An Application Layer Non-Repudiation Wireless System: A Cross-Layer Approach

by

Sasan Adibi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Signature

Abstract

Non-repudiation techniques are to ensure any communication taking place between two or more parties will be undeniable. Therefore it is crucial to include digital signatures of the involving parties while the communication is taking place. In medical practices, involved parties include; patient(s), doctor(s), pharmacist(s), who are involved in series of visits, diagnosis, prescriptions, and possible operations. To avoid possible conflicts, deploying non-repudiation techniques help immensely. This thesis considers this issue in a wireless medium and studies the Quality of Service (QoS)/Security requirements in terms of network parameters and performance metrics.

In terms of research contributions, this thesis embodies a thorough research on layered and cross-layer QoS and security schemes, in particular, featuring an adaptive Forward Error Correction (FEC) at the application layer, adapting to channel conditions. This leads to a cross layer design, which considers various QoS and security parameters export and import to and from various layers with a special focus on the application layer.

The aim of this thesis is to consider a practical implementation and associated complexities of a non-repudiation system, including analytical and experimental testbeds and results. The security schemes are based on Suite-B cryptographic algorithms, including: The Elliptic Curve Diffie-Hellman (ECDH) for key agreement, the Advanced Encryption Standard - Galois/Counter Mode (AES-GCM) for encryption and authentication, the Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signatures, and the Secure Hash Algorithm (SHA) for integrity. A key aspect of Suite-B is the deployment of Elliptic Curve Cryptography (ECC).

The non-repudiation aspect of this thesis is based on the Suite-B's digital signature scheme; ECDSA. The digital signature and the hashing function target the entire multimedia data (i.e., text, video, and voice) and the challenge is to offer such extensive security treatment, while guaranteeing certain Quality of Service settings. These settings include: minimum round trip delay, maximum overhead, and minimum bandwidth allocation.

Acknowledgements

I would like to thank Prof. Gordon B. Agnew for his supports throughout my Ph.D. program. His comments, suggestions, and feedbacks were essential, very constructive, and thorough. I would also like to thank Dr. Tom Tofigh and WiMAX Forum for their support and collaboration.

Dedication

I would also like to thank my wife; Negar, who has been extremely supportive and encouraging and my mother, Shahla Ejtemai, who has always believed in me. A big thank to all my friends (too many to mention) who created everlasting memories during this episode of my life in Kitchener/Waterloo area.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

The goal of this thesis is to present a novel and systematic approach to a P2P (Point-to-Point) non-repudiation and adaptive-multimedia-based communication. A non-repudable communication often involves a transparent mechanism through which, no party involved in the communication can deny having participated in the whole or a part of the communication. Therefore security is one of the key points being considered in this thesis. As traffic between involved parties is multimedia-enabled, (including voice, video, text, file-sharing.), another important key point is Quality of Service (QoS). Quality of Service guarantees certain performance measures from both network and user points of views. Such a guarantee of service is of great importance due to the fact that many communications conveying multimedia traffic payloads are often time-sensitive and require certain network-centric performance guarantees. These network-centric parameters include; delay, jitter (variable delay figures) and bandwidth, which require guaranteed maximum delay and jitter figures and minimum bandwidth for proper performance.

The multimedia-enabled, non-repudiation system under consideration, involves at least two communication parties; namely Party A and Party B. The traffic flows between these two parties will include a number of QoS- and security-based elements, with varying payload sizes, depending on the specific QoS/security settings. Through traffic analysis and traffic classification concepts, the average packet payload sizes on both wired and wireless links are evaluated and compared to the proposed system's varying packet payload sizes.

1.1 Problem Definition

Security requirements for current communication systems are becoming more demanding as the nature of attacks has become more complex, requiring more versatile security audits and measures. Taking QoS into account adds more to the complexity, due to the fact that security measures usually add redundant information (overhead) and delay figures, which may undermine QoS schemes. Therefore having a system where both security and QoS can coexist may be a challenge and involving power consumption constraints to the picture will add more dimensions to the mentioned challenges. The challenges involving security, QoS, and power consumption figures will always exist, as long as both wireless broadband communication systems (with limited power resources) and the associated level of attacks, are progressing alongside one another.

A challenge is to maintain an application layer, offering an end-to-end adaptive QoS scheme and simultaneous security supports for a P2P-non-repudiation system, within acceptable power consumption figures, where both security-related data (e.g., integrity, authentication, non-repudiation) and multimedia-based (e.g., audio, video, text) traffic payloads are transmitted simultaneously. An application layer-based provisioning provides flexibility and increased performance compared to lower layer provisioning schemes and to add more flexibility, an adaptive scheme based on the quality of the channel is added to accommodate multimedia encoding quality with the channel capacity.

National Security Agency (NSA) initiated three efforts to address widespread cryptographic interoperability and security issues, one of which is [1]; Cryptographic Interoperability Strategy (CIS) in which, Suite-B algorithms are one of CIS's core to protect both classified and unclassified information. An end-to-end security mechanism may include the deployment of Suite-B cryptographic algorithms, including: the Elliptic Curve Diffie-Hellman (ECDH) for the key agreement, the Advanced Encryption Standard - Galois/Counter Mode (AES-GCM) for the encryption-authentication, Elliptic Curve Digital Signature Algorithm (ECDSA) for the digital signatures, and the Secure Hash Algorithm (SHA) for message digest and integrity schemes. Suite-B uses Elliptic Curve Cryptography (ECC) exclusively for key exchange and digital signature [2].

The system under consideration accounts for a P2P (which can be extended to cover Point-to-MultiPoint "P2MP" and MP2MP) communication, which uses UDP (User Datagram Protocol) as its transport engine between Parties A and B, transporting QoS- (cross-layer) and security- (hash, encrypted data, and digital signature) related parameters, offering a secure bounded delay, jitter, with minimum throughput communication.

This thesis features a multilayer Suite-B support, which addresses various security requirements at the application, transport, and network layers. The added overheads and incurred delay figures related to such a deployment are calculated and compared with experimental results using Crypto++ 5.6.0 codes based on C++ [3] and Labview V.8.5 [4] (which is a real traffic simulator) implementations. Comparisons between experimental and theoretical results, as well as possible attacks on the system are also provided. Many current researches offer limited performance analyses, therefore a thorough end-to-end performance analysis for a system offering QoS to support multimedia-enabled traffic in the presence of security has so far not been considered in the literature, which is being covered in this thesis.

1.2 Contributions

The contributions of this thesis are listed in the following categories:

1.2.1 Application Layer Provisioning

This thesis covers details on a well organized QoS and security enabled application layer-based system, where QoS and security related parameters are exported from different layers and imported at the application layer, while incorporating multimedia-enriched traffic in the presence of security-enabled data with an emphasis on the non-repudiation support.

1.2.2 Novel Cross-Layer Design

In the structure of the proposed system, a novel cross-layer design approach is deployed, which serves for both application layer-based QoS and security purposes.

1.2.2.1 QoS-based Cross Layer Design

In the cross-layer QoS-based system, a number of QoS-related parameters (e.g., Received Signal Strength Indicator “RSSI”, Signal to Noise Ratio “SNR”, Wireless Multimedia, “WMM”, and Differentiated Services Code Point “DSCP”) are gathered from three layers and imported at the application layer. The current values of these three parameters define the quality of the wireless link and are used to determine the quality of the multimedia encoder. This way, an adaptive multimedia encoder/decoder system is designed and deployed, which adapts its coding quality to the quality of the environment and adjusts the encoder bit-rate and the integrated Forward Error Correction (FEC) scheme accordingly. The values of these three parameters are transmitted alongside the rest of the data to the receiver and the receiver uses them to determine the correct decoding procedures. The effectiveness of the cross-layer design will also be evaluated.

1.2.2.2 Security-based Cross Layer Design

In the cross-layer security-based scheme, a number of layered-based security parameters are gathered from various layers (i.e., , MAC, network, and transport layers), which are accompanied with other gathered QoS-related parameters and fed into an application layer security-based module used by the multimedia encoder. The security/QoS cross-layer feedbacks at the application layer will determine the multimedia encoder quality, as well as the error correction scheme.

1.2.2.2.1 Multilayer Suite-B Cryptographic Schemes Deployment

A multilayer Suite-B support has been adopted in this thesis to address various security requirements at the application, transport, and network layers. A detailed analysis is provided to show the impacts of Suite-B deployment to the overhead and delay figures. Non-repudiation functionality is one of the schemes supported by Suite-B algorithms.

1.2.3 Adaptive Forward Error Correction (A-FEC) Scheme

The cross-layer parameters are imported at the application layer from lower layers and as mentioned, an adaptive mechanism is present at the application layer, which processes the cross-layer information and its outputs are directly used in the encoder (at the sender), decoder (at the receiver) systems, and in the Forward Error Correction (FEC) scheme, adapting to channel and network conditions.

1.2.4 Adaptive Multimedia Encoder Selection

As mentioned, based on the imported cross-layer parameters, multimedia (i.e., voice and video) encoders are selected according to the current quality of the channel/signal.

Following the functions of the encoder and the FEC module, Suite-B algorithms are applied to the multimedia traffic.

Finally, a thorough analysis is performed to investigate the performance of this system under various security and QoS conditions.

1.2.5 Security System Analysis

The security model of this system is discussed in chapter 3. This includes a thorough consideration of the security algorithms and protocols used in various layers as well as security analyses of the system.

1.2.6 QoS-Security Testbed Evaluations

Through analyses and experimental results using C++ based cryptographic codes and Labview testbeds, QoS capabilities offered by the system are investigated under various traffic flows, containing: Security-enabled data (i.e., authenticated, encrypted, hashed, and digitally signed information), multimedia data, and cross-layer information.

1.3 Thesis Organization

Chapter 2 covers the background literature including: Traffic classifications, cross-layer, security (e.g., non-repudiation), and multimedia communication related concepts.

Chapter 3 presents the security and QoS models. The security model discusses the various security parameters and algorithms used in this system and the QoS model also discusses the QoS-related parameters.

Chapter 4 contains the analytical and experimental results. In this chapter we show that the proposed system is capable of offering simultaneous QoS and security capabilities.

Chapter 5 provides conclusions, followed by references and appendixes.

Chapter 2
Background and Literature Survey

In this chapter, the legacy and current approaches to wireless QoS- and security-enabled multimedia services are considered. In this chapter we will review the background literature of the following areas: *QoS, security traffic classifications, wireless multilayer approach, wireless cross layer design,* and *non repudiation multimedia-based wireless systems.*

2.1 Quality of Service (QoS)

In a best-effort scenario, all traffic flows have the same priority. QoS, on the other hand, provides priority for flows that fall in specific criteria or groups. QoS has become an important requirement in enterprise networks as well as in scenarios where both wireless and wired networks are deployed. This includes QoS schemes for transmitting aggregation of packets, admission control, and other scenarios where packets and flows may compete for channel access.

QoS comes in two main forms; network-perspective-QoS [5] and user-perspective-QoS; which is also known as; Quality of Experience (QoE) [6]. A few network-perspective-QoS parameters include; throughput, end-to-end delay, and jitter, which are explained in the following subsections.

2.1.1 QoS versus QoE

QoS, as mentioned, is the network-centric performance consideration, including: Delay, jitter, bandwidth, etc. QoS measures are used to quantify the quality schemes configured at each layer and the results of layers interactions with one another and the impact on the overall delay/jitter/throughput figures for the purpose of improving data flows.

QoE, on the other hand, is the measure of the quality from the human experience point of view. Therefore what human hears and sees have direct impacts on the QoE. Therefore fidelity and latency (delay and jitter) are two very important factors characterizing QoE.

There is usually a direct relationship between QoS and QoE in a sense that the improvement of QoS figures may result in a better quality perception by the user. This is due to the fact that some parameters (e.g., delay and jitter) are used in both to measure the quality, however different criteria are given for each. This thesis is more concerned with the QoS measurement.

QoS schemes are used to actively measure the quality of voice in Voice over IP (VoIP) systems, using delay and jitter figures to calculate the MOS (Mean Opinion Score, introduced in section 2.5.2.3). QoE, on the other hand, uses a non-intrusive voice quality analysis for predicting MOS value using a passive voice clarity evaluation. This may involve measuring delays based on vocal methods, deploying audio recording and playback, and simple analyses that identify impairments in the voice quality.

2.1.2 Bandwidth or Throughput

The definitions for bandwidth and throughput differ in the sense that bandwidth is the maximum number of bits transmitted between two physical end-points per unit of time (second) [7], whereas throughput (effective bandwidth), is the actual payload transmission per unit of time [8]. Throughput is typically less than the related bandwidth.

The difference between bandwidth and throughout becomes increasingly noticeable in the presence of noise. As the noise level increases, it degrades the throughput figures, while the bandwidth may not be noticeably affected.

Throughput is comprised of actual data, control/parity bits, and other retransmission information. Goodput however contains useful bits transmitted per second. From the application perspective, throughput refers to the data rate (bits per second) generated by the application. Therefore application throughput and goodput are synonyms.

Throughput is sometimes called bit-rate, which is considered to be a network resource parameter that needs to be properly and efficiently managed and allocated. Most often, priority has to be assigned (through proper QoS provisioning) to ensure larger amounts of throughput are allocated to more critical data types [9].

2.1.3 End-to-End Delay (E2ED) or Round-Trip Time (RTT)

Delay is the main parameter that directly impacts both network related QoS as well as the users' satisfactions, particularly for real-time applications. In real-time applications, data is required to be delivered from the source to the destination within a certain time frame. Long delays may increase the chances for network faults and error messages, such as: TCP timeouts and ICMP error messages (e.g., destination unreachable), which in turn reduce the efficiency of multimedia traffic transmission, thus resulting a reduction in audio/video fidelity [10]. Moreover, this can cause user frustration during interactive communications. While data traffic is transmitted through various segments and devices (e.g., switches and routers) throughout the communication networks that interconnects the source to the destination, every segment/device introduces a specific amount of delay figure. These delays include [11]: *Source-processing delay (*including *packetizing and digitization delays),* which is a delay introduced by the source that generates the packets, which also depends on the source hardware configuration (e.g., CPU, RAM, bus-rate) and the number of applications running at the same time. *Transmission delay,* which is the packet's transmission time. This is a function of the packet size and transmission speed. *Network delay,* which is comprised of *propagation, protocol, output queuing,* and *destination processing delays*. *Propagation delay* is a function of the physical distance between the source and the destination. *Protocol delay* is a delay caused by the specific communication protocol executed in different network components such as; routers, gateways, and network interface cards. This type of delay depends on the protocol, the load of the network, and the configuration of the hardware that executes the protocol. The *output queuing delay* is the delay caused by the time duration that a packet spends in the output queue. For example, such a delay can be incurred at an intermediate router output queue. This delay depends on the network congestion condition, the configuration of the

hardware, and the link speed. The *destination processing* delay is a delay introduced by the destination processing components. For example, such a delay can be incurred in the packet reconstruction process. Similar to the source processing delay, this delay depends on the destination host hardware configuration and specifics of the load.

2.1.4 Jitter

When discussing network-related parameters, latency is often presented as an average value. A multimedia protocol, such as VoIP, which is time-sensitive, uses IP for the control signaling and data transfer. In a scenario where packets are delivered using IP, there is no guarantee that every voice data packet will travel over the same path, unlike in the circuit-switched network scenario. Packet switched networks are comprised of numerous node and when two nodes wish to communicate, packets may be forced to take different routes, based on network conditions, unavailability of intermediate nodes, and the routing protocol in use. Therefore it's common that some packets are forced to take paths with more or less hops than other packets, in which case, packet arrival times may encounter variable delays, causing much higher latency effect, called jitter [12].

Another issue that contributes to a higher jitter figure can be related to having too many packets processed at the intermediate gateways/routers, which may overwhelm the processing duties momentarily, causing jitter effects.

2.2 Security

Security is a set of mechanisms, which provides the capability to support: privacy, confidentiality, integrity, availability, non-repudiation, authenticity, and access control, for the legitimate users and/or the information transmitted among them.

There are interesting interactions between QoS and security, in which strengthening one may often result in weakening of the other [13], especially in wireless technologies, where resources are scarcer. This was traditionally the case as wireless systems had relatively limited powers to perform frequent extensive computational mechanisms and tasks and to serve best practices for simultaneous QoS and security deployments.

However thanks to the advances in the wireless broadband technologies, networks have evolved (e.g.., faster processors, cheaper memory, higher computational power capabilities, and higher bandwidth allocations), which help us design network structures with coexisting QoS and security schemes. From the infrastructural point of view (concerning bandwidth allocation), the realization of 802.11n, 3G and 4G access technologies contributed immensely to allocate enough bandwidth for security- and QoS-enabled multimedia-enriched traffic payloads.

Therefore the QoS-security coexistence and their possible implications are important aspects to be considered. In this thesis, security is addressed in a multilayer/cross-layer sense. The cross-layer mechanism will have the security/QoS parameters created at different layers accessible at the application layer. Therefore in the design of our system, security, QoS, and their coexistence will be considered step-by-step.

The following subsections discuss the fundamentals of security. The most basic requirements of any secure system should include capabilities to prevent common passive and active attacks, through the following functionalities [14, 15, 16, 17, 18]:

2.2.1 Confidentiality

Confidentiality or *privacy* is the ability to secure the content of the information communicated among parties. Enabling confidentiality (through encryption) should prevent the intruder from recovering any data (data confidentiality). In a broader sense, an intruder should not be able to determine the parties involved in a communication (user confidentiality) or whether a communication session has ever been established. This can be achieved by encrypting the source and destination addresses or by using a VPN (Virtual Private Network) tunneling scheme, such as IPSec (IP Security) [19].

2.2.2 Data Integrity

Integrity is a mechanism that safeguards accuracy, reliability, and completeness of information assets. Data integrity, in its simplest form, prevents unnoticed modification

of the communicating data by an unauthorized user. In a broader sense, integrity should ensure that data is current, unaltered, and undamaged, therefore it guards against unwanted alteration (e.g., addition, deletion, and undue delays) [20].

2.2.3 Authentication

Authentication is a very important security requirement, which provides the facility to verify the identity of parties taking part in a communication. There are three types of authentication procedures: *Entity (user/address) authentication*, where the identity of an entity, a device, or a person is checked to ensure legitimacy before the communication begins. *Geo-authentication* is a process by which the location of a node or another location-dependent attribute is checked before the actual communication takes place. *Attribute authentication* is a process of establishing confidence in an attribute, an action, or a certain property associated with an entity. *Data authentication* is the capability of the authorized parties to ascertain the authenticity of the received data.

It is an important security requirement to have a mutual authentication scheme, where both communication parties are required to be authenticated to one another. This is to reduce the chances of a Man-in-the-Middle-Attack (MitMA) [21]. In a MitMA scenario, an illegitimate user is able to intercept communications between legitimate end-points or users and based on the information acquired, it can either take over the already established link or establish a new link with a legitimate user. In this scenario, the illegitimate user can act as either a client or a server and steal credentials of a legitimate client or server.

2.2.4 Authorization

Authorization is a mechanism that checks the legitimacy of a device or process before a communication begins. In a client-server scenario, authorization gives the server permission to seek credentials from the client, however authentication proves the identity of the client to the server. It is possible to have authentication without authorization, however, one cannot have authorization without authentication [14, 22].

2.2.5 Non-Repudiation

A non-repudiation mechanism is the ability to prevent an authorized user from denying its involvement in a previous communication or activity. This is further subcategorized as: *Protection against sender denial*, which prevents the sender from denying its role in sending the transmitted information. *Protection against forward denial* that prevents against the denial of forwarding entities on the path, disputing the fact that they received any data to forward and the fact that they did forward the data. *Protection against delivery denial***,** which protects against disputing final delivery of data to the destination. *Protecting against receiving denial*, which protects against the recipient's denial of the fact that it has ever received the data [14]. Non-repudiation techniques may use hashing and digital signature schemes, which are introduced in the following subsections.

2.2.5.1 Hashing

A Hash or a message digest is a mathematical function that takes a random number with bounded size (limited by the minimum and maximum number of input bits) and creates a fixed-length message-digest output string with the following general specifications:

1. The output string is fixed length, independent of the input string length
2. A single bit difference between two data inputs results in close to 50% change in the hash's output strings
3. Given the hash value, it should be computationally impossible to find the initial data prior to hashing. This is based on the current accessible computational power within the time frame that the message content is useable.
4. A collision is a situation where two different input messages produce identical hash output message. The probability of finding such a collision, which depends on the type of hashing function and the number of output hash length, is generally very low. [23]

A hash value derived from the message m is denoted by $H(m)$.

2.2.5.1.1 Keyed versus unkeyed Hashes

There are two different types of hash functions; *keyed* and *unkeyed* [24]. Keyed hash functions (e.g., Message Authentication Code "MAC" schemes) accept a secret key as a secondary input, such as block cipher-based MAC (DES-CBC-MAC) and Hash-based MAC (HMAC) schemes.

An unkeyed hash function (e.g., Manipulation Detection Code "MDC") requires no secret key input message, such as: One Way Hash Function (OWHF) and Collision-Free Hash Function (CFHF).

A few hashing functions that are used in the construction of both keyed and unkeyed hash functions are: MD2, MD4, MD5, SHA-1, SHA-256, SHA-384, and SHA-512. Almost all Message Digest (MD) versions (e.g., MD2 to MD5) have proven to be weak [25, 26, 27], prone to internal collisions. An internal collision is a situation where a cryptographic algorithm computes two different input arguments, however returns the same output argument.

All hash functions are subject to the birthday attack. In a birthday attack, given a function $g(x)$, the aim of the attack is to find the probability that two different inputs; x_1, x_2 would yield: $g(x_1) = g(x_2)$. Then the pair; x_1 and x_2 is called a collision. In the cryptographic sense, the hashed value is known and in different scenarios, the message m, may or may not be known. In the worst case scenario, the message m is also given.

Assuming function $g(x)$ has an output space of N and N is large enough (larger than 10^9), for an unknown x_1 and known $g(x_1)$, it is expected to find x_2, which satisfies the equation $g(x_1) = g(x_2)$, after computing $g(x_i)$ for about $\sqrt{\frac{\pi}{2}N} \approx 1.25.\sqrt{N}$ different arguments on average.

