in a qualitative manner. Perhaps over time our understanding will then lead to meaningful quantitative metrics.

Chapter 8

Conclusions and Future Works

8.1 Overall conclusion

In this research, our main contribution has been to optimize security and QoS in Ad Hoc networks. We have studied four aspects of QoS and security in Ad Hoc networks:

- A policy based security architecture is proposed and simulation research shows that the associated overheads are not significant;
- A multi-layer QoS guided routing algorithm is shown to provide more efficient and reliable QoS;
- An architecture to optimize QoS and security provides optimum QoS and security;
- A new metric to measure the relative security of Ad Hoc protocols is presented.

The proposed on-demand security and QoS optimization architecture has been evaluated using the network simulator ns-2. The simulation results show that the proposed optimization architecture can produce similar performance as non-secure QoS routing protocol under various traffic loads. It provides more secure ad hoc networks

without compromising the QoS performance, especially under light and medium traffic conditions.

The proposed network security measurement method allows us to measure the relative security between two or more routing protocols. Our application of measuring relative metric to AODV and DSDV give results that show the relative security of the two protocols.

8.2 Policy Based Security

8.2.1 Conclusion

The proposed distributed policy based security architecture provides a flexible and scalable network security approach that fits in with the dynamic and distributed nature of ad hoc networks. Security policies can be used as a plug-in module for any of the existing ad hoc routing protocols. Simulation results indicate that although there is overhead introduced to the routing protocols; it is within an acceptable range. Given the increasing sophistication of computers, cell phones, PDAs etc., that now form ad hoc networks, as well as the increasing complexity of the services such networks provide, the proposed scheme provides a much needed additional level of security to the existing security approaches based on secure routing and intrusion detection. The proposed scheme therefore complements secure routing and intrusion detection.

8.2.2 Future Work

Future work will involve more research into the overhead of security management. Besides the communication overheads, more work is needed on identifying the memory requirements for implementing such a system. Another area than needs further work is the synchronization of distributed security policies in real-time.

8.3 Multi-layer QoS Interface Guided Routing

8.3.1 Conclusion

Quality-of-service (QoS) routing in an Ad Hoc network is difficult because network topology may change constantly, and the available state information for routing is inherently imprecise. In this dissertation, we propose a multi-layer QoS surface guided routing, which separates metrics at the different layers, MAC layer metrics, network layer metrics, and application layer metrics. Our model considers not only the QoS requirement, but also the cost optimality of the routing path to improve the overall network performance.

Achieving QoS at high mobility is a difficult problem and the simulation results indicate the proposed multi-layer QoS routing protocol can produce higher throughput and lower delay then traditional QoS routing protocols in a high mobility ad hoc network environment. The proposed protocol does not provide much improvement under low network mobility. The main drawback with the proposed protocol is the need for more internal memory at each node.

8.3.2 Future Work

Future work will analyze the factors that contribute to QoS at each layer and the amount of extra memory needed.

8.4 Security and QoS Optimization

8.4.1 Conclusion

Due to the overheads caused by implementing security in ad hoc networks, security and QoS must be considered together. In this dissertation we have proposed a distributed, flexible mechanism to optimize security and QoS in mobile ad-hoc networks. The proposed architecture is based on three components: a policy based plug-in security framework, multi-layer QoS guided routing and a proportional integral derivative (PID) controller. The multi-layer QoS surface guided routing mechanism, which separates metrics at the different layers, provides an adaptable technique for obtaining desired QoS. The policy based security framework provides a dynamic and modular approach to providing security. The simulation results indicate the proposed PID optimized security and QoS algorithm can produce similar performance as non-secure QoS routing protocol under various traffic loads. The level of security can be adaptable due to different traffic load. The best case scenario is under light traffic, where it can provide the same security level as any other secure protocols, but the same performance as non-secure QoS protocols. The worst case scenario is under extreme heavy traffic, it provides similar performance as QoS protocols, but with no security feature at all. Under medium and light traffic conditions, the proposed protocol can provide more secure networks without compromising the QoS performance.

8.4.2 Future Work

This work can be easily extended to cater a network where security is of prime importance or where both QoS and security are important based on some weightage scheme. One issue that needs investigation is the memory and computational resources required at each node to implement the proposed scheme.

8.5 Security Measurement

8.5.1 Conclusion

The proposed measurement method not only identifies the potential attack goals in the system, but also captures the attack sequence for an attack path and the relationship among different attack paths. This mechanism allows us to measure the relative security between two or more routing protocols, hence identifying the most secure protocol.

Our state machine is a general model of network behavior and attack. Our security measurement method can be applied to any network. Our application of metric and method to AODV and DSDV give results that show the relative security of two protocols.

8.5.2 Future Work

Measuring security has been a long-standing challenge to the community. We suggest that the best way to begin is to start counting what is countable; then use the resulting numbers in qualitative manner. In other words – absolute metric, rather than relative is we have done. Perhaps over time our understanding will then lead to meaningful quantitative metrics.

REFERENCES

[1] Ramanathan, R.; Redi, J. ***A brief overview of ad hoc networks: challenges and directions,*** IEEE Communications Magazine, Volume: 40 Issue: 5 Part: Anniversary, Page(s): 20 -22, May 2002.

[2] Broch, J.; Maltz, D. A.; Johnson, D. B.; Hu, Y. C. and Jetcheva, J. *A* ***Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols***, Proc. of the Fourth Annual, Oct 1998.

[3] Perkins, C. E.; Bhagwat, P. ***Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,*** ACM SIGCOMM Symposium on Communication, Architectures and Protocols, 1996.

[4] Johnson, D.; Maltz D.; Broch Josh. ***The dynamic source routing protocol for mobile ad hoc networks (Internet-Draft)***, Mar, 1998.

[5] Perkins, C. E.; Royer, E. M. ***Ad hoc on demand distance vector (AODV) routing (Internet-Draft)*** Aug 1998.

[6] Manel Guerrero Zapata and N. Asokan, ***Securing ad hoc routing protocols***, Proc. International Conference on Mobile Computing and Networking, pp. 1-10, 2002.

[7] Y. C. Hu, A. Perrig, and D. B. Johnson, ***Ariadne: A secure on-demand routing protocol for ad hoc networks***, Proceedings of 8th annual international conference on Mobile Computing and Networking, page 12-23, 2002.

[8] Y. C. Hu, D. B. Jhonson, and A. Perrig, ***Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks,*** Proceedings Fourth IEEE Workshop on Mobile Computing Systems and Applications, 2002.

[9] U. Lu, B. Pooch, ***Cooperative security enforcement routing in mobile ad hoc networks***, Proc 4th International Workshop on Mobile and Wireless Communications Networks, Vol., I., pages 157-161, 2002.

[10] P. Papadimitratos and Z. Haas, ***Secure routing for mobile ad hoc networks***, Proc of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, Jan 2002.

[11] S. Yi, P. Naldurg, and R.Kravets, ***Security-aware ad hoc routing for wireless networks***, Proceedings 2nd ACM international symposium on Mobile ad hoc networking & computing, pages 299-302, 2001.

[12] R. Bobba, L. Eschenuauer, V. Gligor, and W.Arbaugh, ***Bootstrapping security associations for routing in mobile ad-hoc networks***, Technical report, University of Maryland, May 2002.

[13] Y.-C. Tseng, J.-R. Jiang, and J.-H. Lee, ***Secure bootstrapping and routing in an ipv6-based ad hoc network***, Proc ICPP Workshop on Wireless Security and Privacy, 2003.

[14] Harold Zheng Sherry Wang Robert A. Nichols, ***Policy-Based Security Management For Ad Hoc Wireless Systems*** Proc MILCOM, 2005

[15] S. Chen and K. Nahrstedt, ***An overview of quality-of-service routing for the next generation high-speed networks: problems and solutions***, IEEE Networks, Special Issue on Transmission and Distribution of Digital Video, pp 64-79, Nov./Dec. 1998.

[16] B. Awerbuch, Y. Azar, S. Plotkin, and O. Waarts, ***Throughput competitive online routing***, in Proc. 34th Ann. Symp. Fundations of Computer Science, Palo Alto, CA, Nov. 1993.

[17] S. Chen and K. Nahrstedt, ***On finding multi-constrained path***, in Proc. IEEE ICC'98, pp. 874-879.