Brute-force attack is another group of attacks, which features traversing the search of all possible keys space until the correct key is found. This is used to break the encrypted information by searching every key combination to decipher the ciphertext. On average, half of all possible keys should be tried before a match could be found, therefore the complexity of a brute-force attach scales exponentially with the increasing key size. The strengths associated to the brute-force attack lie in the simplicity of the attack algorithm and wide applicability. On the other side, the weaknesses associated to the brute-force attack is the inefficiency of the attack algorithm; that it requires a long period of time before a result is yielded and the attack process is relatively very slow [28].

2.2.5.1.2 SHA (Secure Hash Algorithm)

SHA was originally introduced in 1993 (FIPS PUB 180), initiated by the National Institute of Standards and Technology (NIST) and promulgated by the National Security Agency (NSA), which was called SHA-0. Later on, SHA-1 was introduced, which differs with SHA-0 in a single bitwise rotation in its compression function of the message schedule. Basically these two schemes take any arbitrary message length (of less than 2^{64} – 1 bits, less than 2 billion G bytes) and produce a 160 bit (fixed length) hash code [29].

Newer SHA schemes (SHA-2 family) were developed starting in 2001, including SHA-224, SHA-256, SHA-384, and SHA-512. The development of SHA-3 family is underway and the publication of the new standard will take place in 2012 [29].

SHA-0 is considered to be the weakest of all SHA family and it was announced broken in 2004. SHA-1 is considered secure till mid 2010 [29]. Table 2.1 shows the security portfolio of SHA-0 and SHA-1.

Table 2.1 Security portfolios of SHA-0 and SHA-1

	SHA-0	**SHA-1**
Secure or Vulnerable	Weak	Secure till 2010
Number of Operations	2^{39}	2^{63}
Collision Probability	2^{-43}	2^{-69}

2.2.5.2 Digital Signature

A digital signature is a mathematical expression intended to provide integrity and authenticity of the message and the person generating it. The generator of the message creates the digital signature and attaches it to the ongoing transmissions, which enables the recipient to correctly identify the real originator of the transmission and ensures that the message hasn't been altered by an illegitimate user. In some schemes, encryption would also be required to ensure that the information was not revealed to outsiders.

Before introducing a few digital signature schemes, it is necessary to discuss public and private key systems. Key exchange schemes are used when two parties (e.g., *Alice* and *Bob*) require exchanging keys exclusively and privately in an insecure environment. There are two main key exchange cryptographic schemes; *Symmetric and Asymmetric Key* systems. A symmetric key system (such as DES "Data Encryption Standard") is based on a simple encryption system that requires both sender and receiver to use a single key to encrypt and decrypt messages.

Therefore key exchange is an important step in symmetric key systems. One method to perform key exchange is through Diffie-Hellman (D-H) key exchange, which goes back to 1976 [30]. A D-H key exchange between *Alice* and *Bob* is shown in Figure 2.1, where *a* is Alice's secret key and *m, p* are public key information, where *p* is a very large prime number and *m* is a primitive root mod *p*. *Alice* calculates the value *D* based on *m, a,* and *p* and sends *D* to Bob. Bob takes *D* along with the public key information (*m* and *p*) and its own private key; *b,* and calculates *E* and sends *E* to *Alice.* Now both *Alice* and *Bob* can calculate *K* based on the values received from the other person. The values of *a* and *b* are chosen in a way that even if a third party intercepts both *D* and *E*, it is computationally infeasible to calculate *a, b, or K.*

An *asymmetric key* system, however uses two different keys for encryption/decryption. These keys are mathematically related to one another, forming a key pair; public/private keys. The private key should be kept private, while the public key can be available publicly. Thus a public key cryptosystem is an example of an asymmetric key system.

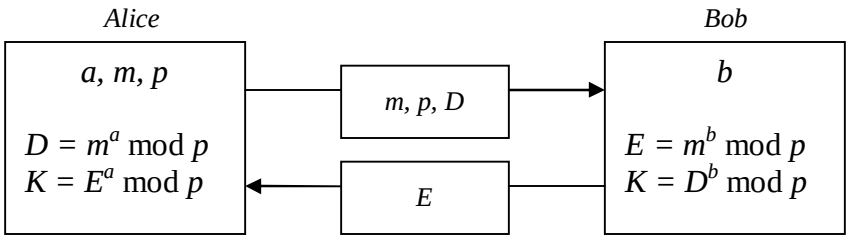

Figure 2.1 Diffie-Hellman key exchange protocol

A public key is typically used to encryption and a private key is used for decryption.

An example of public-key-based system is the Public Key Infrastructure (PKI), which binds public keys with user identities through a certificate authority (CA).

Symmetric keys are simpler compared to asymmetric cryptosystems, as they require just one key, however they require key exchange in a secure method through an insecure channel. A public key cryptosystem does not require key exchange, however it is more complex compared to a symmetric key system. Asymmetric key encryption is relatively slower than symmetric key encryption and therefore they are only used for key exchange and digital signature algorithms.

A digital signature usually consists of three steps: 1). Key generation (public/private key pair), 2). Signing operation, and 3). Verification.

There are several well-known digital signature schemes, namely; Digital Signature Algorithm (DSA) [31], RSA [32], and Elliptic Curve Digital Signature Algorithm (ECDSA), which is based on DSA's algorithm [33, 34].

Digital Signature Algorithm (DSA): The DSA in the Digital Signature Standard (DSS) was published in 1994 by the National Institute of Standards and Technology (NIST) and specified by Federal Information Processing Standard (FIPS) 186-3. DSA is based on discrete logarithm problem, which tries to find the exponent value (E) in the equation: $B^E = P \ (mod\ M)$, where B is the base, P is the power, and M is the modulus. The key generation, signing, and verification procedures of DSA are given in the Appendix A.

RSA: RSA cryptosystem is based on the assumption that factoring a very large number, which is the product of two very large prime numbers, and calculating those two prime numbers, is a very difficult task. The three stage (key distribution, signing, and verifying) are given in Appendix A.

DSA versus RSA: RSA systems can be used for both digital signature and encryption, whereas DSA systems are only used for digital signature. The DSA signing is faster than verifying, whereas in RSA, verifying is much faster than signing. The precise signing and verifying delays for both schemes depends on the codes used to implement these schemes, the platform (e.g., OS) running the algorithms, and the hardware (e.g., CPU speed, RAM) on which the tests are carried out.

Elliptic Curve Digital Signature Algorithm (ECDSA): The ECDSA is the elliptic curve-based version of the DSA scheme. It has been accepted in NIST and IEEE standards since 2000 [35, 36]. The Elliptic Curve Cryptography (ECC) utilizes the arithmetic operations and calculations of points, which are coordinates of an elliptic curve equation solutions defined over a finite field. The Elliptic Curve Digital Signature Algorithm (ECDSA) has been explained in Appendix A [36].

The bottom line is that elliptic curve cryptography involves shorter signature messages compared to DSA and RSA, for the same cryptographic strength. For instance, RSA 1024 bits and ECDSA 160 bits have the same cryptographic strength while the key length required for ECDSA is almost one seventh of the key length required for RSA [37].

2.2.6 Access Control

Access control is used to enable a legitimate user to have access to the resources. Access control uses one or more security mechanisms for granting access to the communication channel, application, and/or database. The following scenarios are categorized under access control: *User identification*, which is used to grant access to legitimate individuals. *Emergency access*, which is part of the access control scheme, where high-priority emergency access procedures (e.g., disaster relief) take precedence.

Data encryption-decryption, which are used for privacy and data integrity procedures to perform user- and application-dependent encryption/decryption schemes. *Automatic logoff/logon*, including shutting down parts of the network due to security breaches. Granting access permission to those parts is also within access control limits [14].

2.2.7 Availability

Availability is a probabilistic measure of resources being available for possible communication upon request. The higher the probability of resources being available, the lesser is the chance that resources become unavailable. Denial of Service (DoS) attack lessens the availability of the channel and resources, therefore DoS and availability oppose each other's effects [14].

2.2.8 MAC Layer Security Mechanisms

In this section we briefly consider a few security mechanisms built around MAC layer mostly deployed in Wi-Fi systems. These include *WEP (Wired Equivalent Privacy)* [38], *WPA (Wi-Fi Protected Access), PSK-TKIP (Pre-shared Key - Temporal Key Integrity Protocol)* [39], and *WPA-AES (Advanced Encryption Standard)* [40].

WEP has shown to have serious security flaws, including [41, 42]: keys reuse and weak Initialization Vector (IV).

2.3 Traffic Analysis – Packet-based Approach

Traffic analyses can be categorized in three major classes: packet-, flow-, and application-based analyses. In this section, packet-based traffic analysis is considered.

2.3.1 Packet Classifications Concept

Traffic classification techniques are used to categorize traffic flows into tangible selections. These selections may be categorized in two major forms; packet information (e.g., packet size, flow duration) and packet representing the application in use.

There are numerous categories of packet classifications, which are based on how they are observed and analyzed. An observation can be made at the packet level, focusing on; packet size, duration, burstiness, and transmission patterns. Another approach is the context and application-based packet approach, where the statistics of the packets are linked to the application, performance measures, and different underlying protocols stacks in use. These will be discussed in later sections.

2.3.1.1 Traffic Analysis in the Literature

In the literature covering traffic classifications, a classical method of identifying flows is based on various parameters, such as IP and port addresses. Reference [43] proposes a method in which the first five TCP packets are observation to identify the application in use. The proposed classification technique in reference [43] operates in two phases; an online traffic classification phase and an offline learning phase. The offline learning phase uses the training data, checking the TCP flows to extract common behaviors. The traffic classification phase is used to extract the applications running above the TCP layer.

Reference [44] discusses a new approach in traffic classification, which is called; BLINC; a multilevel traffic classification technique, which considers hosts flow activities instead of considering every individual TCP/UDP flows. The only limitation of BLINC is that it is capable of analyzing the statistics only after the connection is terminated. Therefore BLINC is incapable of analyzing the flows on the fly.

Reference [45] presents a framework for traffic classification, which operates in the presence of packet payload. The scheme utilizes several building blocks that are used to create sufficient confidence for application identity. This is done by collecting packets with payloads collected on the Internet backbone and sorted based on the TCP/UDP flows and their port numbers. The results show that a classification based on simple port numbers will provide approximately 70% accuracy for the traffic classification.

Reference [46] is based on NBAR (Network-Based Application Recognition), which is a traffic classification technique based on Internet applications (e.g., web-based), TCP/UDP port assignments, and other difficult-to-classify applications.

A few studies [47, 48] have shown that there are orthogonal correlations between four main traffic classification dimensions; *rate, duration, burstiness,* and *size*. These correlations are more accurate for heavy-hitters (e.g., long lasting connections), which contain DNS (Domain Name System) traffic.

Reference [49] presents App-ID, which is an application security-based classification scheme, which operates by establishing application sessions and identifying different traffic flows using one of the following approaches: Protocol and port numbers, SSL (Secure Socket Layer) decryption, application decoders, or application signatures.

In the study of traffic classifications, Peer-to-Peer (P2P) networks are also important to consider where either TCP or UDP is used on top of IPv4 to convey file sharing data between individual users [50, 51, 52, 53]. P2P applications have gained popularity over the past few years and their usage is and will be on the rise. Reference [50] emphasizes on two main issues; the P2P traffic patterns and the fact that P2P applications use non-standard and random port numbers, and the fact that conventional flow classification techniques are not adequate for proper classifications. Reference [51] demonstrates the accuracy, feasibility, and robustness of high speed P2P signature-based traffic applications. It discusses a number of P2P application protocols, such as eDonkey, BitTorrent, DirectConnet, Gnutella, and Kazaa protocols. The measurements show that by using application-level signatures techniques, less than 5% false position/negative ratios may be encountered.

A few studies [52, 53] offer comparative approaches towards studying P2P traffic behaviors. Reference [53] offers three such approaches for P2P application classifications; *port-based, application-layer signature,* and *transport-layer longitudinal approaches* using empirical network traces over a two-year period. The results show that a classic port-based analysis is not accurate, which is inline with the results achieved in reference [50]. Application-layer signature approach, on the other hand, yields more accurate results, inline with the results achieved in reference [51].

Reference [54] uses Naïve Bayesian estimator for Internet traffic classification analysis. With fine-tuning of the estimator's variants, the results show 65% accuracy for a per-flow traffic classification technique. The accuracy can be increased to 95% when data from the same period is further analyzed using additional tools, such as Bayes based kernel-estimator combined with FCBF (Fast Correlation-Based Filter).

Reference [55] uses a supervised Naïve Bayesian estimator algorithm, which features building statistical models, describing the classes based on training data (machine learned classification). The results show an accuracy of 83% and higher for both per-byte and per-packet classifications.

Reference [56] provides an accuracy of 82-100% based on an empirical evaluation technique, which models both host-specific- and aggregate-protocol behaviors. Such an accurate classification is independent of port label, which is not in-line with the traditional classification methods. The aggregate models are able to classify an unclassified server flow and determine whether flows this server match the behavior of previously seen flows from that server or not. The significance of these classifiers is the ability to augment traditional intrusion detection systems and detect artifacts of successful attacks.

The performance of a traffic classifier directly depends on consideration of various traffic attributes. An example of such traffic attributes is flow size and failing to consider it effectively will contribute to the reduction of traffic classification accuracy [57]. Other attributes include; QoS measures and identifiers, which require CoS (Class of Service)-based classifications [58].

Some protocols have certain attributes, which are suitable for traffic classification purposes. One set of protocols that are often noticed on the Internet backbone are routing protocols. Two different types of routing protocols are; internetwork and intranetwork routing protocols. Internetwork (e.g., Internet Autonomic System "AS") routing schemes operate on larger scales, such as BGP (Border Gateway Protocol), whereas interanetwork

routing schemes work inside one network's boundaries, such as OSPF (Open Shortest Path First). It is obvious that only internetworking routing schemes are observed on the Internet backbone. Flow classifications based on classifying BGP level prefix flows are one example of routing traffic classifications [59, 60]. Reference [59] uses a method based on Dirichlet Mixture Processes, which models flow histograms with a capability of examining macroscopic flows while distinguishing between various classes of traffic.

An empirical approach to Inter-AS traffic classification [60, 61] includes extensive Internet-wide measurements and characterizing, classifying, and ranking them into individual ASs based on the utilities they derive (e.g., residential and business). The related scatterplots show that there are correlations between various pairs of utilities.

Machine Learning (ML) methods have also been widely used in traffic classification techniques [62, 63], where traffic clusters are created based of various traffic characteristic. Early ML techniques mostly relied on offline and static analyses of traffic batch traces. However recent work is mostly based on real-time ML-based IP traffic classifications.

Traffic classifications with various security measures in mind have been considered in various literatures [64, 65, 66]. It is shown that it is possible to classify and categorize Internet traffic flows without proper content analysis [54]. Using statistical signatures, it is possible to classify services even when they are running on non-conventional port numbers [65]. Reference [66] argues that the application of SSL (Secure Socket Layer) is on the rise and the characterization of SSL, which recognizes applications running on encrypted SSL connections based on the first packet size, provides an accurate traffic classification technique with more than 85% accuracy.

Many of the parameters used in the study of traffic classifications, exist at the network layer. Therefore several studies [67, 68] included more focus on the Internet Protocol (IP), which operates at the network layer in the TCP/IP suite.

More information on traffic classification is given in Appendix A.

2.3.2 Wireless QoS Requirements in the Absence of Security Mechanisms

QoS in general falls into two categories; user perspective and network perspective. User perspective QoS refers to the quality as perceived by the user, including multimedia (e.g., video, audio, streaming, text, and file transfer) subjective quality. Other user perspective parameters in regards to QoS include:

Connection Drop – When the delay or jitter figures increase passed certain limits, the link quality either becomes unbearable to the user or the underlying application drops the connection, causing a link failure. In either case, it will negatively affect the user greatly.

Depending on the application in use (e.g.., audio, video, voice messaging, and audio streaming), the requirements for the subjective QoS (user perception) figures may change. For instance, if the end-to-end audio delay becomes more than 150 msec, the user level of discomfort starts to increase dramatically.

In regards to network perspective QoS, disregarding security, the QoS-related parameters for multimedia applications include: Bandwidth or Throughput, Round-Trip Time (RTT), End-to-End Delay (E2ED), Bite error rate (BER), Packet Loss Ratio (PLR), Packet drop ratio (PDR), and Jitter [69, 70, 71, 72, 73]. A few of these parameters were introduced earlier in this chapter and the rest are defined as followed:

Bit Error Rate – BER is the ratio of the number of error bits to the total length of information messages bits; such as sending 1 and receiving 0 at the receiver or vice versa. Channel conditions contribute to the value of BER, so when noise and/or interference levels rise, BER values rise as well.

Packet Loss Ratio – PLR is a parameter that represents the ratio of the number of lost packets to the total number of packets sent. The performances of the link and the intermediate nodes have direct impacts on the value of the PLR. The higher the PLR value, the less efficient the communication path is between the source and the receiver.

Packet Drop Ratio – PDR is a performance measure that is mostly affected by the receiver's input buffer. When the input buffer starts to fill up, a mechanism starts discarding (dropping) the packets. The lower the PDR value, the better is the quality of these buffers.

2.3.2.1 Bandwidth Requirements

Depending on the multimedia application in use, bandwidth constraints are different. Table A.1 (adapted from [74, 75, 76], Appendix A) shows bandwidth requirements for various MPEG formats (combination of video and audio).

2.3.2.2 Voice over IP (VoIP) Bandwidth Requirements

Voice over IP is an important multimedia application, which has become a dominant engine of transporting voice across IP networks [77].

VoIP systems deploy specific codecs to packetize voice messages. Each of these codecs has specific characteristics with unique bandwidth and delay requirements. The bandwidth requirements of a number of codecs are mentioned in Table A.2 (adapted from [78, 79, 80, 81]). The qualities of these codecs have direct effects on both user-perception (voice/video qualities), as well as network perspective QoS (e.g., overall delays).

2.3.2.3 End-to-End Delay

In a VoIP system, the transmission of voice data packets is not instantaneous and latency is the term used to describe the time durations that a voice packet is packetized, encoded, moved across the network to an endpoint, decoded and de-packetized, and de-jittered, at the receiving end.

As mentioned, the end-to-end delay has to be minimized for real-time and interactive applications. End-to-end delay reduction improves throughput figures directly. A thorough end-to-end delay analysis is needed for precise throughput calculations.

Total latency is so-called; end-to-end latency, mouth-to-ear latency, round-trip-delay (RTD), or round-trip time (RTT) [78].

In VoIP, real conversations usually involve “turn-taking” with 200 msec breaks. When the latency of a network approaches the 200 msec limit, the conversation flow becomes distorted. The two end parties may interrupt each other by starting to talk simultaneously or remain silent at the same time. Higher delays passed the 150 msec (300 msec two-ways) limit will affect the quality greatly [82]. Video codecs also have delay limits, for instance H.261 and H.263 are typically within the 200 msec to 400 msec limits.

Multimedia applications often require bounded delay figures to offer seamless QoS. An end-to-end delay is comprised of the following delay figure combinations: Packet loss compensation, packet processing delay (comprised of; codec, serialization, queuing, and propagation delays), and dealing with the network jitter.

Codec delay is the combination of frame processing and lookahead delays, which are defined as followed:

- Frame processing delay is a delay of processing a single voice data frame.
- Lookahead delay is the next frame processing delay, which is needed for algorithms with correlation schemes (e.g., ADPC)

Table 2.2 Audio codec delays

Codec	Look Ahead Delay	Codec Delay
G.711	0	0.25 msec
G.722	0.125	1.25 msec
G.722.1	20	20 msec
G.723.1	7.5	67.5 msec
G.726	0	0.25 msec
G.728	0	1.25 msec
G.729	5	25 msec
G.729a	5	25 msec
GSM-HR	4.4	44.4 msec
GSM-EFR	0	40 msec
GSM-FR	0	40 msec

Table 2.2 (adapted from [83]) shows a few audio codecs delay figures. The processor for which the codecs are tested with are based on Packetcable audio and video Multimedia Terminals (MTAs) and Trunking Gateways (Media Gateway).

The rest of the delays are from: BER, PLR, PDR, PRDeR, echo, and Jitter. Jitter is one of the most important phenomena affecting the quality of a VoIP system.

Jitter happens due to the fact that there is no delivery guarantees for the voice packets across IP networks, therefore there are possibilities that not all voice data packets travel the same path, causing variation in the packet arrival times. This may happen because some packets may be forced to travel paths with more hops than other packets, depending on the routing decisions. Therefore packets arrive at the destination node with variable delays causing much higher latency effect, called *jitter*, which is calculated per seconds [84, 85, 86].

Too many packets being processed at the intermediate gateways/routers may overwhelm the processing duties momentarily. Both of these circumstances cause latency to become irregular and this irregular delay, as mentioned, is called jitter. To lessen the jitter's effect, packets are gathered in jitter buffers in the intermediate transmission devices and at the receiving-end device. Table 2.3 (adapted from [87]) shows the acceptable VoIP jitter figures, which should be below the 70 msec level and for inter-frame delay (frame delay jitter) in video, it should be less than 150 msec (for H.261).

Table 2.3 Jitter figures in VoIP (Cisco-based) systems

Jitter	**Quality**
Less than 40 ms	Excellent *(unnoticeable jitter)*
40-75 ms	Acceptable *(noticeable jitter)*
Larger than 75 ms	Unacceptable

Table 2.4 Comparison of R-values and MOS scores

Characterization	MOS Value
Very Satisfied	4.3+
Satisfied	4.0-4.3
Some Users Dissatisfied	3.6-4.0
Many Users Dissatisfied	3.1-3.6
Nearly All Users Dissatisfied	2.6-3.1
Not Recommended	1.0-2.6

Table 2.5 Upper limits of codec's MOS values

Codecs	Mean Opinion Score (MOS)
G.711 (64 kbps)	4.3
G.722 (64 kbps)	4.2
iLBC (13.33/15.2 kbps)	4.14
AMR (12.2 kbps)	4.14
G.729 (8 kbps)	3.92
G.728 (16 kbps)	3.9
G.723.1 (6.3 kbps)	3.9
GSM EFR (12.2 kbps)	3.8
G.726 ADPCM (32 kbps)	3.8
G.729a (8 kbps)	3.7
G.723.1 (5.3 kbps)	3.65
GSM FR (12.2 kbps)	3.5

The combination of end-to-end delay, jitter, noise-levels, and other factors, are used to calculate a subjective measure for VoIP system, which is called; the Mean Opinion Score (MOS) value. MOS values vary from 5 (highest) to 1 (lowest).

The quality of the audio codec has a direct impact on the MOS value (Table 2.4). Table 2.5 (adapted from [88]) shows a few codecs and their upper MOS limits.

In general, multimedia traffic requires bounded limits on the QoS-related parameters (e.g., delay) to guarantee certain performance measures. Table 2.6 (adapted from [89]) summarizes general QoS requirements for multimedia traffic.

Table 2.6 General QoS requirements for multimedia traffic

Multimedia Format	Application	Flow Direction	Nominal Date Rates (kbps)	One-Way Delay (msec)	Jitter (msec)	PLR
Audio	Voice Conversation	Duplex	4-64	< 150	< 70	< 3%
Audio	Voice Messaging	Half Duplex	4-32	< 1	< 70	< 3%
Audio	High Quality Audio Streaming	Simplex	16-128	< 10	< 5	< 1%
Video	Video Phoning	Duplex	16-384	150-400	< 30	< 1%
Video	Video Streaming	Simplex	16-384	< 10	< 30	< 1%
Data	Web Browsing (HTTP)	Duplex	10	< 100	N/A	0%
Data	Transaction (SSL)	Half Duplex	< 10	< 100	N/A	0%
Data	Control	Duplex	1	< 250	N/A	0%
Data	Interactive Gaming	Duplex	<1	< 200	N/A	0%
Data	Telnet	Half Duplex	< 1	< 200	N/A	0%
Data	Email (Server Access)	Half Duplex	< 10	< 100	N/A	0%
Data	Bulk Data Transmission	Simplex	10 – 10,000	< 60k	N/A	0%
Data	Still Image	Simplex	< 100	< 60k	N/A	0%
Data	Email (Server-to-Sever)	Half Duplex	< 10	< 60k	N/A	0%
Data	Fax (Real Time)	Half Duplex	10	< 700	N/A	< 1%
Data	Fax (Batch)	Half Duplex	10	< 10k	N/A	< 1%
Data	Background Priority	Half Duplex	< 10	< 30k	N/A	0%
Data	Usenet	Simplex	> 10,000	< 10k	N/A	0%

2.3.3 Security

In the previous section, QoS parameters were discussed without considering security mechanisms. In this section, requirements are considered in the presence of various security schemes.

The main security mechanisms are [14]: *Confidentiality (data, address), integrity, authentication, authorization, non-repudiation, access control,* and *availability*.

Each of the above mentioned mechanisms will have effects on the latency, bandwidth, and processing power. In this section we will study these effects.

2.3.3.1 Confidentiality (Privacy)

Privacy is needed to protect the communication, both the communicating parties and the transmitted information among them. To keep the communication private, encryption schemes are used and for the privacy of the communicating parties, VPNs (Virtual Private Network), such as IPSec (IP Security), are used. VNPs are used to create secure tunnels from the source to the destination, in which they hide the real IP addresses of the parties and encrypt the payload transmitted between end-points.

Such a deployment (VPN) has an overhead. In a non-interactive environment, such an overhead may not be critical. However for an interactive communication, such an overhead becomes critical [90, 91]. To carry-out real-time application, RTSP (Real-Time Streaming Protocol) is used which uses RTP (Real-Time Protocol) on top of UDP or TCP.

RTP has a relatively large header with a minimum 12 byte length and is usually accompanied with IP (IP header of 20 bytes) and UDP (UDP header of 8 bytes), resulting a 40 byte header packet. This header could be extended by 20 to 800 bytes, depending on the codec in use.

There are various VPN technologies, including [92]: PPTP (Point-to-Point Tunneling Protocol), L2F (Layer-2 Forwarding), L2TP (Layer-2 Tunneling Protocol), IPSec, GRE (Generic Routing Encapsulation), MPLS (Multiprotocol Label Switching), ATM (Asynchronous Transfer Mode), and Frame Relay. ATM and Frame Relay are Layer II VPNs and GRE, MPLS and IPsec are layer III VPNs.

Depending on the platform and implementation specifics, IPSec's may have varying impacts on the throughput figure for both FTP and HTTP downloads in the absence and presence of other security schemes [93, 94].

2.3.3.2 Integrity

Integrity is used to ensure the receiving partner that the data received has not been altered by any means after it has been transmitted from the sender. Integrity and non-repudiation mechanisms are often provided at the same time. Therefore hashing schemes are used to provide integrity and are used in the mechanisms of digital signatures to offer non-repudiation capability. A few hash schemes are: SHA-1, -256, -384, and -512 [95], Message Authentication Code (MAC), and HMAC (Hash-based Message Authentication Code).

2.3.3.3 Authentication

User authentication is a process by which, identity of an entity is confirmed. Data authentication confirms what data belongs to which user, which is provided by certificates and digital signatures. Once a user has been authenticated with a server, the user remains authenticated as long as the session is ongoing. Once the session is terminated and reestablished, authentication may be required again, however there are a few mechanisms (e.g., key caching [96]) that may exempt an entity from requiring the authentication process after being authenticated at least once and session termination. A strong authentication mechanism involves two-way authentication handshaking to validate both client and server (or two clients in a peer-to-peer scenario) to one another. A few authentication mechanisms include: IEEE 802.1X (both wired and wireless), EAP

(EAP-TLS, EAP-TTLS, LEAP, PEAP and EAP-FAST, EAP-SIM), RADIUS, public key (X.509) [97, 98]. Depending on the authentication protocol, it can cause a delay between 50 msec to 400 msec.

2.3.3.4 Non-Repudiation

Non-repudiation is required so that none of the involved parties can later on deny their involvements in a communication, which has already taken place among them. Digital signature was introduced in section 2.2.5.2 Reference [99] discusses a few digital signature schemes and compares the performance of ECDSA and RSA. In the design of a non-repudiation system, it is vital to take the end-to-end (key distribution, signing, and verifying) delay figures into account. The end-to-end delay of a digital signature scheme has an impact on the system's throughput, which should also be considered.

2.3.4 IEEE 802.11i

IEEE 802.11i is a security amendment for IEEE 802.11 standard with WPA (Wireless Protected Access) as its subset. IEEE 802.11i specifies a four-way handshaking for the encryption key distribution and includes a combination of privacy (encryption), authentication, integrity, and confidentiality. The standard specifies WPA2 (RSN-AES) on top of EAP using CCMP [100].

Each of the security mechanisms contributes to the delay and overhead figures, which reduce the performance. Authentication has an initial impact when client/server enter the authentication phase. Once authenticated, no more authentication procedure is required until device roams. Encryption and digital signature may impact all the involved packets. The impact of security has been investigated in various references [101, 102, 1103, 104].

2.4 Wireless Traffic

Wireless traffic differs from wired traffic in variety of ways. First of all wired data transmitted over a LAN is band/channel free. Therefore a wired packet analyzer is able to

monitor all packets simultaneously. However, wireless traffic is band/channel dependent. In general, for current Wi-Fi networks, there are two functional bands; radio *b/g* (2.4 GHz) and radio *a* (5 GHz). Each radio has specific number of functional channels.

2.4.1 IEEE 802.11 b/g

IEEE 802.11 *b/*g radio operates at 2.4 GHz and the channels associated to these two radios are regional dependent. These regional dependent channel schemes are: North America: 11, Europe: 13, and Japan: 14 channels. The detailed regional schemes are depicted in Table A.3 [105, 106].

Each 802.11*b*/*g* channel is 5 MHz apart from the neighboring channel and in order for two neighboring APs to be able to operate simultaneously without any RF interference, they are bound to transmit on two channels with at least 5 channel (25 MHz) separation. Therefore it's a universally accepted scheme to label channel 1, 6, and 11 as the *most frequently used* or *main channels*. Any neighboring APs operating on these non-overlapping channels can operate simultaneously. The rest of the channels are labeled as *less frequently used* or *sub-channels*.

Due to the fact that in many geographical regions, main Wi-Fi channels are overloaded, some mobile carriers have started deploying sub-channels, such as; channel 5, 7 and 10 on their Wi-Fi capable mobile devices.

2.4.2 IEEE 802.11 a

Each Access Point (AP) advertises its presence and capabilities through a short message it transmits, which is called a *beacon*. Beacons are usually 102.4 msec apart from one another. In a beacon one can find the wireless MAC address, radio and channel assignments, extended offering rates, SSID name, QoS and security capabilities and etc..

Radio *a* operates in the 5 GHz frequency band and is also regional dependent. Due to the wider frequency spectrum band compared to radio b/g, more channels are allocated to

this radio compared to radio *b*/*g*. Since each radio *a* channel is already 20 MHz apart from the neighboring channel, there is no restriction on the neighboring APs in regards to the channel usage. If two neighboring APs use even neighboring channels (e.g., 36 and 40), they will still be able to operate without causing any interference on one another. Therefore there is no main-channel/sub-channel issue in radio *a*. However some channels are used more often. For example in North America, 9 channels out of 12 available channels are used more often. Table A.4 [105, 106, 107, 108, 109] shows the regional frequency allocations in IEEE 802.11*a*.

Besides physical differences between wired and wireless traffic (e.g., channel, band, etc.), most of the attributes between wired and wireless traffic classification are very similar.

2.5 Multilayer Wireless QoS/Security Provisioning

2.5.1 Introduction

Security and QoS provisioning and their co-existence are important aspects in the design of new protocol stacks. Such provisioning will become more critical for wireless systems as there are more limitations associated to wireless systems compared to wired systems.

Security and QoS provisioning has traditionally been considered at lower OSI layers (e.g., MAC). However there is an increasing interest in provisioning shift from lower OSI layers to higher layers; ultimately to the application layer. It is believed that such a transition will add user-awareness and efficiency of the network with more adaptability to the current demands.

This section will include discussions on QoS and security requirements at selected layers and in multilayer and cross-layer contexts and eventually we will study various criteria for considering such provisioning at the applications layer.

2.5.2 QoS/Security Parameters at the Physical (PHY) Layer

PHY layer deals with the actual transmission of bits and bytes on the physical medium (e.g., wire, air, etc.). The physical limitations of the medium usually prevent much maneuver in terms of applicable QoS and security schemes at the PHY layer. Therefore in most communication systems, as the information packets travel from the application layer towards the lower layer, most of the QoS and security mechanisms are being applied in higher layers before the information packets reach the PHY layer. The followings are the parameters related to QoS/Security at the PHY layer:

2.5.2.1 QoS at the PHY layer

At the PHY layer of wireless links, QoS-related parameters mostly deal with packet loss ratio rates and throughput rates, which are functions of the channel quality. One of the attributes of wireless channels is the continuously varying channel quality. The existence of this attribute makes it almost impossible to maintain constant date rate and low packet loss rate. The parameters that deal with QoS at the PHY layer include: Packet Drop Ratio (PDR) [110], Bit Error Rate (BER) [111], Signal to Interference Ratio (SIR) [112], transmission power, and built-in strengths for coding and modulation techniques (e.g., Orthogonal Frequency Division Multiplexing “OFDM”, Direct Sequence Spread Spectrum “DSSS”, Frequency Hopping Spread Spectrum “FHSS”, Infrared, etc.).

2.5.2.2 Security at the PHY layer

The security strength related to the PHY layer comes from its interactions with higher layers where security has been considered and the limited security features incorporated in PHY layer itself. The most prominent security threat in PHY layer is the Denial of Service (DoS) attack, where the shared physical medium (air) is used by an attacker to launch threats to the physical resources. DoS in its simplest form can be in the form of a radio jamming that causes the legitimate users to experience limited (or no) service from the radio transmitter. This is done when the attacker is located between users and the transmitter and is sending interfering and jamming signals [14]. Another PHY layer DoS

issue is tampering, where the attacker has physical access to the infrastructure. DoS is a multilayer security issue and we will be considering it in other layers as well.

Therefore the security measures applied to the PHY layer is a guarantee of low-probability-of-interception (LPI) [113], which is dependent on several transmission properties, including; modulation schemes, channel, and signal characteristics.

2.5.3 QoS/Security parameters at the Data Link Layer

In legacy wireless protocols, data link layer is the most important layer in terms of security and QoS realizations.

2.5.3.1 QoS at the MAC (Medium Access) Layer

In many current and legacy protocols, MAC layer plays major roles in terms of providing QoS. The following protocols provide wired-based QoS at the MAC layer:

IEEE 802.1p – This is the standard providing 8 levels (3-bits) of user traffic classes.

IEEE 802.1Q – This is the bases for VLAN (Virtual LAN) technology, where switching is done at the MAC layer. IEEE 802.1Q frame has the following fields: 16-bit Tag Protocol Identifier (TPID), 3-bit Priority Code Point (PCP), which is also known as IEEE 802.1p, one bit Canonical Format Indicator (CFI), and a 12-bit VLAN Identifier (VID).

2.5.3.1.1 QoS for Wi-Fi (EDCA Mechanism)

IEEE 802.11e is an IEEE 802.11 amendment offering variety of enhancements including queue-based QoS functionality. Two new mechanisms were included in IEEE 802.11e; EDCA (Enhanced Distributed Channel Access), which is based on DCF systems and HCCA (HCF Controlled Channel Access), which is based on HCF (Hybrid Coordination Function) for PCF mechanism. EDCF (Enhanced Distributed Coordination Function) combines the advantages of both DCF and PCF while operating in both contention free and contention periods. It provides guarantee of service with a much higher probability in respect to EDCA. Figure 2.2 (adapted from [114]) shows the

EDCA/HCCA block diagram in the IEEE 802.11e standard. WMM (Wi-Fi Multimedia) is another name for EDCF and HCCA is also known as WMM Scheduled Access.

There are eight different User Priority (UP) schemes associated to the EDCA mechanism (Table 2.7 (adapted from [127]), including Background (BK), Best Effort (BE), Video (VI), and Voice (VO). In order to comply with the eight UP values in Ethernet frames (IEEE 802.1D) and Class Selector (CS) values in DSCP frame, these four QoS queuing schemes are repeated twice each, therefore the UP values are: 000 (BK), 001 (BK), 010 (BE), 011 (BE), 100 (VI), 101 (VI), 110 (VO), 111 (VO). Figure 2.3 (adapted from [114]) also shows the IEEE 802.11e's queuing systems.

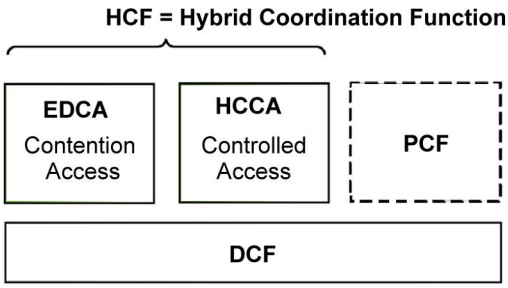

Figure 2.2 IEEE 802.11e QoS system

Table 2.7 User Priority schemes in 802.11e

User Priority	User Priority	IEEE 802.1D Designation	Access Category	Designation
Lowest	1	Background (BK)	AC_BK	Background
	2	-	AC_BK	Background
	0	Best Effort (BE)	AC_BE	Best Effort
	3	Excellent Effort (EE)	AC_BE	Best Effort
	4	Controlled Load (CL)	AC_VI	Video
	5	Video (VI)	AC_VI	Video
	6	Voice (VO)	AC_VO	Voice
Highest	7	Network Control (NC)	AC_VO	Voice

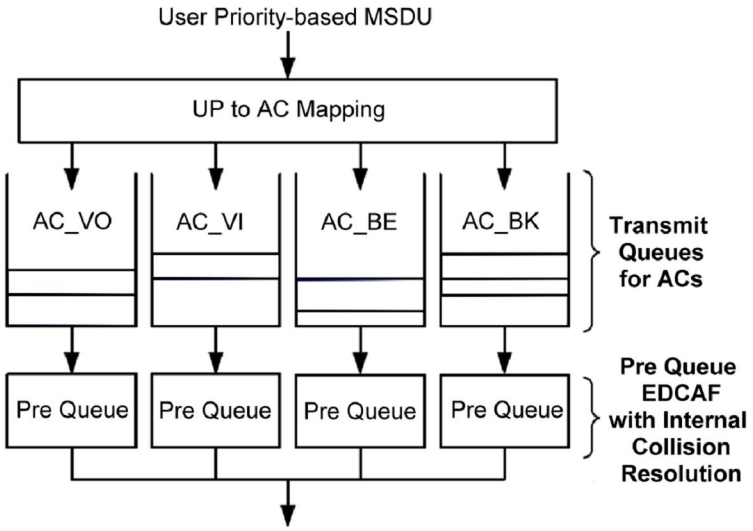

Figure 2.3 Queue system in 802.11e QoS system

2.5.3.1.2 QoS Scheme for IEEE 802.16 (WiMAX) Systems

WIMAX systems follow the same principles as in Wi-Fi systems. For instance, as mentioned, IEEE 802.11e follows a four level QoS queuing system, whereas in IEEE 802.16, there is a five level queuing mechanism offering QoS classes of services, namely; `UGS` (Unsolicited Grant Service), `rtPS` (Real-Time Polling Service), `ertPS` (Extended Real-Time Polling Service), `nrtPS` (non-Real-Time Polling Service), and `BE` (Best Effort) [115].

2.5.3.2 Security at the Data Link Layer

Many current and legacy security/QoS systems have QoS and security mechanisms built-in to their MAC layers. Therefore a thorough observation of this layer is required.

For each new access technology (e.g., Wi-Fi, WiMAX, etc.), MAC layer stack is updated, however LLC remains relatively unchanged. This is due to the fact that the interaction of upper layers (e.g., network) and the DLL should remain the same and independent of the underlying MAC/PHY technology in use. That is why the term MAC is used more often than DLL and LLC.

2.5.3.2.1 Logical Link Control (LLC)

The only functional entities operating under LLC are: Service Access Points (SAPs) and flow control with minimal variations between different access technologies. Therefore there are no significant security/QoS-related parameters under LLC.

2.5.3.2.2 Medium Access Control (MAC)

Protocol stack variations between different access technologies are best visible under MAC layer, where most of the current and legacy mechanisms lie. For Wi-Fi systems, the two available security protocols include; WEP and WPA.

WEP (Wired Equivalent Privacy) – WEP was the initial scheme intended to bring access control, privacy and confidentiality for IEEE 802.11 legacy systems (Figure 2.4). WEP uses RC4 stream cipher and relies on a secret key being shared between the client and the server [116].

There are number of security flaws associated to WEP, which makes it vulnerable to various attacks. Shared key authentication requires that a client uses a preshared WEP key for encrypting the challenge text transmitted from the AP to the client. The client is then authenticated by the AP by decrypting the shared key response and checking that it matches the challenge text.

The process by which the challenge text is exchanged, takes place over the wireless link. An eavesdropper is able to capture both the plain-text challenge text and the associated cipher-text response. WEP encrypts using an exclusive OR (XOR) function on the plain-text with the key stream, which produces the cipher-text. Therefore, an eavesdropper can use a protocol analyzer and easily derive the key stream by sniffing the shared key during the authentication process. Therefore the shared key authentication is vulnerable to a man-in-the-middle attack.

The other vulnerability in WEP comes from the weakness in the IV (Initialization Vector) mechanism, which is 12 bits (for both 64 and 128 bits WEP schemes). This weakness involves the IV repetition in various WEP packets. Therefore the secret key can be found by analyzing the traffic patterns [116].

IEEE 802.11i and WPA (Wi-Fi Protected Access) - IEEE 802.11i is the security amendment of IEEE 802.11 following the official acceptance of WEP's weakness in 2004. The Wi-Fi Alliance certification programs for IEEE 802.11i are called WPA.

The 802.11i architecture is built around the IEEE 802.1X authentication scheme, which involves the deployment of EAP (Extensible Authentication Protocol), Authenticator (e.g., Access Point "AP"), and the Authentication Server, RSN (Robust Security Network), and either TKIP (WPA) for authentication (entailing the use of EAP and an authentication server), RSN (Robust Security Network), which keeps track of associations, and AES (Advanced Encryption Standard)-based CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol, bases of WPA2), which provides integrity, confidentiality, and origin authentication. IEEE 802.11i authentication process involves a four-way handshaking scheme, which takes place between the wireless client station (STA) and the Access Point (AP) [117].

There are two types of keys used in this handshaking scheme, the transient keys (Pairwise Transient Key "PTK") and the permanent keys (Pairwise Master Key "PMK"). In this process PMK is first generated and then a PTK is created by the concatenation of the following information: 1). PMK, 2). STA nonce (`SNonce`), 3). STA Wireless MAC address 4). AP nonce (`ANonce`), and 5). AP Wireless MAC address. Following the creation of PMK and PTK, a GTK (Group Temporal Key) is generated, which is used for broadcast and multicast decryption [117, 118].

2.5.4 QoS/Security Parameters at the Network Layer

Network layer mostly deal with efficient routing of packets all the way from the source network to the destination network. There are network-layer schemes offering both QoS and security capabilities, which will be discussed in this section.

2.5.4.1 QoS at the Network Layer

Integrated Services (IntServ) is one of the earliest QoS schemes, where end-to-end QoS solution is provided using Resource Reservation Protocol (RSVP)-flows. In this case all nodes between the source and the receiver had to agree on QoS specifics. This scheme lacked in scalability. Networks became very complicated when nodes were increased.

Internet Protocol (IP) (both IPv4 and IPv6), which operates at the network layer defines an eight bit QoS related field; ToS (Type of Service). Differentiated Services (DiffServ) later on adopted a scheme based on the ToS field in which it renamed the field to DSCP (Differentiated Services Code Point). This way, when receiving packets, DiffServ-enabled nodes can act independently from the other nodes, which is called Per-Hop Behavior (PHB), which is purely based on individual DSCP values given to every flow passing through each nodes. Therefore the DiffServ scheme solves the scalability issue of IntServ.

2.5.4.2 Security at the Network Layer

There are various security schemes and mechanisms operating at the network layer, which will be discussed in this section.

2.5.4.2.1 IP Security (IPSec)

IPSec (all versions) is a well known network layer security scheme, which possesses versatile applications in both consumer and professional markets [119].