[18] R. Guerin and A. Orda, ***QoS based routing in networks with inaccurate information: Theory and algorithms***, in Proc. IEEE INFOCOM'97, Japan, pp. 75-83, 1997.

[19] J. Moy, *"OSPF Version 2,"* Internet RFC 1583, Mar, 1994.

[20] I. Cidon, R. Rom, and Y. Shavitt, ***Multi-path routing combined with resource reservation***, in Proc. IEEE INFOCOM'97, Japan, pp. 92-100, 1997.

[21] C. Hou, ***Routing virtual circuits with timing requirements in virtual path based ATM networks***, in Proc. IEEE INFOCOM'96 pp. 320-328, 1996.

[22] H. F. Salama, D. S. Reeves, and Y. Viniotis, ***A distributed algorithm for delay-constrained unicast routing***, in Proc. IEEE INFOCOM'97, Japan, pp. 84-91, 1997.

[23] K. G. Shin and C. C. Chou, ***A distributed route selection scheme for establishing real time channel***, in Proc. 6th IFIP Int. Conf. High Performance Networking (HPN'95), 1995.

[24] ATM Fourm, ***Private Network Network Interface (PNNI) v1.0 Specifications***, June 1996.

[25] D. H. Lorenz and A. Orda, ***QoS routing in networks with uncertain parameters***, in Proc. IEEE INFOCOM'98.

[26] 17Vineet Srivastava and Mehul Motani, ***Cross-Layer Desing: A Survey and the Road Ahead,*** IEEE Communications, Vol. 43, No. 12, Dec 2005.

[27] 18Hai Jiang, Weihua Zhuang and Xuemin Shen, ***Cross-Layer Desing for Resource Allocation in 3G Wireless Networks and Beyond,*** IEEE Communications, Vol. 43, No. 12, Dec 2005.

[28] Taesoo Kwon, Howon Lee, Sik Choi, Juyeop Kim and Dong-Ho Cho, ***Design and Implementation of a Simulator Based on a Cross-Layer Protocol between MAC and PHY Layers in WiBro Compatible IEEE 802.16e OFDMA System,*** IEEE Communications, Vol. 43, No. 12, Dec 2005.

[29] Chien-Chao Tseng, Li-Hsing Yen, Hung-Hsin Chang and Kai-Cheng Hsu, ***Topology-Aided Cross-Layer Fast Handoff Designs for IEEE 802.11/Mobile IP Environments,*** IEEE Communications, Vol. 43, No. 12, Dec 2005.

[30] Wei Lianp and Wenye Wang, ***An Analytical Study on the Impact of Authentication Local Area Networks***, Proc. IEEE 13th International Conference on Communications and Networks (ICCCN'04), Oct 2004.

[31] Wei Liang and Wenye Wang, ***A Quantitative Study of Authentication Networks***, Proc. IEEE INFOCOM, 2005.

[32] Wei Liang and Wenye Wang, ***On Performance Analysis of Challenge/ Authentication in Wireless Networks***, Computer Networks Journal, 2005.

[33] Wenye Wang, Wei Liang, and Avesh K. Argawal, ***Integration of Authentication Management in Third Generation and WLAN Data Networks***, Proceedings Communications and Mobile Computing (WCMC) - special issue on WLAN/ Generation Heterogeneous Mobile Data Networks, Dec 2004.

[34] K. Jensen, Coloured Petri Nets. ***Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts*** Monographs in Theoretical Computer Science, Springer-Verlag, 1997.

[35] Simon Godik, and Tim Moses, ***OASIS Extensible Access Control Markup Language (XACML)*** in Techincal Committee Working Draft, Version 15, July 2002.

[36] S. McCanne and S. Floyd. ***Ns-Network Simulator,*** http://www-mash.cs.berkely.edu/ns/. [Last accessed Oct 2006]

[37] Chenxi Zhu, ***Medium Access Control and Quality-of-Service Routing for Mobile Ad Hoc Networks***, PhD thesis, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20906, 2001.

[38] B.C. Kuo, ***Automatic control Systems***, Upper Saddle River, NJ, Prentice-Hall, 1997.

[39] Baoxian Zhang and Hussein T. Mouftah, ***QoS Routing for Wireless Ad Hoc Networks: Problems, Algorithms, and Protocols,*** IEEE Communications Magazine, pp 110-115, Vol 43, Issue 10, Oct 2005.

[40] S. H. Shah and K. Nahrstedt, ***Predictive Location-based QoS Routing in Mobile Ad Hoc Networks***, Proc. IEEE ICC '02, pp. 1022–27, Apr 2002.

[41] P. Sinha, R. Sivakumar, and V. Bharghavan, ***CEDAR: A Core-extraction Distributed Ad Hoc Routing Algorithm,*** IEEE JSAC, vol. 17, no. 8, pp. 1454–65, Aug 1999.

[42] C. R. Lin and J.-S. Liu, ***QoS Routing in Ad Hoc Wireless Networks,*** IEEE JSAC, vol. 17, no. 8, pp. 1426–38, Aug 1999.

[43] C. Zhu and M. S. Corson, ***QoS Routing for Mobile Ad Hoc Networks***, Proc. IEEE INFOCOM '02, pp. 958–67, June 2002.

[44] S. Chen and K. Nahrstedt, ***Distributed Quality-of-service Routing In Ad Hoc Networks***, IEEE JSAC, vol. 17, no. 8, pp. 1488–505, Aug 1999.

[45] B. Zhang and H. T. Mouftah, ***QoS Routing Through Alternate Paths in Wireless Ad Hoc Networks***, Int'l. J. Commun. Sys., vol. 17, no. 3, pp. 233–52, Mar 2004.

[46] S. Jha, O. Sheyner, J. Wing, ***Two Formal Analyses of Attack Graphs,*** Proceedings of the 15th IEEE ComputerSecurity Foundations Workshop, Nova Scotia, Canada, June 2002.

[47] Michael Howare, ***Fending off future attacks by reducing the attack surface,*** http://msdn.microsoft.com/library, 2003. [Last accessed Oct 2006]

[48] Michael Howare, Jon Pincus and Jeannette M. Wing, ***Measuring relative attack surface,*** Proceeding of Workshop on Advanced Developments in Software and System Security, 2003.

[49] Avizienis, A., et al., ***Basic concepts and taxonomy of dependable and secure computing,*** IEEE Trans. Dependable and Secure Computing, **1**(1): p. 11-33, 2004.

[50] Ibrahim, L., et al., ***Safety and Security Extensions for Integrated Capability Maturity Models,*** US FAA & DoD, Sep 2004.

[51] Landwehr, C.E., ***Computer Security,*** International Journal on Information Security (IJIS), **1**: p. 3-13, 2001.

[52] Jones, A.n. and D. Ashenden, ***Risk Management for Computer Security: protecting your network and information assets,*** Oxford: Elsevier. 274, 2005.

[53] Howard, M. and D. LeBlanc, ***Writing Secure Code,*** Microsoft Press International. 800m, 2003.

[54] J. Alves-Foss and S. Barbosa, ***Assessing computer security vulnerability,*** ACM SIGOPS Operating Systems Review 29, 3, p. 3-13, 1995.

[55] S.T. Eckmann, G. Vigna, and R.A. Kemmerer, ***STATL: An attack language for state-based intrusion detection,*** Journal of Computer Security 10, 1/2, p. 71-104, 2002.

APPENDICES

APPENDIX A

Glossary

802.11:	A standard for wireless local area networks (WLAN) developed by a working group of the Institute of Electrical and Electronics Engineers (IEEE).
802.11a:	An international IEEE standard for WLAN networks, operating at 5 GHz and providing 54Mbps. Range up to 30m.
802.11b:	An international IEEE standard for WLAN networks, operating at 2.4 GHz and providing 11 Mbps. Range up to 100m.
802.11g:	An international IEEE standard for WLAN networks, operating at 2.4 GHz and providing 54Mbps. Range up to 100m.
Access Point:	A transceiver or radio component in a wireless LAN that acts as the transfer point between wired and wireless signal, and vice versa. The Access Point (AP) is connected to antennas as well as to the wired LAN system.
Ad-Hoc Network:	An Ad-Hoc wireless LAN is a group of computers each with wireless adapters, connected as an independent wireless LAN.
Bridge:	A device which connects two or more networks.