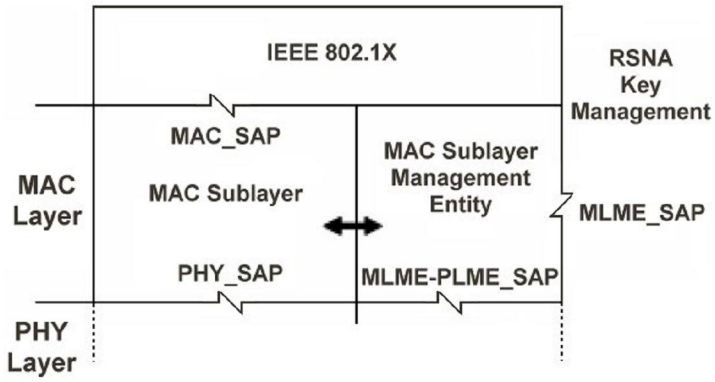

Figure 2.4 IEEE 802.11 security protocol stack

IPSec's mechanism works around the idea of using Security Associations (SAs) and cryptographic encryption/decryption keys (e.g., DES, 3DES, or AES) [119]. IPSec uses IKE (Internet Key Exchange) mechanism to manage (establish and change) keys, where IKE is a public key-based scheme. IPSec may use Certificate Authorities (CAs) or shared-secrets for the authentication purposes. IPSec uses Diffie-Hellman scheme for the security key exchange scheme. IPSec is able to detect and prevent message alteration using message hashing algorithms (e.g., ESP-SHA-HMAC).

There are two flavors of IPSec mechanism; Authentication Header (AH) and Encapsulating Security Payload (ESP). An AH scheme provides data authentication without encryption. Therefore AH-IPSec messages can be viewable by a third party.

An ESP scheme also comes in two flavors; Tunnel mode and Transport mode. In tunnel mode, the IP header (e.g., source, destination, etc.) is encrypted in addition to what AH provides. Therefore in tunnel mode, headers require frequent decryptions/re-encryptions passing through each node and hub for routing purposes.

In transport mode, the payload is only encrypted and the IP header is left unencrypted. Therefore in transport mode, normal routing procedure is in place and no extra operations are required, therefore transport mode is used for host-to-host communications.

2.5.5 QoS/Security Parameters at the Transport Layer

Transport layer is responsible for the correct delivery of data all the way from the source to the destination. However most of the traffic flows on the Internet backbones are based on TCP (Transmission Control Protocol, RFC 793, created in 1981) and UDP (User Datagram Protocol, RFC 768, created in 1980) and in time of their creation, QoS, security, and wireless communications were not considered. Therefore in order to provide transport layer mechanisms offering QoS, security, and wireless capabilities, enhancements are required, which were not included in the legacy transport layer architecture, however added to the newer protocol versions, such as NSIS (Next Step In Signaling), AOTP (Application Oriented Transport Protocol), and SCTP (Stream Control Transmission Protocol), which are enhanced with QoS provisioning capabilities and Transport Layer Security (TLS) [120], which provides security at the transport layer.

2.5.5.1 TCP versus UDP

TCP is a connection-oriented protocol that offers the following features: Reassembly of the arrived data in order (though that they might have been received our-of-order), correctness of the received data, detection and correction of lost and duplication of the received data, and traffic congestion control. For these, TCP has a complex frame structure, featuring source/destination port, sequence, and acknowledgement numbers, window, control, option, payload, and other fields. TCP also features a three-way handshaking scheme to provide guarantee of delivery.

TCP is used for important content delivery and the delivery of data with very low tolerance for lost packets. When TCP is used, guarantee of delivery is favored over transmission speed.

UDP, on the other hand, is a connectionless protocol with a less complex frame structure, including source/destination port numbers, length, checksum, and payload fields. Applications running over UDP may be tolerant to some degree of packet loss and the speed of data delivery is the most important factor for these applications. Audio/video streaming is an example of an application that uses UDP as its transport protocol.

2.5.5.1.1 Challenges in deploying TCP and UDP

For applications running over TCP, delay and speed are limited due to the complexity of the inbuilt mechanisms in TCP for the delivery guarantee. Therefore if speed, low-delay figures, and guarantee of delivery are all required for an application, this may be a challenging task in the deployment of TCP.

In regards to the deployment of UDP, as mentioned, speed of delivery and lower delay figures are provided, however there is no guarantee of delivery. Therefore in a busy network with numerous stations communicating simultaneously, chances will increase for packet loss and incurring congestions periods, which may become unacceptable. In such scenarios, there is no mechanism in UDP to correct these issues, therefore a separate mechanism, possibly in another layer (e.g., application layer), should be deployed to detect and correct such issues.

2.5.6 QoS/Security parameters at the Application Layer

In TCP/IP protocol suite, application layer covers all top three OSI layers (application, presentation, and session layers). In general, an application is the highest level of user-interface interaction. Traditionally, less power was given to applications in terms of how the underlying layers (transport and lower layers) interact with the network parameters. However in recent approaches, it has become evident that more intelligence and control in the application layer will result in better performance especially for multimedia-rich mobile traffic contents.

2.5.6.1 QoS at the Application Layer

The application layer QoS's objective is to create QoS-ready schemes at the application layer. At this layer, it is possible to have negotiation of applications on the run and transfer QoS-enabled parameters among different layers to create a soft-state QoS provision. In soft-state, QoS parameters can renegotiate during the application run, creating a dynamic QoS scheme. Soft-state QoS is particularly important during mobility and handoff for performance adaptation.

Application layer QoS requires the per-user-based traffic support of the QoS scheme [121] to enable different levels of qualities for different users. Table 2.8 (adapted from [122]) shows the application QoS requirements. The mentioned classes of service are to enable different levels of services to different types of services and applications.

The application layer QoS includes [122] provisioning of QoS triggered by requests from a QoS policy manager in the network and provisioning of QoS on requests that are signaled to the QoS policy manager at the service layer. Such a signaling scheme can be originated either from outside of the wireless network, from the Subscriber Station (SS), from a host connected to the SS, or from an intermediate server. Another scenario is the provisioning of QoS by the QoS policy manager triggered by on-path signaling processed by policy enforcement points in the network, which in turn generate requests to the policy manager. Such a signaling can be originated either from outside or inside of the wireless network, the SS, or from a host connected to the SS. Application layer QoS should conform to the provisioning for Best Effort or Differentiated QoS across active service flows for backward compatibility and should be consistent to the Service Level Agreements (SLA) across Base Stations (BSs), including during handovers.

Current application layer QoS applications include streaming video and video-telephony applications. In video streaming, the application layer QoS controller resides in the streaming server and is fed by the compressed video/audio feeds and controls the transport protocols. The mentioned application layer QoS controller reduces the chance of congestion and maximizes the video quality in the presence of packet loss. This is done by delay-constrained retransmission and error-resilient encoding mechanism [123].

Application layer QoS is used in variety of applications for the mobile technology, one of which is SIP (Session Initiation Protocol) [124, 125]. The application layer QoS is responsible to grant QoS resources ensuring that both application service provider (ASP) and SBC downstream traffic payloads will comply with the agreed QoS level.

Table 2.8 Application QoS requirements

Service Class Number	Service Class	Application Layer Throughput	End-to-End Transport Layer One-Way Delay	End-to-End Transport Layer One-Way Delay Variation	Transport Layer Information Loss Rate
1	Real-Time Games	50-85 Kbps	< 60 ms preferred	< 30 ms preferred	< 3%
2	Conversational (e.g., VoIP, Video, Phone)	4-384 Kbps	< 60 ms preferred < 200 ms limit	< 20 ms	< 1%
3	Real-Time Streaming (e.g., IPTV, Video Clips)	> 384 Kbps	< 60 ms preferred	< 20 ms preferred	< 0.5%
4	Interactive Applications (e.g., Web Browsing, Email Access, IM)	> 384 Kbps	< 90 ms preferred	N/A	Zero
5	Non-Real-Time Download (e.g., P2P, Movies)	> 384 Kbps	< 90 ms preferred	N/A	Zero

Dynamic Rate Adaptation (DRA) is an application layer QoS mechanism where transmission rates are dynamically changed according to the application criteria. In particular, video and audio applications running on wireless links, use DRA schemes for more efficient bandwidth allocations. In a multimedia application offering DRA, the algorithms adjust the encoding parameters to achieve a target throughput dynamically.

Efficient DRA provisioning at the application layer could lead to congestion control. Congestion control mechanisms are window-based or rate-based. For the rate-based congestion control, the sender should incorporate rate adaptation compression and rate shaping schemes. At the receiver, jitter, error, and delay concealment schemes are required for an end-to-end application layer congestion control system.

Application layer parameters include: Application QoS handler, QoS aware applications, and session-based priorities.

Recent application layer QoS control techniques control packet loss and transmission delays due to network congestion, without any support from the network infrastructure. What we are intending to design is a cross-layer approach to have various feedbacks from other layer for an intelligent application layer QoS scheme.

Application-layer-QoS includes Congestion Control mechanisms and Error Control schemes. Congestion control mechanisms can further be classified into Rate Shaping methods and Rate Control methods. Error Control mechanisms is comprised of: retransmissions, Forward Error Correction (FEC) coding, error resilient coding and error concealment. FEC schemes include: *Hamming distance, Convolutional forward error correction (CFEC), Golay forward error correction (GFEC),* and *Reed-Solomon forward error correction with interleaving (RSFECI)* [126]. Cross-interleaved Reed Solomon codes (CIRCs) are used in both error detection and correction mechanisms, specifically used for countering mixture of random and burst errors.

We are specifically interested in RSFECI as we will be using an adaptive version of this in our system design. Reed-Solomon FEC codes use fixed (input and output) block codes structures. Most commonly used R-S code is the (255, 223) structure, which has 223 input block of 8 bits long symbols producing a 255 encoded output symbols using a systematic conversion scheme. In a systematic conversion, some portion of the output symbols contains the original form of the input symbols. The R-S (255, 223) code can correct up to 16 R-S errors in each codewords, therefore the code can correct up to 16 short burst errors. In general the format is RS (n, k, n-k+1), where n is the length over the finite field F and k is the dimension with a minimum distance of n-k+1.

Rate control can be achieved at the source or receiver or at both of them. Source based rate control techniques are either model based or probe based. In model based approaches, they are based on the throughput model of the TCP, whereas in probe based approaches

at the source, which are experimental in nature and rely receiver feedbacks to adapt the sending rate to the network bandwidth. In a receiver based rate control mechanism, the source is required to transmit data in separate channels with different quality. When receiver detects no congestion then it will add a channel to improve the video's visual quality. However if congestion is detected then the receiver drops a channel, which performs a degradation of the video's visual quality. Besides these individual approaches a hybrid technique exists in which both source and receiver cooperate in achieving rate control. Another technique is the rate shaping that is used to provide congestion control. The basic idea behind rate shaping is to perform transcoding by using rate adapting filters for transmission between links with different bandwidth requirements [127].

2.5.6.1.1 Jitter and Delay Concealment

Multimedia transmission over wireless access technology, in particular, audio applications are especially prone to quality degrading jitters and delays. Delay and jitter concealment schemes at the application layer may reduce such effects. Such schemes introduce adaptive packet-based time-scale modifications at the application layer using adaptive playout algorithms [128] that minimize packet drop, variable arrival delays (jitter), and packets arriving late at the receiver. These schemes maintain constraints on the end-to-end delay using voice segment length stretch and silence interval integration mechanisms.

2.5.6.1.2 Program Clock Synchronization

In the application process for decoding high quality audio and video transmissions, it is crucial to recover high-quality system clocks. For instance, MPEG-2 decoder contains audio and video sample clocks. A program clock synchronization is used to support a high quality application layer audio/video decoding scheme.

There is a time-shift relationship between the maximum nominal bandwidth speed attainable in wired networks compared to that of a wireless network. Approximately there is a 10 year gap, therefore the maximum bandwidth attainable for current wireless

networks (802.11n and WiMAX at nominal speed of approximately a hundred Mbps) match the wired network speeds back in mid 90's. The current line rates attainable according to the current Cisco System switch capacity, is in 5 Gbps [129]. This rate will be raised to 15 Gbps in less than a year. Processing of such a high speed requires hardware-based network processors and since the lowest speed in the 5 layer Internet protocol hierarchy is achieved in the application layer, for a system, in order to keep up with the extreme high rate of line, the application layer has to be implemented in the hardware or firmware. This transition from pure software to hardware is inevitable. Therefore ASIC-based chips will not only house network processors, they will also accommodate application processors as well.

The increase of the line speeds, and as a consequence, the increase of the wireless bandwidths, have so far been incremental. This increase trend of speed will soon find a saturation limit, caused by the physical limitations of the channel and electronic parts. The remedy to this saturation of the line speed is the deployment of new communication strategies and devices.

One method for the hardware-based fast application level switching is the deployment of the Network Based Application Recognition (NBAR) [130, 131] concept. NBAR is capable of recognizing packets with complex fields and attributes combinations. NBAR is able to identify if a packet belongs to certain traffic stream by performing a deep-packet inspection and with an appropriate policy scheme, specific packets can be dealt with accordingly. An entity that works with NBAR is the Protocol Description Language Module (PDLM), which is an application signature. Another entity that NBAR works with is the Cisco Express Forwarding (CEF), which together with NBAR they provide deep-packet classification only on the first stream packet.

The following approaches are the current application layer QoS frameworks: 1). Application Layer Dynamic Services [132], 2). MS-triggered and MS-initiated service flow creation [129], 3). QoS API over Socket (QAoS) Framework [129], and 5). QoS Push-Pull Models [133].

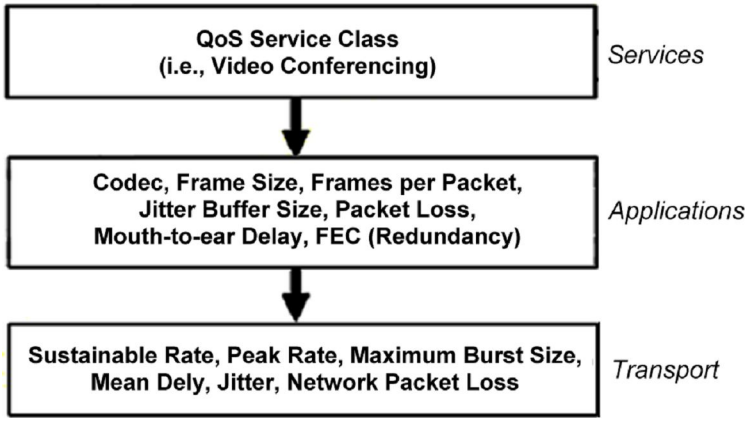

Figure 2.5 Server-to-application-to-transport level flows

In these approaches, application layer agents work closely with agents operating in other layers (especially in MAC layer) and they pass information between each other. A few schemes include: Hybrid Automatic Repeat Request (ARQ) and Error Concealment Jitter Concealment (Program Clock sync). Figure 2.5 (adapted from [134]) shows the approach from service- to application- and to transport levels.

2.5.6.2 Security at the Application Layer

There are various applications offering security modules, which operate at the user interface level. These applications may or may not interact with the underlying layer. The followings are a few application layer security schemes [135]: 1). Identity-based security scheme including authentication, authorization, and shared secret across security domains, 2). Non-repudiation and accountability mechanisms including integrity, archived audit trails, and message level security, 3). Content-based security scheme including application specific security option and buffer-overflow protection scheme, 4). Proxy firewall filters, 5). Pretty Good Privacy (PGP) [136], which provides encryption and authentication for electronic mail services.

Simple Object Access Protocol (SOAP) [137] is another application layer security mechanism used in web services. SOAP relies on Extensible Markup Language (XML) for specification exchange structure. SOAP is able to encrypt messages at the application layer however it reduces the flexibility and interoperability [138].

2.5.6.2.1 DoS at the Application Layer

An adversary may overwhelm the network with application-layer queries, causing the network too many application messages between nodes and the base station. This type of attack wastes network bandwidth that drains energy.

Another application layer DoS attack is the session hijack. For this, an attacker needs to locate all sessions and the participating entities and to hijack all ongoing sessions and keep the sessions ongoing as long as possible. This way all resources will be wasted with no useful communication taking place.

The remedy for the session hijack is to use application layer encryption and authentication mechanisms. Another method is to run a watchdog application to cease applications, which are running open-ended processes aimed for resource drainage.

2.5.6.3 FEC at the Application Layer

Error control mechanisms are classified into four categories: 1).Transport-layer (FEC- and delay-constrained retransmission), 2). Error-resilient encoder coding, 3). Decoder error concealment, 4). Interactive encoder-decoder error control.

Error resilient compression prevents error propagation by limiting the error damage in algorithms that are prone to error propagation.

Application layer Forward Error Correction schemes incorporate content delivery protocols (CDPs) for reliable delivery [139]. In a CDP source blocks (SBs) are created using transport payload multiplexing from various flows.

FEC integrates the following redundant messages into the original message for packet loss compensation: 1). Source-code FEC, 2). Channel code, and 3). Joint source and channel coding.

FEC CDP together with Service Discovery Data (SDD) form a complete FEC protocol suite that covers MPEG-2 (both multicast and unicast for RTP transport stream encapsulation). Another FEC scheme uses raptor codes [140]. Raptor codes repair data using a complex XOR operation sequence.

Error control techniques use FECs with added redundant information to the bit-stream. This is used to facilitate the packet loss reconstruction. Retransmission mechanisms are applicable only in scenarios where obtaining lost packet through retransmission without violating its presentation deadline is possible. Error resilient techniques use multiple encoding description methods for packet loss compensation.

2.5.6.3.1 Error Concealment

Error concealment schemes are often applied at the receiving end where packet loss is detected and measured, using temporal and spatial interpolation algorithms. These schemes are able to estimate the amount of lost data to help conceal and hide the fact that any errors took place and remain transparent from the user and network point of views.

For instance in video applications, error concealment methods use temporal interpolation and spatial techniques to reconstruct the lost information between or within frames. Temporal interpolation scheme copies the pixel information at the same spatial location as in the previous frame. In spatial interpolation, the algorithm operates by estimating the missing pixels using the same frame data.

2.5.6.3.2 Processing Cost Associated to FECs

Forward Error Correction schemes, as mentioned, operate with integrating a number of redundant code bits into the transmitting data streams. The inclusion of these extra redundant bits increases the overhead, as a consequence, impacts the total end-to-end delay figures, batter-power consumption, and throughput. The strength of the FEC code deals with the number of the bits and the type of code used in the FEC scheme. Another important factor in the performance of a FEC scheme is the types of encountered errors in

the transmission. The FEC codes suitable for a bursty environment may be different than those suitable for random single-bit errors. Reference [141] presents a Reed-Solomon (RS) FEC code system for bursty errors, featuring an IEEE 802.11a wireless link (5 GHz band). An FEC technique is used to provide means to reduce Bit Error Rate (BER) figures on an IEEE 802.11a link using Quadruple Phase Shift Keying (QPSK) modulation technique, which provides a good tolerance to noise and interference. The radio channel has a slow fading Rayleigh character. It shows for a burst length of 15 symbols, the FEC scheme requires 73 machine cycles to decode. The number of decoded FEC machine cycles start to increase exponentially as the number of symbols in the burst length increases. As the number of symbols in the burst length increases to 30, the number of decoded FEC machine cycles increases to 450 (exponentially increasing).

Reference [142] presents an application layer Hybrid Error Correction scheme based-on Reed-Solomon mechanism used for DVB (Digital Video Broadcasting) server. Some applications tolerate certain degrees of packet loss and the number of bits used in the HEC scheme (Redundant Information “RI”) is exponentially inversed proportional to the Packet Loss Ratio (PLR).

Though the added required bits in FEC schemes may initially increase the end-to-end delay figures, however the total amount of delay may be far more due to the increased packet loss and retransmissions, in the presence of noise and errors. This is due to the fact that the presence of an FEC scheme may decrease the PLR and/or retransmissions requirements. However in good channel qualities with relatively less amount of interference and noise, the added overhead for FEC schemes may not be beneficial.

The effects of FEC on battery and power consumption figures are also similar to those effects on the delay figures. The inclusion of redundant bits, increases the power consumption, however such an increase of power consumption effects can be equalized and even reduced in erroneous environments where FEC schemes are utilized to correct errors with requiring lesser number of retransmissions. This is due to the fact that in erroneous environment with relatively high numbers of bit error rates (BERs), the

application of an FEC scheme will add a fixed number of redundant bits, which are used to correct error bits without requiring retransmissions. However with the deployment of an FEC scheme, depending on the channel condition, unpredictable number of data blocks may be required to be retransmitted, which may increase the overhead much more than the overhead increase due to the FEC scheme deployment. Reference [143] presents a 4x4 64-QAM (Quadrature Amplitude Modulation) systolic soft detector scheme featuring a single detector used in a MIMO (Multiple Input Multiple Output) system. The performance of this system is compared against two other schemes without FEC mechanisms. The performance measures includes power consumption and Energy per bit, which shows major power reduction in these two power-related parameters.

Reference [144] presents an adaptive FEC scheme used in an interactive streaming IP application. This scheme takes aim at the packet loss rates and loss burst sizes by constructing a predictive adaptive scheme.

2.5.6.3.3 Encryption/Decryption Interactions with FECs

Several references [145, 146, 147] suggest the integration of encryption/decryption techniques with FEC mechanisms to optimize both functions simultaneously. In particular in the case of block cipher schemes, where a single error could propagate to the consecutive blocks, such as in Advanced Encryption Standard (AES) Cipher Algorithm in Cipher Block Chaining (CBC) Mode (AES-CBC) or AES-Counter with CBC-MAC (AES-CCM) [148]. Such integration could prevent the error propagation, thus reducing large amount of retransmissions. Reference [149] presents the idea of integrating FEC (based on Reed-Solomon) and encryption (based on AES) at the MAC layer for telecommand and telemetry applications.

2.6 Cross-Layer Wireless QoS/Security Provisioning

In this section we will discuss QoS and security parameters that interact between various layers in a cross-layer context and eventually we will study various criteria for considering such provisioning at the applications layer.

2.6.1 Cross Layer Design

In today's wireless networks, in order to have maximal performance delivery; capacity, QoS, and security have to be maintained at high levels, while keeping the complexity and energy consumption at lowest levels. This is an extremely difficult task to do and one method to reach this goal is cross layer design. The idea of cross layer is similar to the idea of team-work, where each layer considers the current conditions at other layers before performing any process.

The main objective of cross layer design falls into the following categories:

Bounded End-to-End Delay – In order to decrease the overall time delay from application layer down to the physical layer, a suggestion would be to pass around delay-specific parameters between layers to speed up processing in bottleneck layers. This is particularly efficient in the lower layers (Network, MAC, and PHY layers)

Bounded Packet Loss – Some decisions made at certain layers may contribute to relatively more packet loss as the information moves down the layers. In order to find the best relative delay/loss figures optimized for most layers, cross layer design may be helpful.

Bandwidth – The amount of data passing through each layer per unit time is one of the performance measures. Such a metric tends to increase at information moves down from application layer down to the physical layer. Parallel processing and pipelining schemes are available at the application layer and on the sender's processing units, however bottlenecks are towards the MAC and PHY layers where resources are limited and access technologies deal with multiple access challenges. However using cross layer approaches, a more unified method can be utilized where all layers operate with enhancements destined for the lower layers.

Energy resource – Energy is scarce in wireless environments, therefore any schemes that preserve more battery resources for the same performance, would be welcome in such environments. Layers prone to Denial of Service (DoS) attack (e.g., PHY, MAC, network, and transport layers) are especially under consideration, due to the fact that power resources can be compromised easily in these layers. A cross layer design helps with a sound energy consumption scheme between these layers.

Mobility – Mobility requires seamless connectivity while the client is on physically changing locations. This mechanism involves various layers, including PHY, MAC, network, and transport layers [150].

2.6.2 Challenges in Cross-Layer Designs

As mentioned in this chapter, one of the philosophies behind layered architectures is to assign specific tasks handled at each layer. The cross-layer design considers import and/or export of parameters to and from layers. One of the challenges is that gathering parameters may increase the delay figures. The other challenge is the extra overhead that passing cross-layer parameters will cause. This extra overhead can affect the performance of the communication system, including the throughput, delay, and jitter figures. Another challenge is that suboptimal performance of the cross-layer design. This is due to the fact that the functions performed at each layer are optimized in a traditional sense, therefore when those functions are affected by the cross-layer design, the resulting interactions usually increase performance in some areas and may cause suboptimal performance in other areas. Therefore finding a balance between performance, the number of gathered parameters, and the impact on the throughput and delay, is a challenging task.

2.6.3 Cross-Layer Interactions between Layers

In this section, the cross-layer interactions between every individual layers will be discussed briefly.

2.6.3.1 Interactions and challenges between Physical Layer and Other Layers

Mechanisms operating at the physical layer are often involved with power transmission, modulation and coding schemes, bit error rate (BER), and transmission rates.

The adoption of higher power transmission may usually lead to the increase of transmission rates and decrease of the BER. However there are limitations to this, one of which is the presence of other transmitters in the proximity of the transmitter, specially if they are transmitting on the same or close channels, which in addition to the increased noise level, wireless collisions will also be increased, which drops the performance greatly. Therefore the challenge is to find the balance between the power, allocation of transmission channel, and optimal performance.

Importing parameters from different layers down to the physical layer may increase adaptability of the system in a cross-layer concept. For instance Physical + MAC layers may adapt the transmission rates according to the channel access scenarios. Transmission rates at the PHY layer may inform the MAC layer to adapt the Maximum Transmission Unit (MTU) for better efficiency.

At the network layer, physical layer information may be used for proper routing algorithm selection that suits the energy/bandwidth constraints at the PHY layer.

At the transport layer, PHY layer parameters are used to adapt the retransmission schemes according to the power and bit-rate limitations.

At the application layer, many multimedia encoding techniques can use rate adaptations according to the information imported from the physical layer [151].

All these transfer of parameters may increase the complexity of interfaces between layers, which may translate to higher delay and jitter figures and lower throughput numbers. In particular, the challenge is to find the optimum point for the highest transmission rate and the lowest transmission power which is the challenge that cross-layer design faces at the physical layer.

2.6.3.2 Interactions and challenges between MAC Layer and Other Layers

MAC layer is a very important layer in terms of the support for legacy protocols. This is due to the fact that many current and past protocols have extensive MAC layer QoS and security provisioning implementations.

MAC layer security and QoS parameters can match with the rate adaptation schemes at the physical layer for a better QoS/security performance that matches the PHY rates. Another issue is energy where power level and rate adaptation are coupled [152].

MAC/network layers may involve efficient routing algorithm (e.g., for ad-hoc networks) selections based on MAC layer control signals. For instance one scheme [153] proposes a MAC/network layers interaction combining CDMA/CA, RTS/CTS, and scheduling algorithms, which all operate at the MAC layer to support a multicast routing protocol, which operates at the network layer.

At the transport layer, MAC layer information can efficiently influence the TCP algorithm. For instance, the Maximum Transmission Unit (MTU) at the MAC layer can be coupled with the Sliding Window size at the transport layer. This way a fairly good wireless link with a large MTU size may also experience relatively smooth TCP handshakes. Therefore MTU information can be passed to the TCP algorithm for a faster Sliding Window size adaptation.

At the application layer, QoS/Security parameters at the MAC layer can be imported to avoid applying more QoS/Security algorithm than needed. On the other hand, the lack of proper QoS/Security mechanisms at the MAC layer can be compensated at the application layer.

The challenge of cross-layer design at the MAC layer is to make decisions based on QoS and security parameters, which usually negate one another. Finding an optimum scenario to take both QoS and security cross-layer parameters into consideration and to select appropriate actions (based on scheduling, queuing, security mechanism, etc.) within limited delay figures, is the real challenge of cross-layer design at this layer.

2.6.3.3 Interactions and challenges between Network Layer and Other Layers

Network layer is mostly concerned with routing. Therefore any optimization at this layer may have an effect on how packets are routed from the source to the destination networks.

At the network layer, transport layer-based parameters can be imported to update routing information based on link quality. That is on a failing link, before application layer starts get impacted (user-experience dropping below acceptable range), the poor TCP performance can be monitored at the network layer triggering a route change.

One of the cross layer interactions between network and application layers manifests itself in the QoS (Quality of Service) to QoE (Quality of Experience) relationship. QoS schemes at the network layer includes the DiffServ aggregation scheme and QoE directly deals with how quality is perceived by the user at the application layer.

In the TCP/IP protocol suite, IP is the network layer protocol and the fields (processes) in IP header that can be used in the network-layer cross-layer design, include: DSCP, Identification (ID), Flags, Fragment Offset, Time-to-Live (TTL), Protocol, and Options. The DSCP field, as mentioned, plays a vital role in providing network-layer QoS provisioning. The ID field can be used to add packet-tracing information to datagrams to help trace back datagrams, which my be subject to spoofed source addresses. Flags are used to control fragmentation. This is particularly important during fragmenting large amount of continuous data. TTL is also used to limit the number of hops before the packet is removed to avoid packets entering in countless loops. The Protocol field defines the protocol used in the data field. The Option field is used in regards to the fragmentation functionality and routing capabilities (i.e., LSRR “Loose Source Record Route” and SSRR “Strict Source Record Route”). These parameters can be exported from the network layer and used in the cross-layer design.

The challenge of cross-layer design at the network layer is to make fast routing decisions based on the parameters gathered from various layer. The type of application in

use may be important in the routing selection. For example the selection criteria of a route for a heavy multimedia-based traffic requires relatively high amount of bandwidths with limited tolerance for delay, is different than the route selection criteria for an email-based traffic, which is non-interactive and allows extra delay. This information can be imported from the application layer. However the links are not usually dedicated to one specific application and the transmitting data is usually a mixture of various data types with various delay-bandwidth requirements, which makes the routing selection difficult. Therefore routing selections based on cross-layer design may encounter suboptimal decisions, which is another challenge of cross-layer design at this layer.

2.6.3.4 Interactions and challenges between Transport Layer and Other Layers

Link quality at the transport layer can trigger coding rate adaptation at the application layer. Therefore if the quality of the link is perceived to be high at the transport layer, then the application layer multimedia encoding schemes can adapt to higher rates or incorporate stronger encryption algorithms without compromising the mechanism at the transport layer. Application layer, on the other hand, may impact the way sessions are created and handled. The duration of a session may be limited or extended based on the application in use. Through cross-layer design, parameters from other layers may be imported and the decision over the session quality (e.g., duration, size of the sliding window) could be made based on the imported parameters.

The challenges the cross-layer design face at the transport layer are based on the decision on the quality of the session. The feedback from the application layer may not always go hand-in-hand with the feedback from lower layers, therefore the decision may lead to a suboptimal selection of the session quality, which may have rather negative impacts on the quality of one or more layers. The challenge is to take all imported cross-layer parameters and make the best possible decision that would have minimum negative impact on the operation of other layers.

2.6.3.5 Interactions and challenges between Application Layer and Other Layers

The cross layer design, involving the application layer, is particularly important in this thesis due to the fact that this layer holds the key to a new paradigm shift. In this new architectural design, many important parameters belonging to various layers can efficiently be imported to this layer via cross layer interactions, particularly known to increase efficiency for wireless multimedia applications.

There are two major application-layer cross layer interaction flows: Importing parameters from different layers to the application layer, and exporting processed information back to various layers.

Application layer is an important layer for multimedia processing. Encoding, decoding, user-selective QoS settings, application-based encryption and security, and many other mechanisms could take place at this layer. These mechanisms can work more efficiently when they interact with the underlying layers.

The challenges that the cross-layer design face at the application-layer are based on the fact that user-driven applications may be impacted by the import of parameters from lower layers, which in turn may not increase the QoE to the user. The challenge is to have a balance between the QoE-level and the feedback from lower layers.

As mentioned, wireless multimedia applications require application-intensive processing for higher utilization and content-aware bandwidth usage, as well as processes done at the application encoder-decoder. Therefore many parameters are to be imported from various layers. These include:

2.6.5.1.1 PHY-to-Application Layers

Information about the signal strength and/or transmission rates can inform a dynamic application system to adjust the encoder quality according to the link quality. From security point of view, actual bit-level conversations (e.g., encryption, decryption, block-cipher, etc.) are done at PHY layer.

2.6.5.1.2 MAC-to-Application Layers

MAC layer features numerous performance metrics and parameters. From QoS point of view, MAC layer offers: Call Admission Control, MAC layer queuing (e.g., AC_VO, AC_VI for Wi-Fi). From security point of view, MAC layer offers frame structures for security protocols, such as WPA and 802.11i and works hand-in-hand with the PHY layer, where the bit-related mechanisms (e.g., encryption) are realized. A cross-layer scheme that transports parameters from MAC/PHY Layers to the application layer, is able to extend and optimize the security capability to the application layer.

2.6.5.1.3 Network-to-Application Layers

Availability of network layer QoS (e.g., DiffServ) and passing through high quality routes are valuable indications that can be used at the application layer, which can trigger optimized usage of the application encoder. From the security point of view, IPSec (Internet Protocol Security) operates at the network layer. The availability and deployment of IPSec (and VPNs “Virtual Private Networks”) poses some operational limitations to avoid any security compromises. These parameters are to be transported and used at the application layer.

2.6.5.1.4 Transport-to-Application Layers

A high sliding window size (transmitting a relatively large number of segments at once) is an indication of a high quality link with better than acceptable range of end-to-end delays. This is due to the fact that a large number of segments can be transmitted from the source to the destination before the source requires an acknowledgement from the destination. This can be transported to the application layer to inform the application multimedia loader to adapt to a better quality efficiency.

2.7 Non-Repudiation Multimedia-based Wireless Systems

We have so far had detailed discussions on both security and QoS requirements for secure multimedia communications. We also considered cross-layer approaches with

special focus on QoS and security. This is particularly important due to the fact that a new shift in the protocol design is required, by which higher layers are given more intelligence in dealing with QoS and security decisions.

2.7.1 Introduction

This section discusses the mechanism of non-repudiation systems. The doctor/patient pair is a good example for a medical environment, where there happens to be many scenarios where previous communications are needed to be verified in regards to the involving parties, such that no involved parties should be allowed to deny his/her involvements. For this a digital signature is accompanied on every transmission flow that conveys specific information about the sender. This piece of evidence can not be disputed later on by any involving parties.

The effects of the addition of these signed information bits are required to be studied and the performance of this system should be discussed in detail.

2.7.2 Non-Repudiation Existing Solutions

Robust authentication for multimedia data in the presence of channel noise can be of a challenge in particular for multimedia-enabled wireless traffic [154]. The reference [154] integrates a signature-based authentication framework of Joint Source and Channel Coding (JSCC) for an adaptive approach achieve to efficient bandwidth utilization using authentication graph construction and optimal resource allocation.

Such a signature-based authentication scheme is used to offer non-repudiation and integrity mechanisms while resisting packet loss. This is achieved through a resource allocation authenticity protection algorithm.

End-to-end security for mobile application is another challenge including providing security for Short Message Service (SMS) and Multimedia Message Service (MMS). The end-to-end security scheme should address the needs for, authentication, confidentiality,

integrity and non-repudiation. Reference [155] uses an identity-based cryptography solution to provide Service Provider (SP)-based end-to-end security coverage, which covers SP-to mobile users (MUs) and MUs-to-MUs with limited on-device storage requirement and the ability of being integrated with the current and existing technologies.

The proposed scheme provides only partial non-repudiation mechanism to the messaging services due to the fact that the SP provides private keys without any non-repudiation assurance from the user, therefore the user may deny that it had submitted a message signed with its private key. To solve this issue a traceable path is required with mobile network's access security agent.

ESAWN-NR (Extended Secure Aggregation for Wireless Sensor Network) is a new scheme that offers data aggregation authenticity and the ability to prove aggregation forgery [156]. Therefore ESAWN-NR not only detects forged data but also automatically corrects it by excluding the compromised and illegitimate nodes and replacing them with non-compromised and legitimate nodes. A compromised node will be prevented from forgery repudiation in any given time. This is due to the fact that data is authentically transported end-to-end between any two nodes and all data aggregations are verified.

Through theoretical analysis and simulations, ESAWN-NR presents a scheme, which has a power-save capability while maintaining non-repudiation functionality.

Session Initiation Protocol (SIP) is an important VoIP application with lightweight computational requirements, which has become an attractive mobile-based application. Various multimedia services can be transmitted on top of SIP and since the user's packetized voice is the predominant traffic in SIP, providing non-repudiation services become vital, which requires a special coordination between the application service provider and the user. Reference [157] presents such a service without requiring any additional time and power consuming processing and any additional modification to the existing SIP accounts. In this scheme an Authentication, Authorization and Accounting (AAA) agent is used to arbiter the public key of SIP proxy for the SIP service sessions.

Reference [158] proposes a mobile-based PKI (Public Key Infrastructure) authentication protocol scheme featuring digital signature and non-repudiation algorithms for AAA. A single sign-on protocol is adopted to reduce the authentication latency and user intervention delay using proxy certificate delegation mechanism. Kerberos functionality has also been considered in this scheme.

Sensor networks are deployed in variety of scenarios, especially in distributed wireless context, where each node has forwarding and processing capabilities. However security can become extremely difficult and crucial due to the complexity of the varying and exposed topology. The reason is that each node can be a potential attach target and all involved nodes can be targeted one way or another. A framework; Tara security framework has been proposed to not only achieve access control, authentication, and non-repudiation services, but to offer security services with energy efficiency in mind [159]. Tara security framework is based on four main components; LSRP (Lightweight Secure Protocol), LKMS (Lightweight Key Management Scheme), LTMS (Lightweight Trust Management System), and LIDS (Lightweight Intrusion Detection System). It is shown that Tara security framework is capable of combating various attacks, including; bogus routing, wormholes, and etc..

2.8 Battery Consumption

Battery power is a limited resource, in particular for wireless devices, and every effort has to be taken to reduce the battery consumption. There are many performance parameters that have effects on the battery consumption, including: security algorithms and average payload size. Figure 2.6 (adapted from [101]) shows the comparative battery consumption for the transmission of the same number of bytes that a wireless device sends, in the presence of WEP and no encryption. The performance results are carried out on a Toshiba 1200 series laptop featuring a Celeron M 1.2 GHz CPU with 256 MB RAM. Figure 2.7 shows the comparative battery consumption in the presence of AES 128, AES 256, and no encryption.

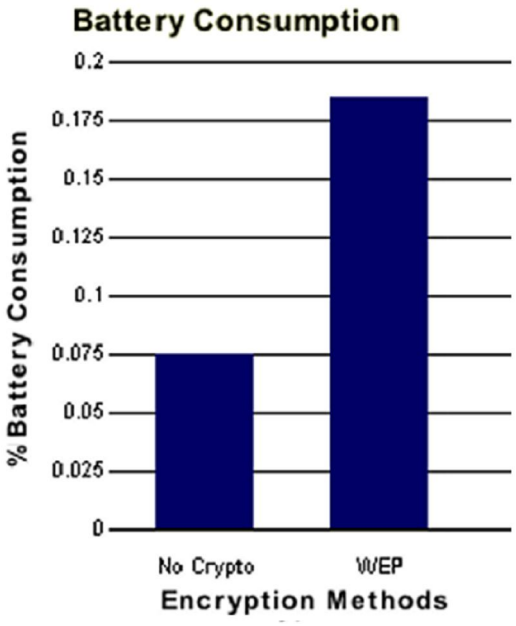

Figure 2.6 Effect of deploying WEP on increased battery consumption percentage

Figure 2.8 shows the battery consumption (Toshiba 1200 series laptop featuring a Celeron M 1.2 GHz CPU with 256 MB RAM) against the packet size. It is understandable that as the packet size increases, for a constant file size, the number of packet transmissions decrease and the energy per packet also drops.

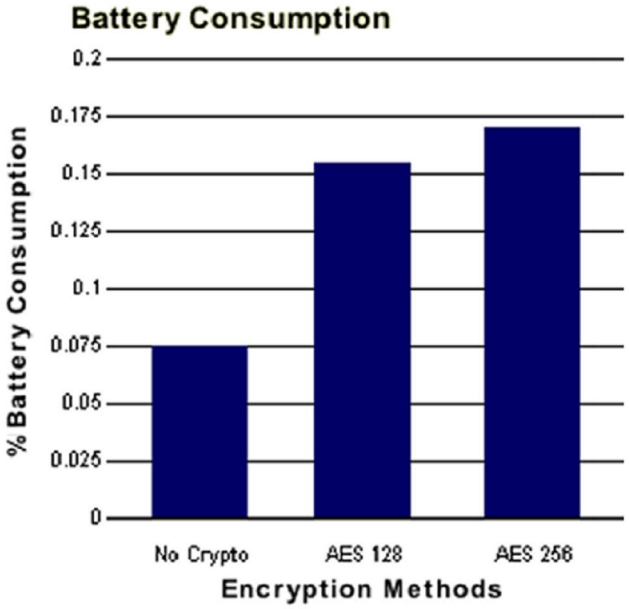

Figure 2.7 Effect of deploying AES 128 and 256 on increased battery consumption percentage

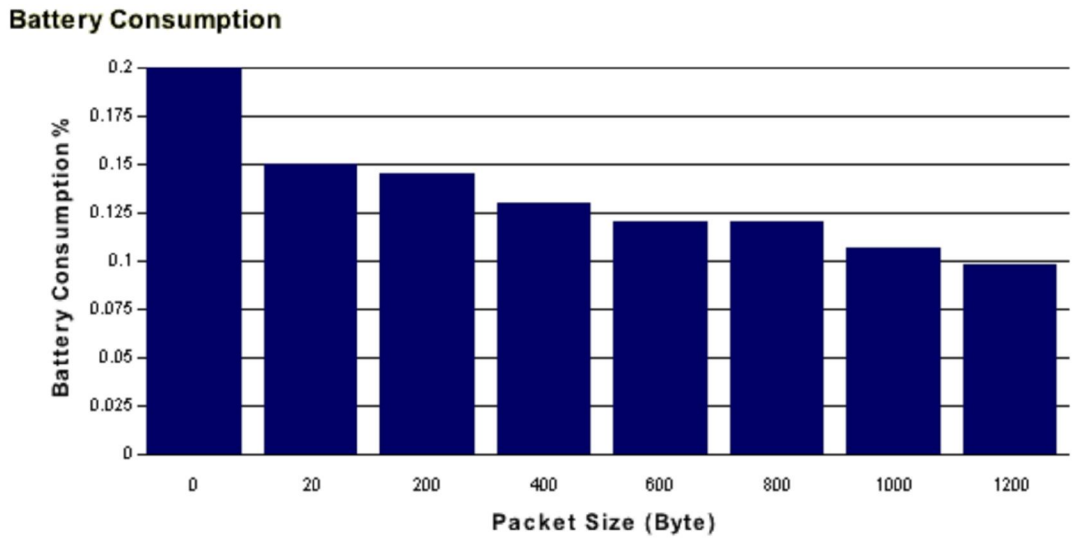

Figure 2.8 Battery consumption versus packet size

2.9 Conclusion

In this chapter we reviewed the legacy and current literatures on QoS and security. We studied traffic classifications in three major categories: packet-, flow, and application-based traffic classification techniques. Wireless multilayer and cross-layer design were discussed, following non-repudiation multimedia-based wireless system techniques.

Chapter 3
QoS and Security Models

In this chapter, QoS and security models are presented, which will be followed by the design, implementation, analysis (e.g., performance and security analyses), and evaluation of an application layer non-repudiation wireless system, which features a cross-layer technique conforming to the traffic classification results. One of the main objectives is the transportation of full-multimedia contents (e.g., text, voice, and video) while supporting Suite-B cryptographic algorithms, offering various security mechanisms, specifically supporting a non-repudiation scheme. For both cases of wireless and wired access technologies, the traffic flows need to be monitored, which indicate the amount of bandwidth usage, connection time periods, average number of bytes per packet, and etc..

Through traffic classification techniques, as mentioned, it would be possible to estimate a few essential flow parameters, including the average number of packets per unit of time on particular wireless and wired links. In the design of this system, the statistical values related to the traffic measures need to be in-line with the medium's average traffic parameter ranges. If so then the deployment of this system will not likely contribute to a congestion scenario in the network.