Media Access Control Address (MAC Address): The unique physical address of each device's network interface card.

Repeater: A device used in a network to strengthen a signal as it is passed along the network cable.

Router: An active network component that connects one network to another network. Routers work with packets that include logical addressing information.

Service Set Identifier (SSID): Service set identifier. A unique identifier that stations must use to be able to communicate with an access point. The SSID can be any alphanumeric entry up to a maximum of 32 characters.

SSID Broadcasting: To “announce” the Access Points presence by broadcasting the SSID.

Transmission Control Protocol / Internet Protocol (TCP/IP): The protocols, or conventions, that computers use to communicate over the Internet.

Wi-Fi Protected Access (WPA): A system to secure Wi-Fi networks, intended to replace the current, less secure WEP (Wired Equivalent Privacy) system. Part of the IEEE 802.11i standard.

Wired Equivalent Privacy (WEP): An encryption system that encrypts data on wireless networks that can only be read by authorized users with the correct decryption key.

Wireless Fidelity (WI-FI): Another name for IEEE 802.11b. A wireless networking technology for PCs and PDAs that allows multiple devices to share

a single high-speed Internet connection over a distance of about 300 feet.

Wireless Local Area Network (WLAN): A wireless LAN is one in which a mobile user can connect to a local area network (LAN) through a wireless (radio) connection.

Wireless Network: A network in which data is transmitted without wires, increasing mobility of the user and their access to data.

VITA

Zhengming Shen

Candidate for Degree of

Doctor of Philosophy

Dissertation: ON-DEMAND SECURITY AND QoS OPTIMIZATION IN MOBILE AD HOC NETWORK

Major Field: Computer Science

Biographical:

Personal Data: Born in Hangzhou, Zhejiang, P.R. China, On December 31, 1972, the son of Mr.Jintao Shen and Mrs.Xianwei Sun.

Education: Completed the requirements for the Doctor of Philosophy degree with a major in Computer Science at Oklahoma State University, Tulsa, Oklahoma in December 2006.

Received Master of Science degree with a major in Computer Science at Oklahoma State University, Tulsa, Oklahoma in June 2003.

Received Bachelor of Engineering in Computer and Applications by the Hangzhou Institute of Electronics Engineering, Hangzhou, Zhejiang, P. R. China in April 10, 1997.

Experience: Employed as Technical Lead, Information Technology Department, Dollar Thrifty Automotive Group, Inc., 2005 to Present.

Employed as Senior System Analyst, Information Technology Department, Dollar Thrifty Automotive Group, Inc., 2003 to 2005.

Employed as Program Analyst, Information Technology Department, Dollar Thrifty Automotive Group, Inc., 2000 to 2003.

Name: Zhengming Shen Date of Degree: December, 2006

Institution: Oklahoma State University Location: Tulsa, Oklahoma

Title of Study: ON-DEMAND SECURITY AND QoS OPTIMIZATION IN MOBILE AD HOC NETWORKS

Pages in Study: 162 Candidate for the Degree of Doctor of Philosophy

Major Field: Computer Science

Scope and method of Study: Security often comes with overhead that will impact link Quality of Service (QoS) performance. In this dissertation, we propose an on-demand security and QoS optimization architecture in mobile ad hoc networks that automatically adapts network security level to changes in network topology, traffic condition, and link QoS requirements, so as to keep the security and QoS at optimum conditions. In order to achieve the overall objective, we introduce three basic frameworks: a policy based plug-in security framework, a multi-layer QoS guided routing algorithm, and a Proportional Integral Derivative (PID) feedback control based security and QoS optimization framework. The research has been evaluated with the network simulator ns-2. Finally, we propose an attack tree and state machine based security evaluation mechanism for ad hoc networks: a new security measurement metric.

Findings and Conclusions: Simulations have been done for small and large network sizes, low and high communication ratios, as well as low and high mobility scenarios. The simulations show that the proposed on-demand security and QoS optimization architecture can produce similar performance to non-secure QoS routing protocol under various traffic loads. It provides more secure ad hoc networks without compromising the QoS performance, especially under light and medium traffic conditions.

ADVISOR'S APPROVAL: _____Dr. Johnson Thomas_____