The traffic generated by this system will feature a few QoS and security related parameters, which will be packed in the UDP payloads and sent via a Wi-Fi link. The amount of added overhead and time delays should be bounded to both limits set by the multimedia performance and traffic classification requirements.

Following wired and wireless traffic analyses, we will introduce QoS and security related model parameters, which are considered in our analytical and experimental results.

3.1 Traffic Classification – Real-Time Data Measures

In this section we analyze real captured data using Wireshark and Omnipeek Software systems. Wireshark is used to monitor wired transmitted packets and Omnipeek is designed to monitor air transmitted packets (wireless) [160,161].

3.1.1 Wired (Wireshark) Data Analysis

Wireshark is a special tool designed for line traffic monitoring. We have probed a data trunk (with prior permission) that is serving a network with tens of desktops, laptops, controllers, and access points (APs), dealing with multimedia-rich and real-time applications, as well as, non-real-time applications.

For the validity of real-traffic monitoring, Wireshark was run for a long period of time capturing live traffic passing through the network. We noticed that four major time periods during a working day had significant unique traffic patterns:

8 AM – 9 AM: During this time period, work-related (e.g., university-based) emails and text-based applications (e.g., remote login) are mostly being used by users. Therefore smaller packet sizes are encountered with less multimedia contents.

12 PM – 1 PM: In this time period, messenger services (e.g., Yahoo, MSN), multimedia applications (e.g., YouTube) and fun-related data (e.g., online gaming), are mostly used. Captured packets show mostly large packet sizes with multimedia related payloads (including RTP, UDP, TCP, codecs, etc.).

4 PM – 5 PM: During this period of time, the nature of the traffic is more of text-based and graphic-based attachments (e.g., Microsoft Word, Adobe Acrobat files, etc.). The sizes are again relatively large with less multimedia contents.

12 AM – 1 AM: In this time frame, most of the data transfer includes backup information with average size packets.

Table 3.1 Average data statistics

Traffic Parameter	Parameter Values
Packets	32,147
Bytes	21,745,794
Duration of Monitoring	411 sec
Average packets/seconds	78
Average Mbps	0.422
Average bytes/sec	52,802
Average packet size	676 bytes

For traffic classification, 32,147 packets were randomly selected from a pool of captured data (+10 GB) from all four groups. The randomness guarantees a more realistic traffic selection.

3.1.1.1 Data Analysis

The traffic statistics are shown in Table 3.1.

3.1.1.2 Connection Durability

As mentioned in chapter 2, heavy hitters are categorized as per packet size and as per duration length. Ongoing connections are the flows which have increased duration lengths. Table 3.1 shows 32,147 packets were under observations, which are grouped based on the protocols in use. The protocols carrying data are: IPv4, TCP, and UDP. Table A.5 shows the protocol hierarchy used in the captured packets.. Table A.6 shows the Ethernet packet statistics ordered by the packet sizes.

IPv4 - There are 28,166 packets out of 32,147 packets (87.62% of the total number of packets) using IPv4 and only 7 packets (0.02%) used IPv6. These 28,166 packets are generated by 305 individual IP addresses (end-points) on the network. Wireshark is able to classify the number of IP packets generated per node. Table A.7 shows the statistics of the nodes and the generated IP packets associated to each node, sorted from the highest

number of IPv4 packets (i.e., 25,903 IPv4 packets generated by one node) showing in the top of the list down to 27 packets. This list also represents the heavy-hitters. For privacy reasons the actual end-point IP addressed are removed.

Most of the heavy-hitters on the top of lists are nodes participating in ongoing multimedia streaming over a large period of time. These nodes are not only categorized as heavy-hitters from the time period point of view, however they are usually packets with relatively large amount of payloads (mostly TCP or UDP based), which would qualify them as heavy-hitters from packet size point of view as well.

As an observation, the first item on the list has transmitted 25,903 IPv4 packets carrying 21,066,311 bytes, each packet contains 813 bytes on average. Another observation points to the fact that this node has been receiving more packets (15,513 packets) than transmitting (10,390 packets). A closer look at the traces shows that this node has received relatively a large number of packets (mostly containing multimedia payloads) in the receiving mode and in the transmitting mode has sent a relatively small number of packets (mostly containing acknowledgments).

TCP - The same scenario holds for TCP. Table A.8 shows ordered heavy hitter TCP end-points. It should be noted that besides IP address, TCP uses port numbers associated to specific applications.

UDP - Both UDP and TCP are transport protocols with similar parameters. The UDP heavy hitter end-points are listed in order of the number of packets in Table A.9.

3.1.1.3 Protocol Packet Scatter Plots

Scatter plots are very useful in terms of representing the population of two and three dimensional data in respect to one or more parameters. Based on the Table A.6, the two-dimensional (duration, number of packets) scatter plot for Ethernet traffic is shown in Figure 3.1 [162].

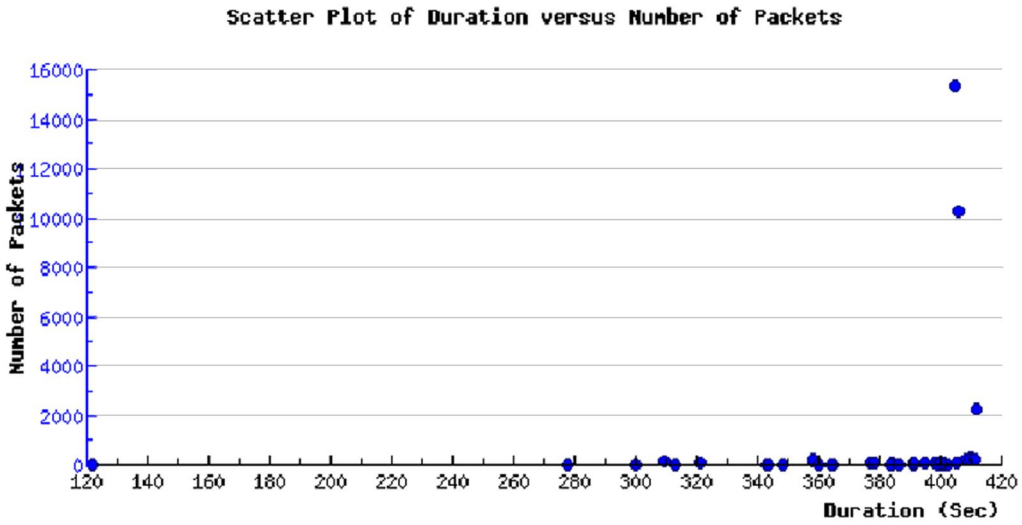

Figure 3.1 Scatter plot of Ethernet packets, duration vs. the number of transmitted packets

The pattern in Figure 3.1 shows that only a few heavy hitters exist (less than 5%). The average duration time is approximately 360 seconds.

Each flow specified in Table A.6 may be associated to a different average packet size. Using the number of packets and the number of transmitted bytes, the average number of bytes per packet can be calculated. For example in the first flow item of Table A.6, the average number of bytes per packet is 20,079,465/15340 = 1309 bytes per packet, whereas in the second flow, we have: 948,450/10272 = 92 bytes per packets, which shows quite a large difference. The first flow may include more multimedia-rich payloads, which are often heavy in size, whereas the second flow may include more control/management frames. Figure 3.2 shows the scatter plot for flow-based average bytes per packet, which shows a relatively high density graph around the 340-410 seconds duration and average number of bytes between 150 to 250 bytes.

Figure 3.3 shows the scatter plot for the number of transmitted bytes against the average bytes per packet per flow. Figures 3.3 and 3.4 are the same, except for the first three points (15340, 1309), (10272, 92), and (2262, 60). Without these three points, the rest of the scatter plot points are more expanded and visible. Figures 3.3 and 3.4 show that the density of the graph increases from the (10, 100) point to the (50, 300) point.

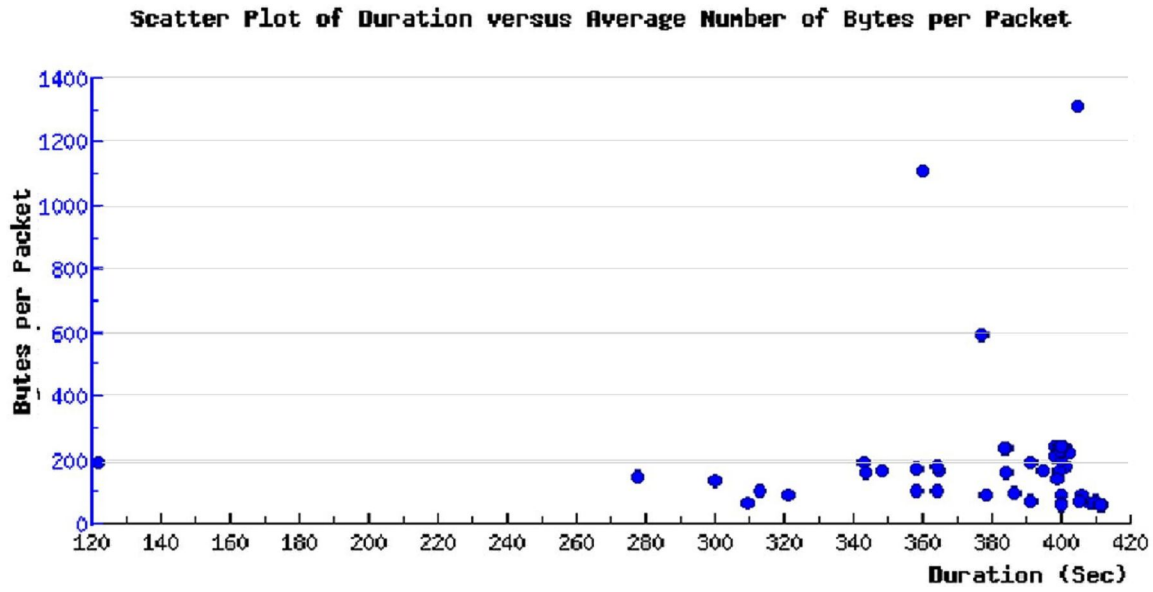

Figure 3.2 Scatter plot of Ethernet packets, duration vs. the number of transmitted bytes per packet

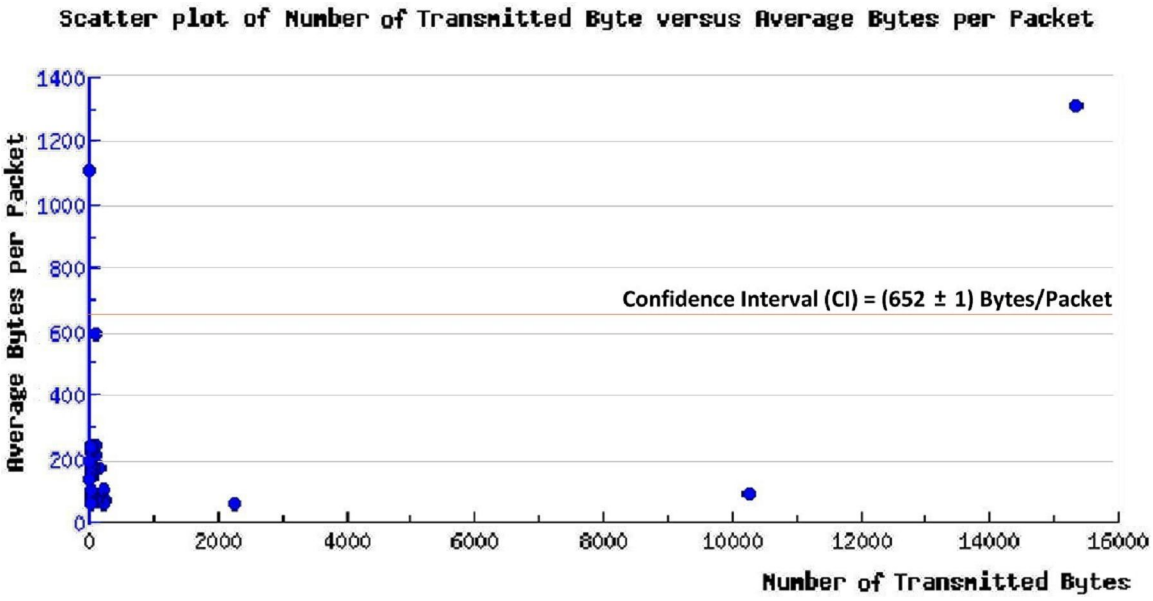

Figure 3.3 Scatter plot of Ethernet packets, number of transmitted bytes per packet vs. average bytes per packet

The average number of bytes per packet is 652 ± 1 bytes/packet. This calculation will be explained in section 3.1.1.5.

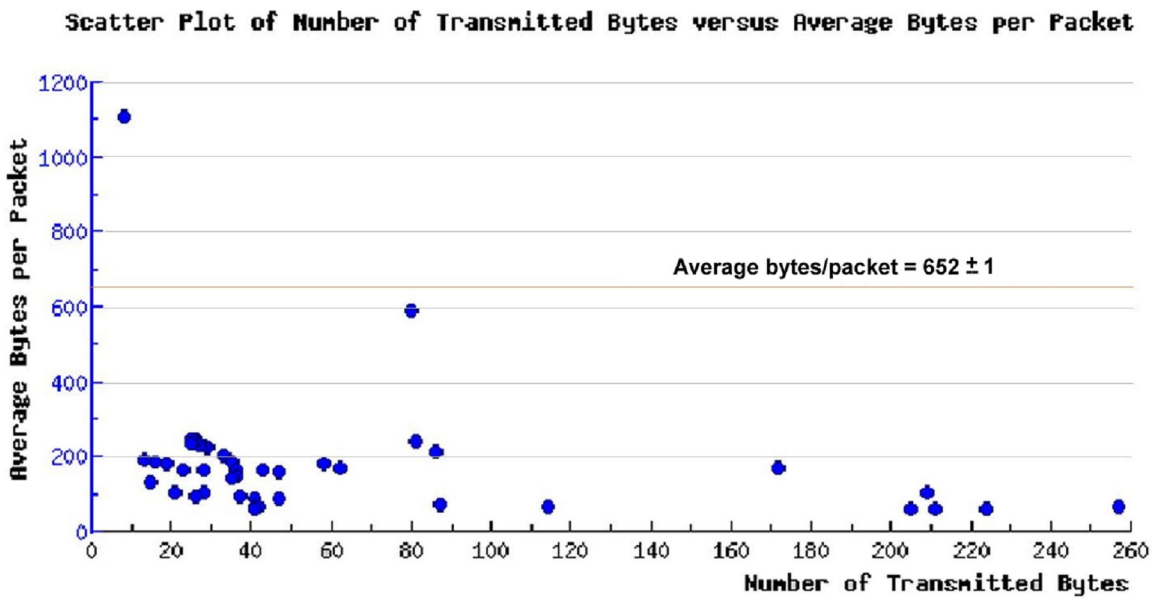

Figure 3.4 Expanded scatter plot of Ethernet packets, number of transmitted bytes per packet vs. average bytes per packet (based on Figure 3.3)

3.1.1.4 Throughput Graph

Table A.5 shows the hierarchy of protocol, as mentioned before. Ethernet frames form 99.6% of the total traffic. Cisco ISL (Inter-Switch Link) traffic form the other 0.04%.

From the 99.6% of the Ethernet traffic, 87.62% belongs to IPv4 and the rest of 12.34% belongs to other traffic types, including: Logical Link Control, Configuration Test Protocol (loopback), IPv6, and etc..

From the 87.62% IPv4 traffic, 78.81% runs on top of TCP, 8.36% runs on top of UDP, and the rest of 0.45% belongs to other types.

Figures 3.5, 3.6, and 3.7 show the bandwidth usages for IPv4, TCP, and UDP. IPv4 and TCP graphs show high degree of correlation. This is due to the fact that the streaming data is transmitted using TCP on top of IP. Furthermore TCP is the main transport protocol used in HTTP connections. UDP on the other hand, seems to be less correlated to IPv4, due to the fact that UDP is mainly used for controlling purposes. Figure 3.8 shows the comparative bandwidth usage between HTTP, UDP, and ARP messages. HTTP uses TCP and ARP is a layer II protocol (MAC layer service).

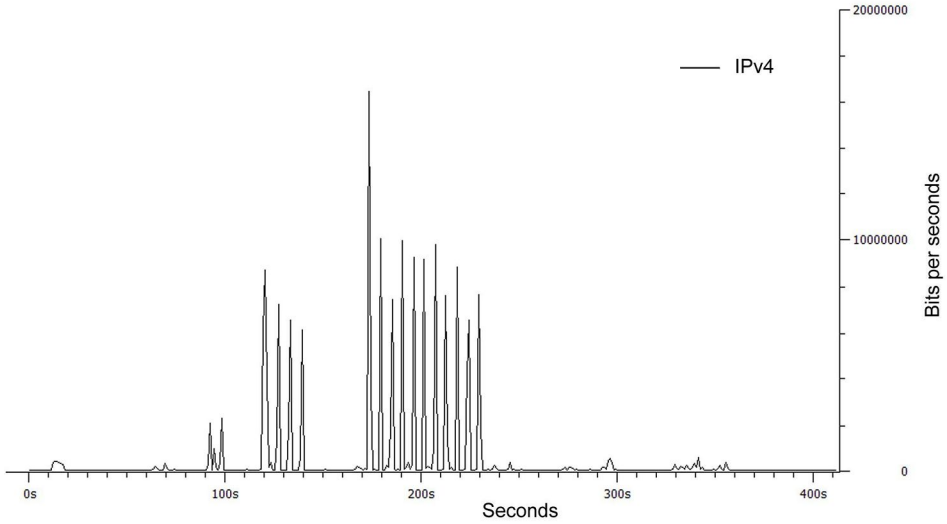

Figure 3.5 IPv4 bandwidth usage

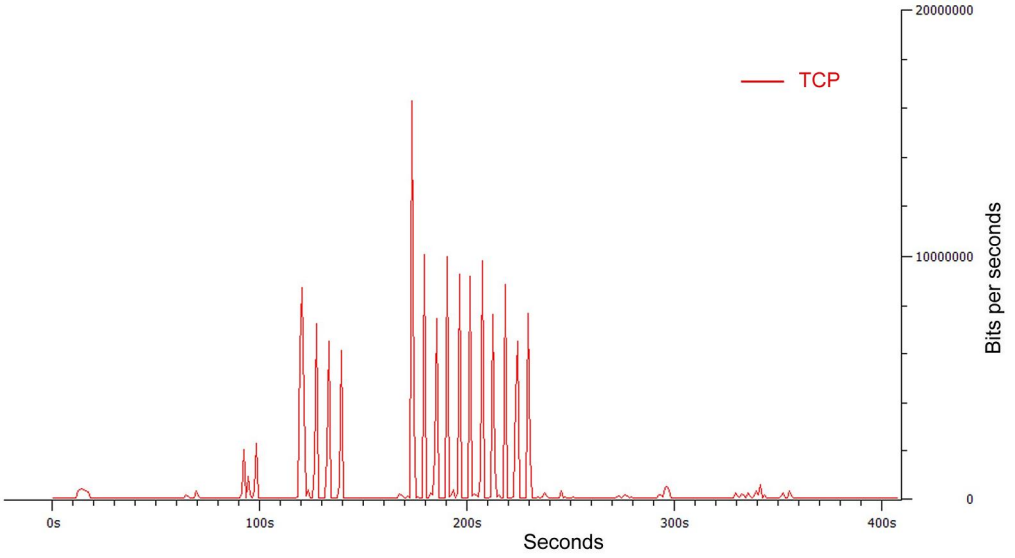

Figure 3.6 TCP bandwidth usage

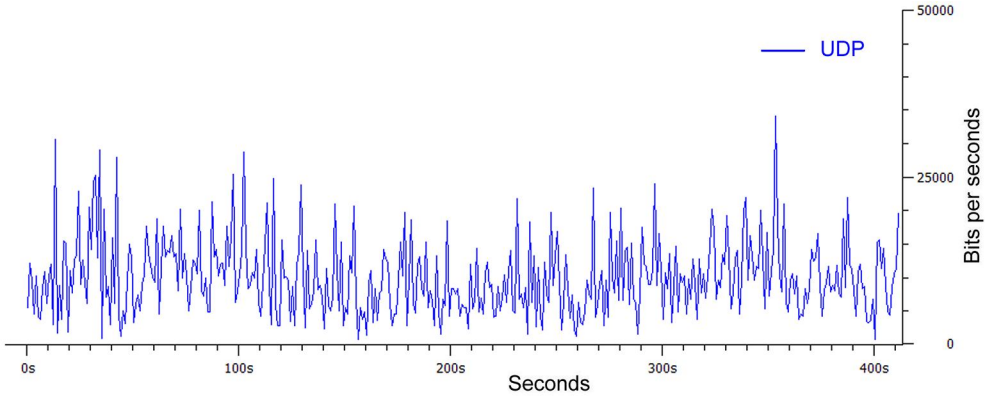

Figure 3.7 UDP bandwidth usage

3.1.1.5 Confidence Interval

The line traffic traces were gathered over a time period of one month. The results were analyzed by the Wireshark software and the statistics were carried out. To represent the graphs, we have used a time-slot, which had the closest statistical variable to the overall statistics. The 400 seconds of time period for graphs 3.5 and 3.6 can be fitted by a Gaussian model. In fact we use Gaussian approximation to calculate the confidence interval. For this we use several statistics, including those of Table 3.1. These statistics are gathered randomly from different dates and daily time periods. These are shown in Tables A.13, 3.2, and 3.3. Table 3.2 shows the statistics about the average number of packet sizes and Table 3.3 shows the average flow durations. Figure 3.9 contains packet size distribution statistics measured using the entire 10 GB of data traces. Figure 3.9 can be approximated well with a Gaussian distribution.

Table 3.2 Average packet size statistics

Packet Size (Bytes)	Number of Packets
676	32,147
793	19,524
794	16,956
834	13,794
791	16,653
454	21,746
369	16,380
581	25,210

Table 3.3 Average flow duration statistics

Flow Duration (Sec)	Number of Packets
411	32,147
270	19,524
429	16,956
169	13,794
242	16,653
500	21,746
431	16,380
489	25,210

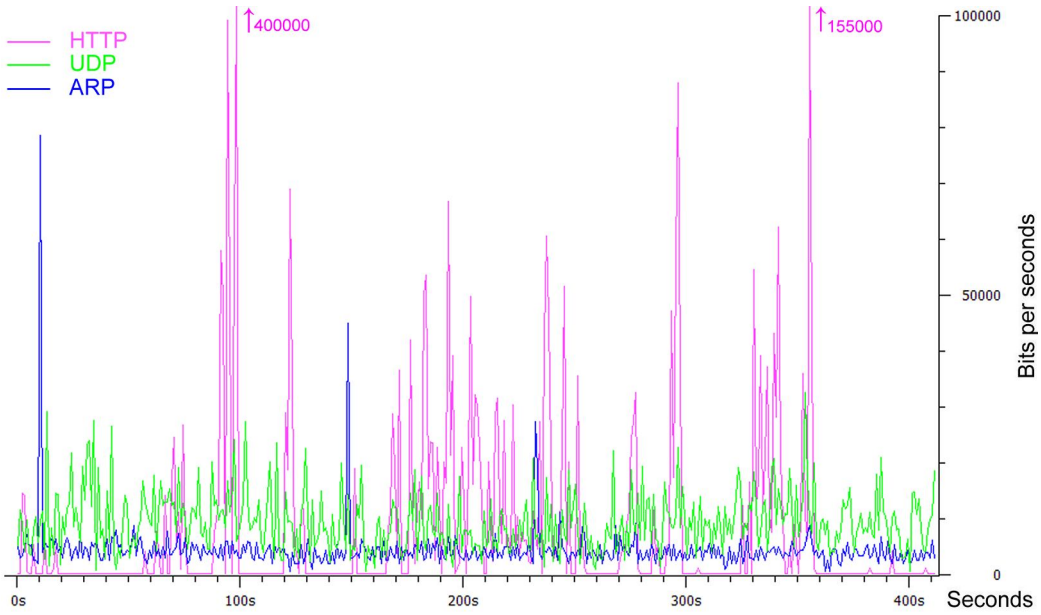

Figure 3.8 IPv4/TCP/UDP bandwidth usage

The average packet sizes, variance, and standard deviation values can be calculated from Table 3.2 data. There are a total of 162,410 packets containing over 105,986,883 bytes. The total average packet size = 652 (bytes), and Standard Deviation (STD) = 153 bytes, and the average bytes per packet is 652 ± 1 bytes. The same calculation is carried out for the flow durations (Table 3.3), which results 384 sec average flow durations.

3.1.1.6 TCP Traffic versus UDP Traffic

The data analysis of the wired traffic presented in section 3.1.1 showed that more than 78% of the traffic was based on TCP (e.g., HTTP) connections. For this section, we analyzed a link mostly containing UDP-based streaming data. For this, more than 5 GB of data is captured and analyzed. The following statistics are observed: IPv4 (%96.91), UDP (%91.50), and TCP (%4.39). The average package size is 594.45 ± 1 byes (with 99% CI) with an average connection time of 365.24 seconds (Table 3.4). The results are very close to the results presented in section 3.1.1. The difference is mostly due to the difference in the TCP and UDP header sizes.

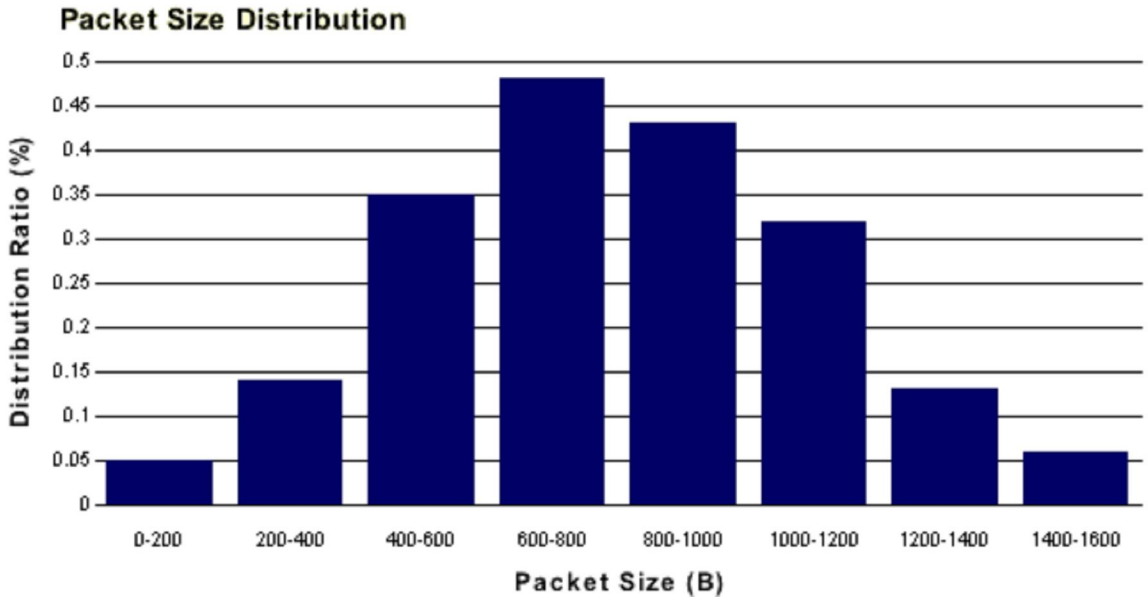

Figure 3.9 Average packet size distributions

3.1.2 Wireless (Omnipeek) Data Analysis

Omnipeek is a special tool designed for wireless traffic monitoring. We have probed two diverse wireless environments (with permission) to find a few average values for the wireless traffic. For the validity of real-traffic monitoring, Omnipeek was run for a long period of time capturing live traffic on the air. We summarized our observations based on the following two batches of data collected from these two diverse environments:

3.1.2.1 Batch 1

For this batch of data, we ran the Omnipeek software system sniffing all 11 channels (North American) of IEEE 802.11 *b/g* band and 28 channels of IEEE 802.11 *a* band in random periods of times. Batch 1 was gathered in an environment with relatively less population of access points and relatively large population of mobile users and clients. It is obvious that the number of supported channels is less and we encountered no radio *a* services being offered. However we noticed limited IEEE 802.11*a* activities, which were generated by the mobile users. The radio *a* activities are related to the Probe Requests that the mobile users with IEEE 802.11*a* capability send on a regular basis. These Probe Requests generated from the mobile devices are meant to try to locate any available APs

advertising services on a specific radio/channel. Since there is no AP running radio *a*, these Probe Requests will not be responded by any Probe Responses. However based on specific scanning algorithm deployed on the mobile devices, these Probe Requests will be sent repetitively to locate the APs if they become available due to location change or AP powering up. Figure 3.10 shows radio *b/g* average activity for this Batch.

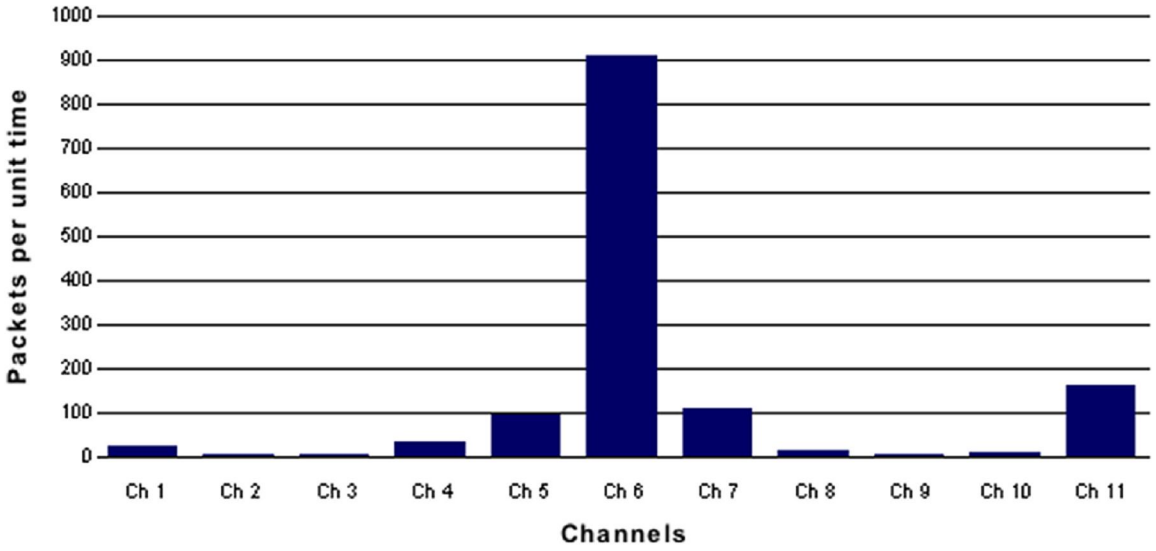

Figure 3.10 Radio *b/g* Activity for batch 1

3.1.2.2 Batch 2

For this batch, the packet monitoring takes place in a highly diverse environment with many serving access points, as well as mobile users and wireless clients. Both IEEE 802.11 *b*/*g* and *a* are actively available. From 28 valid IEEE 802.11 *a* channels (Ch 34 to Ch 165) on only 5 channels (44, 46, 58, 149, and 161), mobile systems were transmitting Probe Requests.

Figure 3.11 shows major activities around channel 6 and interestingly enough it indicates a considerable activities in channel 5 and 7. These two channels are too close to channel 6 and the simultaneous usage of channels 5, 6, and 7 would increase the chance of wireless errors, collisions, and retransmissions. Figures 3.11 and 3.12 show Batch 2's activities in radios *b/g* and *a* respectively.

Table 3.4 Average statistics for wireless and wired flows

	Average Number of bytes/packet	Average Flow Duration (Sec)
Wired	652.588 (TCP Dominant Traffic)	384.24 (TCP Dominant Traffic)
	594.45 (UDP Dominant Traffic)	365.24 (UDP Dominant Traffic)
Wireless	197.206 bytes	412.53 Sec

3.1.2.3 Average Packet Size and Flow Duration

Averaging the entire monitored wireless traces show that the average packet size is 197.206 bytes per packet and the average flow duration is 412.53 sec per flow.

Comparing line and wireless traffic monitor results show a strong correlation of flow duration, roughly in the magnitude of 400 sec, however the average wireless packet size is less than one-third of that of the wired (Table 3.4). This is mainly due to the fact that wired environments can often handle larger packets without experiencing frequent collisions and retransmissions. In wireless environments, however there are relatively more retransmissions and collisions, which enforces lower packet sizes for optimum performance and efficiency.

3.2 QoS-Security Models

In this section, both QoS and security models are presented, which are comprised of QoS and security related cross-layer parameters that deal the quality and security strengths of this system. We discuss these models in terms of QoS only, security, only, and integrated QoS and security models [163].

3.2.1 QoS Model

Cross layer approach is an important concept, which is used in this thesis. Such a deployment serves for both application layer-based QoS and security purposes, which is also discussed in this section.

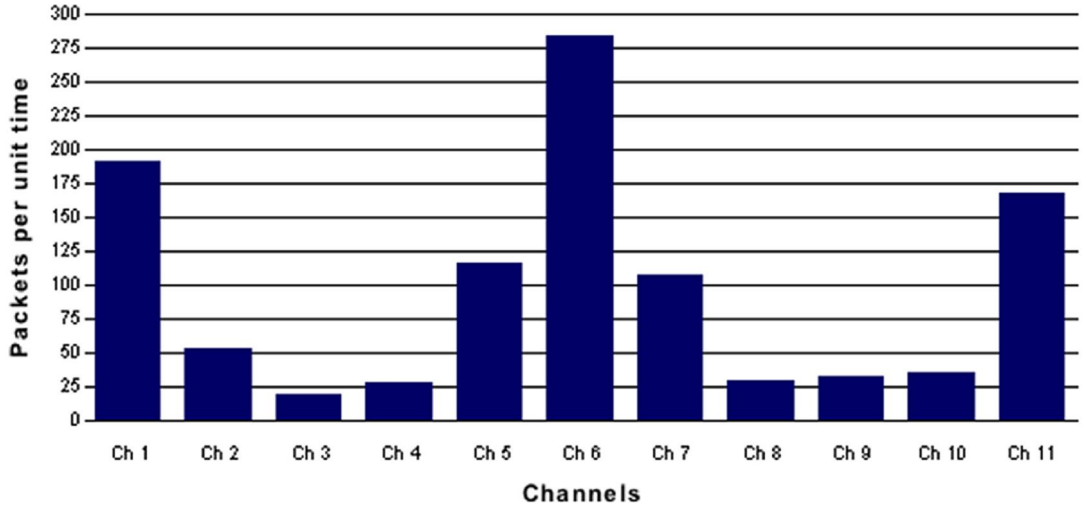

Figure 3.11 Radio *b/g* Activity for batch 2

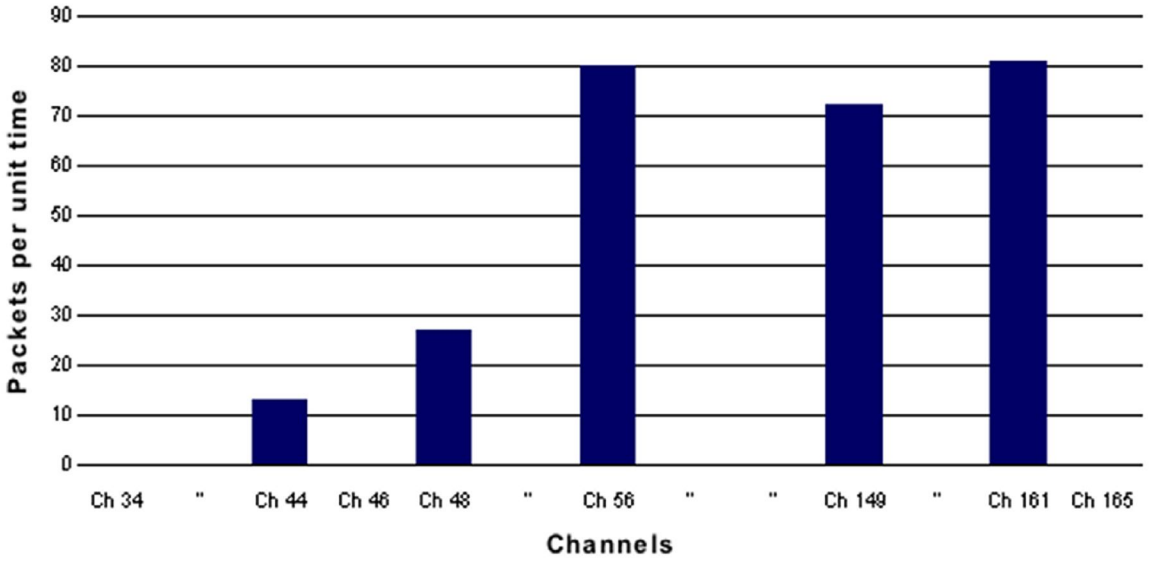

Figure 3.12 Radio *a* Activity for batch 2

Rationale for Deploying a Cross-Layer Design – One of the reasons for deploying a cross-layer design in this system is for supporting QoS and security provided at the application-layer. If there is no cross-layer feedback providing parameters from lower layers, the decisions made at the application layer may not be optimized based on the lower-layer requirements. For instance when the signal strength of the transmitter (a physical layer parameter) drops, the system's throughput capability may decrease, therefore the availability of the knowledge of the signal strength at the application layer

may be useful to adapt the coding scheme to the channel/signal conditions to avoid using high data-rate coding schemes when the signal strength is relatively low. The same argument is valid for other schemes at other layers. Therefore in order to provide the best decisions at the application layer, it would help greatly to have cross-layer information present.

3.2.1.1 Cross-Layer QoS-based Mechanism

The cross layer technique is used to gather parameters from three different layers. These parameters are then imported to the application layer, where they serve as inputs to a few algorithms and mechanisms. The cross-layered parameters are:

Physical Layer - The *RSSI (Received Signal Strength Indicator)* presents the strength of the Access Point (AP) signal. It is represented by dBm. The range is between -94 dBm up to -30 dBm. To present this value range, 6 bits are used.

Signal to Noise Ratio (SNR) shows the relative ratio between the sender signal power to the noise level. This indicator is important because the higher the SNR value, the higher the bandwidth can get. To present this value range, 7 bits are used.

MAC Layer - The *WMM (Wireless Multimedia)* is a three bit MAC layer QoS schemes used in Wi-Fi systems (IEEE 802.11e). Other types of MAC layer QoS metrics can often be translated to WMM. For instance wired MAC layer QoS scheme; IEEE 802.11p and IEEE 802.11Q have exactly 3 bits, which can be translated to WMM one to one.

Network Layer – The *DSCP (Differentiated Services Code Point)* is an 8-bit field in the IP header (network layer). Not all vendors configure all 8 bits, however the first three bits, which are called “Class Selector”, are used by most vendors and they leave the rest of the 5 bits clear. Some vendors easily translate the Class Selector bits into WMM 3 bit values one-by-one. However we take the whole 8 bits. These 8 bits represent 64 different network-layer QoS treatments.

The multilayer QoS treatment has 24 bits (7 + 6 + 3 + 8) and there are 24 other bits gathered for the security side (discussed in section 3.2.2.3.1). These 48 bits are called Cross-Layered Parameters (CLPs).

The CLPs, along with other data are packed in the UDP payload. The CLPs are checked by the both the sender and receiver encoder/decoder to adapt to the best possible coding rates. High quality indications of the CLPs can inform the application layer that the user requires high quality coding. This could also be an indication that the medium is able to handle high throughput requirements. In bad channel conditions, on the other hand, the CLPs values point to low qualities, which in term inform the application layer accordingly to avoid possible uneven drop of performance, congestion, or packet drops.

There are various factors that can affect the measured SNR values. First of all the ratio of the signal strength to the environmental noise level tends to be variable in nature. Therefore such a ratio fluctuates continually and a high noise level in the channel, affects the SNR measurements negatively. As the noise level increases, the SNR reading precision decreases. RF collisions can also increase chances of incorrect SNR readings. In order to have a more reliable SNR reading, the wireless station's distance should be within a few meters to less than 10 meters from the AP. Other APs/stations should be turned off to reduce noise sources and chances for RF collisions. More importantly, SNR readings should be grouped into small ranges for more accurate results. Therefore any values of the SNR, for example from -70 to -75 dBm should be in one group. This way, fluctuations (which are within ± 5 dBm range) have less effect on the precision.

3.2.1.2 Adaptive Forward Error Correction (AFEC)

An adaptive FEC scheme is used at the application layer, which uses the QoS-based parameters gathered through the cross-layer approach. This FEC scheme adapts its performance to the network and channel conditions in such a way that in relatively bad channel conditions, it uses more redundancy, which enabled higher ability to correct possible errors and in relatively good channel conditions, lower redundancy is used.

3.2.2 Security Model

A system's security model is based on two security subsystems: Security algorithms and security protocol. These two security subsystems are introduced in this subsection.

3.2.2.1 Security Protocols

A security protocol is the handshake and protocol flow between two and more end-points, establishing a connection, securely exchanging key-information in an unsecure channel, authenticating each other (mutual authentication), and transmitting the information. The security protocol also defines other aspects of the communication, such as: How often the security algorithms are to be deployed and if they are applied to a stream of data or a block of data. The detail of the security protocol will be discussed in chapter 4. The other security subsystem is the security algorithms, based on Suite-B, which will be discussed in the following subsection.

3.2.2.2 Security Algorithms

The security model of this system, as mentioned, is built around cross-layer Suite-B cryptographic-based non-repudiation system in a P2P scenario between party A (e.g., a doctor) and party B (e.g., a patient or a medical storage device) (Figure 3.13). Party B is capable of decoding the signatures and linking the foregoing communication to the involved party (A) (Figure 3.14). The doctor/patient pair is a good example where this system can be utilized. In such a medical environment, there are many instances where previous communications are to be verified to identify the involved parties and the type of communications between parties. This way no party can deny his/her involvement, neither can deny the involved communication detail. For this, a digital signature is attached to the ongoing communication. This piece of evidence can not be disputed later on by any of the involved people.

Suite-B only specifies a set of cryptographic algorithms. The security protocol specifies the handshakes and exchange messages between end-points. The security protocol is not

only capable of handling privacy, integrity, and non-repudiation mechanisms involving the data exchange between two end-points, it is also responsible to enforce how often these mechanisms should be carried out, based on the minimum security requirements. In the application layer, which is the main layer under consideration, we are dealing with messages. These messages may contain simultaneous transmission of multimedia. To provide privacy, integrity, and non-repudiation, all messages are tagged and hashed, however not every individual message requires signing. This is due to the fact that the process of signing and verification is relatively complex and consumes high power and it has to be done as infrequent as possible. For this, a signature is applied to the hashed value of a message, instead of the message itself. Then the signed hash is encrypted and transmitted to the receiver, where it is verified.

The application layer security protocol (similar to S/MIME) requires a key exchange protocol and a certificate authority to provide key/certificate to both end-points. In a cipher block chaining mode, if a digital signature is used, it will be placed at the end of the last block.

Suite-B cryptographic algorithms can be utilized in various layers. At the transport layer, there is a scheme involving Suite-B Profile for Transport Layer Security (TLS) version 1.2, which was proposed in RFC 5430 (introduced in 3.2.2.3.2). At the transport layer, we are not dealing with messages any more and sessions are being dealt with. A session is created between two end-points using sender and receivers' certificates. A session under TLS is an association between a client and a server, which is created following a handshake protocol.

Suite-B algorithms are being incorporated into IPSec protocol. This has been considered in RFC 4869 (Suite B Cryptographic Suites for IPSec, introduced in 3.2.2.3.1). As long as the validity period of a tunnel is not expired, one certificate suffices for the established tunnel. A general security analysis [163] can be generalized and used to study the security analysis of Suite-B cryptographic algorithms [164]. The detail of the security protocol, the handshakes, and protocol flow will be discussed in chapter 4.

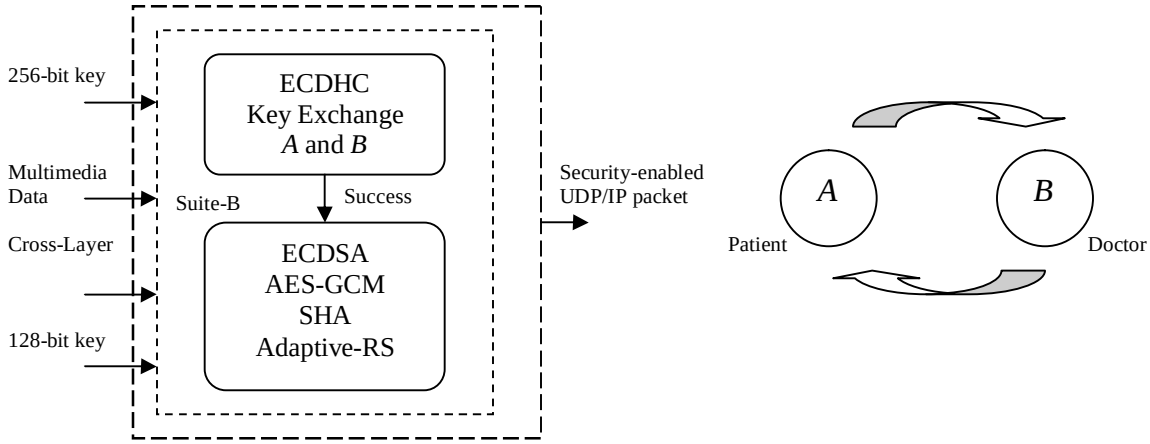

Figure 3.13 Suite-B security schematics adopted in our system

3.2.2.2.1 NSA Suite-B Cryptography

A number of standards including IEEE P1363, ANSI X.9.62, X9.63, FIPS 186.2, specify aspects of Suite-B cryptographic algorithms. As mentioned, Suite-B is a collection of cryptographic algorithms, including; encryption, digital signature, key agreement, and message digest. National Security Agency (NSA) initiated cryptographic interoperability and security requirements issues, including Suite-B to protect both classified and unclassified information and national security systems and is the preferred security options for wireless applications. Suite B's algorithms are [165, 166]:

Symmetric Encryption: AES 128 or 256 key sizes. For authenticated encryption purpose, AES should be used with GCM (Galois/Counter Mode), which is a 128-bit block cipher.

Digital Signatures: This is achieved using Elliptic Curve Digital Signature Algorithm (ECDSA).

Key Agreement: This is achieved using Elliptic Curve Diffie-Hellman (ECDH).

Message Digest: This is done using Secure Hash Algorithm (SHA-256, 384, and 512).

Figure 3.13 shows the high-level end-to-end Suite-B-based security protocol used in this system. Note that the 256-bit key is used by ECDSA-256 and the 128-bit key is used by AES-128.

3.2.2.2.2 Elliptic Curve Diffie-Hellman (ECDH)

The Suite-B key exchange agreement is based on ECDH, which uses elliptic curve-based Diffie-Hellman key agreement scheme. The ECDH protocol allows the two parties to establish a shared secret key over an insecure channel if they have an elliptic curve public-private key pair. This shared secret may be used directly or indirectly as a key. The ECDH scheme does not prevent man-in-the-middle attack because ECDH scheme on its own does not authenticate any of the two parties. For this, an authenticated Diffie-Hellman key agreement protocol is required, which is usually achieved by two parties authenticating themselves to each other by the use of public-key certificates or digital signatures (signed Diffie-Hellman). Through the signed-ECDH adoption, ECDH handshake flows are signed and verified using ECDSA. However this scheme is wasteful of bandwidth. To overcome this issue, a session key is required, which can be derived using a static public key to obtain implicit authentication of the resulting session key, which is the approach used in the MQV (Menezes-Qu-Vanstone) protocol [167]. In this protocol, the assumption is that both parties have long-term static public/private keys. A modified version of the MQV protocol is based on the Elliptic Curve, resulting ECMQV. Both MQV and ECMQV feature an authenticated protocol for key agreement based on Diffie Hellman scheme, providing protection against the man-in-the-middle attack. Here is the protocol handshake flow:

1. *A* has a key pair (P_{ua},P_{ra}), where P_{ua}, is A's public key and P_{ra} is *A*'s private key.
2. Similarly *B* has a key pair of (P_{ub},P_{rb}).
3. A generates a session key pair (X, x) by calculating $X=x*P$, where x is a random integer and P is a point on the Elliptic Curve.
4. *B* follows the same calculation as was performed in 3 and calculates $Y=y*P$.
5. *A* transmits X to *B* and *B* transmits Y to *A*. It is assumed that *A* already has *B*'s public key P_{ub} and *B* already has *A's* public key P_{ua}.
6. *A* calculates S_a (known as the implicit signature) by calculating; $S_aP_{ua} =(x + x' P_{ra})$ mod m, where m is the generating point P's order.
7. *B* calculates $S_bP_{ub}=(y+ y P_{rb})$
8. *Now b*oth *A* and *B* have calculated a shared secret key; S_k.

9. $S_k = CF * SP_{ua} (Y + y' SP_{ub}) = CF * SP_{ub} (X + x' P_{ua})$, where CF is a co-factor and x' and y' respectively represent the first L bits of the first X and Y pair component and where $L = (\log_m 2 + 1)/2$

The problem with the deployment of ECMQV is that it is not part of Suite-B, therefore since the approach adopted in this thesis is based on Suite-B, therefore ECMQV is not used. For this we assume both parties have authenticated themselves to each other using public-key certificates (Figure 3.17).

The application layer Suite-B key-exchange protocol will be further discussed in section 4.2.6.1.

In this section, we will discuss the key establishment protocol (based on ECDH mechanism) further more:

Key establishment protocol – Assuming Alice wishes to establish a shared key communication with Bob in a channel prone to third party eavesdropping. The initial domain parameters (p, a, b, G, n, h), which are defined as: p = a prime number, a and b = elliptic curve constants, G and $n = G$ is a generator with an order that is the smallest non-negative number n such that $nG = O$, must be prime. Finally h is $|E|/n$, where n is the size of a subgroup of E. These domain parameters must be agreed upon. Each party must also have a key pair suitable for elliptic curve cryptography, comprised of a private key d (which is selected as a random integer from the interval $[1,n - 1]$) and a public key Q (where $Q = dG$). If trusted or provided via a certificate, public keys will be static.

Assuming Alice has a key pair of (d_A,Q_A) and Bob has a key pair of (d_B,Q_B). Before any data can be exchanged between these the two parties, each of them requires having the other side's public key, therefore key exchange must occur. Then Alice calculates $(x_k,y_k) = d_AQ_B$ and Bob calculates $k = d_BQ_A$. Assuming x and y are coordinates of a point, then the shared key is x_k and the number computed by Alice and Bob will be equal since: $d_Bd_AG = d_BQ_A = d_AQ_B = d_Ad_BG$.

Since solving the Elliptic Curve Discrete Logarithm Problem for a third party without knowing the private key is a very difficult computational task with the current computational power, therefore the protocol is secure.

Reference [168] presents formal security analysis and proof for the security strength of Diffie-Hellman (DH) and DH-based protocols. This can be generalized to ECDH, which is also based on DH.

3.2.2.2.3 AES-GCM versus AES-CCM

Advanced Encryption Standard (AES) with either 128 or 256 bits key size is used for the encryption in Suite-B cryptographic algorithms, mixed with the Galois-Counter Mode (GCM) block cipher mode for the authentication purpose. The AES-GCM algorithm is an extension of the AES-CCM (Counter with CBC-MAC), which used a 128-bit block cipher authentication scheme. AES-CCM has four inputs [169]: a nonce, an AES key, a plaintext, and an optional Additional Authenticated Data (AAD), generating two outputs; a Message Authentication Code (MAC), (also known as the Authentication Tag or "AT") and a ciphertext.

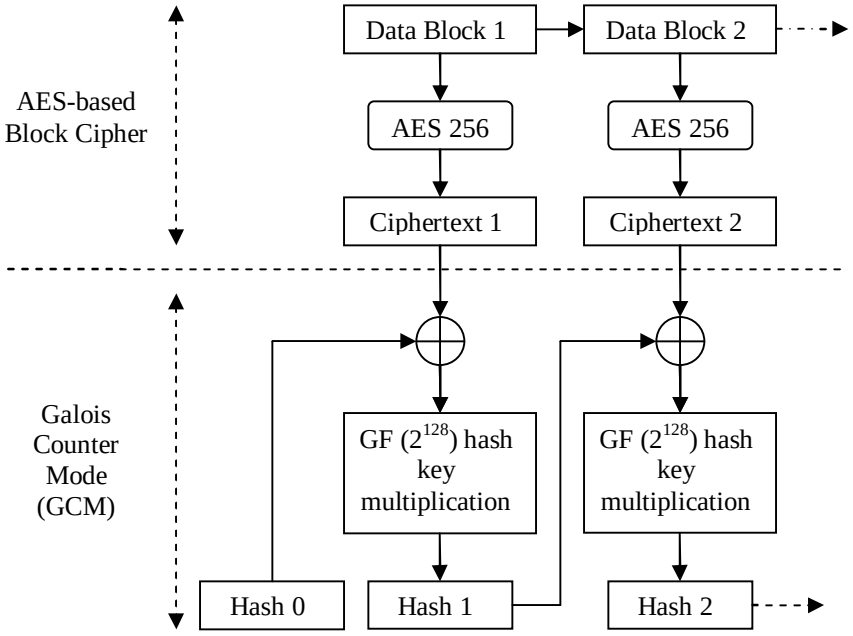

Figure 3.14 AES-GCM encryption/authentication schemes

The AES-GCM is a 128-bit generic authenticated encryption block cipher mode. AES-GCM also has four inputs [169]: an initialization vector (IV), an AES key, a plaintext content, and optional AAD field, generating two outputs: a MAC (AT) and a ciphertext.

Both of these schemes are very similar as they both perform authenticated encryption and accept AAD, however the GCM algorithm includes an authenticated encryption with one pass over the data, which allows a much higher throughput compared to that of CCM, which requires two passes. AES-GCM is shown in Figure 3.14 (adapted from [170]).

The proof of security for AES-GCM has been studied in literature [171, 172]. AES is resistant against linear and differential cryptanalysis and when combined with GCM, the theorems provide proof of security. Reference [172] contains a combination of Lemmas and Theorems, indicating the security strengths of AES-GCM and GCM algorithms.

3.2.2.2.4 Elliptic Curve Digital Signature Algorithm (ECDSA)

Another Suite-B component is the ECDSA scheme, which uses Elliptic Curve Cryptography (ECC) in the structure of the digital signature analogous to DSA. With equal cryptographic power, ECC-based keys can be smaller than RSA-based keys. For example an ECC-based scheme with 224 bit key size is cryptographically comparable to both RSA and DSA with 2048 bit key sizes. ECC-based scheme with 256 bit key size is comparable to RSA- and DSA-3072. However it has been shown in reference [173] that ECC-based scheme with 224 bits key operates more efficiently compared to their the RSA-2048 and DSA-2048 equivalent schemes, which makes it a better fit for wireless applications [173]. The signature sizes for ECDSA-160, ECDSA-224, and ECDSA-256, are 320, 448, and 512 bits respectively, in which the first three correspond to signature sizes of RSA- and DSA-1024, 2048, and 3072 requiring 1024, 2048, and 3072 bits respectively. These comparisons show how the ECDSA scheme outperforms RSA and DSA schemes from overhead point of view. Therefore ECDSA-256 is used in this thesis.

Table 3.5 SHA-2 family properties

SHA Scheme	Security (bits)	Message Size (bits)	Block Size (bits)	Message Digest Size (bits)
SHA-256	128	$< 2^{64}$	512	256
SHA-384	192	$< 2^{128}$	1024	384
SHA-512	256	$< 2^{128}$	1024	512

The signature key generation, signature generation, and signature verification algorithms are briefly described in Appendix A [36].

Reference [174] indicates the security strength of ECDSA as the best known algorithm.

3.2.2.2.5 Secure Hash Algorithm

The Suite-B hash algorithm is based on either SHA-256, SHA-384 or SHA-512 algorithm. Secure Hash Algorithm is part of the Federal Information Processing Standards Publication 180-2 (FIPS 180-2) [175]. Table 3.5 (adapted from [175]) shows SHA-256, 384, and 512 properties. References [176, 177] examine the security strength of SHA-2 family.

3.2.2.2.6 Suite-B Cryptographic Layered Applications

The Suite-B algorithms can be incorporated at any layer, depending on the application and criteria. Here are some examples of such layered-based deployments:

3.2.2.2.7 Suite-B for Transport Layer Security (TLS)

Transport Layer Security (TLS), considered in a number of RFCs (i.e., RFCs 3268, 4346, 4366, 4492, 5246, and 5430), is a protocol, which provides data integrity and privacy for two communication applications at the transport layer. The most current version in the TLS version 1.2 [178] includes a number of revisions compared to the TLS version 1.1.

RFC 5430 [179] proposes a Suite-B profile-based TLS system, which makes use of the Suite-B algorithms (e.g., encryption, digital signature, message digest, etc.). RFC 5430 mandates backward compatibility, which requires that both sender and receiver to either deploy Suite-B TLS or fall back to a non-Suite-B TLS mode. For this we support both sender and receiver to operate using the following profile: AES (128)-GCM, ECDSA-256, and SHA-(256, 384, and 512).

3.2.2.2.8 Suite-B for IPSec

Suite-B IPSec has been proposed in RFC 4869 [180] providing integrity and confidentiality protection for the ESP mode using 128-bit AES-GCM. This enhances the protections provided by IPSec.

3.2.2.2.9 Cross Layer Security

In this scheme, a number of security-related parameters are gathered from various layers, accompanied with other gathered QoS-related parameters, which are all used in the design and integration of an application layer security-based module. The following security parameters are gathered:

IPSec/VPN capability: The availability of IPSec and VPN tunneling is an important factor in the system's security portfolio. To present all variations in the encryption schemes and operational modes (e.g., AH, ESP, AES, 3DES, etc.), which includes Suite-B IPSec algorithms, 8 bits are used.

MAC layer security options: MAC layer security is a readily available security mechanism built in most wireless access technologies (e.g., WEP, WPA, WPA 2, etc.). To show the availability of the MAC layer security (e.g., scheme type, number of keys, etc.), 8 bits are used.

Table 3.6 Reed-Solomon Adaptive Codes

	R-S (*n*, *k*)	Parity bits	Bits/symbol	Symbol Error Correction (#)
Code 1	(255, 251)	4	8	2
Code 2	(255, 247)	8	8	4
Code 3	(255, 239)	16	8	8
Code 4	(255, 223)	32	8	16

Transport layer security options: Security schemes available at the transport layer, include; TLS (Transport Layer Security) and SSL (Secure Socket Layer), and Suite-B TLS algorithm. To show the availability of the transport Layer security, 8 bits are used.

The availability of security at various layers indicates that the applications running on top of these layers may incur various delays to accommodate security requirements. Therefore such availabilities are informed to the application layer to adjust the performance accordingly.

3.2.2.2.10 Cross-layer-based encryption/FEC mechanism

This scheme integrates AES-128 bit – Galois-Counter Mode (AES-GCM) at the application layer with an adaptive Reed-Solomon code. AES-GCM provides encryption and message authentication mechanisms. We choose an 8-bit symbol code with 255 symbols per block. The *k* design parameter is set to be adaptive with a cross-layer feedback inputs from various layers. The RS-codes (in Table 3.6) are being considered.

Both the quality of the multimedia encoder and the adaptability of the encryption/FEC schemes, depend on the cross-layer feedback received from various layer at the application layer. Table 3.6 presents the effects of the cross-layer feedback to the AFEC scheme. The application-layer Encryption/FEC system is shown in Figure 3.15.

According to Table 3.7, higher values of RSSI, SNR, WMM, and DSCP are indications that the communication wireless channel can handle higher throughput with lower error rates and the transmitting device has QoS-enabled markings, which can increase the

probability of better traffic handling. For instance, the effect of higher RSSI can result an increase in the encoding rate and an increase in the *k* factor in the adaptive FEC scheme. On the other side, the availability of security options (e.g., WPA, WPA2, IPSec, TLS, etc.) are indications that the system cannot operate with optimal throughput figures due to the delays incurred by various security algorithms.

3.2.3 UDP Payload Discussion

Once keys are agreed upon using an authenticated ECDH algorithm, encrypted traffic can flow between two end-points. In the considered system, UDP is the transport protocol conveying security and QoS-enabled traffic between the end-points. Therefore in this section, the UDP payload is going to be constructed based on the cross-layer information, digital signature, and the multimedia (text, voice, and/or video) information.

3.2.3.1 Data Transmission Methods

According to Figure 3.16, the UDP payload includes: Cross-layer, digital signature, hash, and A to B communication data, which are further specified in Table 3.8. The functional chart of the UDP payload construction is shown in Figure 3.17. It has to be noted that there are two hash functions; one used exclusively in the digital signature algorithm (ECDSA) and second one is used to create a digest of the UDP payload. The discussion over the authenticated key exchange protocol (based on ECDH) is presented in 3.2.2.2.2 and 4.2.6.1 sections.

From the security point of view, there are four methods to transport the UDP packet information from Party A to Party B.

3.2.3.1.1 Method 1

In this method we propose an application layer Suite-B cryptographic profile covering necessary data and functional entities (i.e., hashing, digital signature, cross-layer information, and communication codeword data), which are being pushed to the transport

layer and being packed in the UDP payload. Except for the communication data, the rest of the UDP packet contents are transmitted in the clear without any encryption. Therefore the communication data only has privacy and the rest of the functional entities are left without privacy.

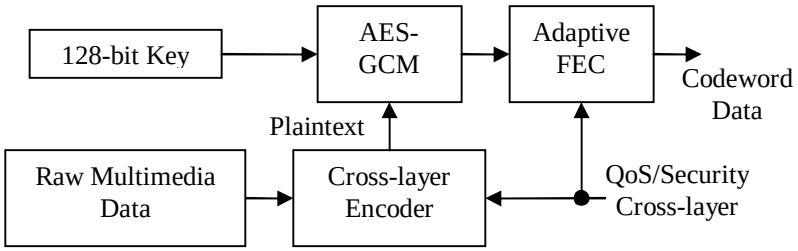

Figure 3.15 Adaptive/encryption FEC schemes

Table 3.7 Cross-layer feedback to the application-layer

Cross-Layer Parameter	Layer	Effect on the Encoder	Effect on FEC Scheme Selection
RSSI	PHY	Encoder rate increase	Increase of *k*
SNR	PHY	Encoder rate increase	Increase of *k*
WMM	MAC	Encoder rate increase	Increase of *k*
DSCP	Network	Encoder rate increase	Increase of *k*
IPSec/VPN/Suite-B-IPSec	Network	Encoder rate decrease	Decrease of *k*
WPA2-AES	MAC	Encoder rate decrease	Decrease of *k*
TLS-Suit-B-TLS	Transport	Encoder rate decrease	Decrease of *k*

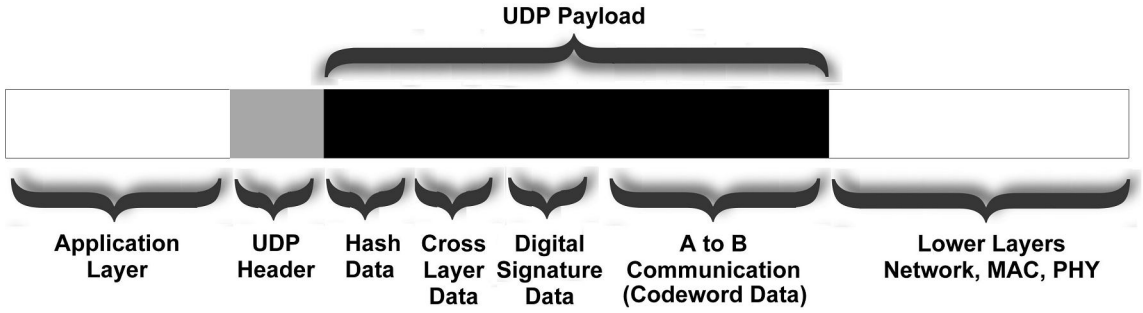

Figure 3.16 UDP traffic payload from A to B

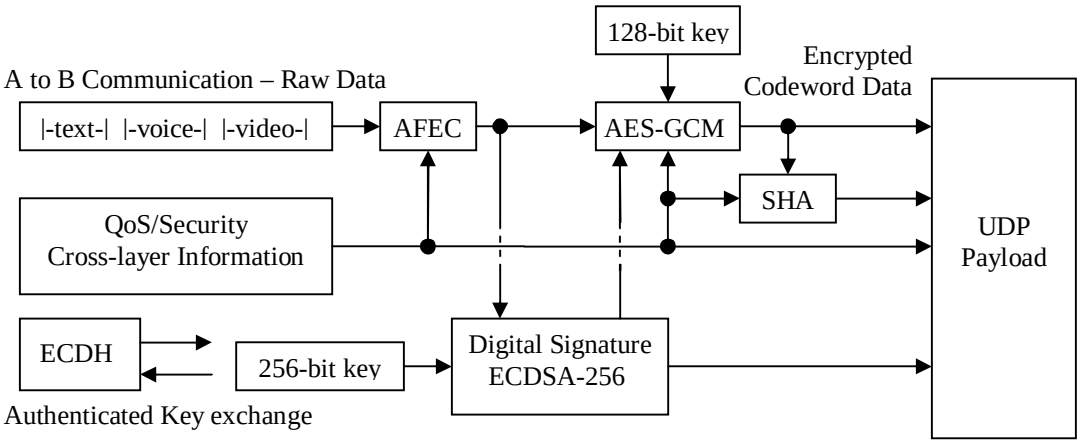

Figure 3.17 UDP traffic payload construction

Table 3.8 UDP payload details

UDP Payload Functional Entity	Details
Cross-layer information	CLP: 48 bits
Digital Signature (ECDSA-256)	512 bits
Communication data codewords	Variable, depending on the input data
Hash	SHA-256: 256 bits
	SHA-384: 384 bits
	SHA-512: 512 bits

Table 3.9 Security profile for method 1

UDP Payload/Port Entities	Privacy	Integrity
Cross-layer information	No	Yes
Communication data	Yes	Yes
UDP Source/Destination Ports	No	No
IP Source/Destination Addresses	No	No

Table 3.10 Security profile for method 2

UDP Payload/Port Entities	Privacy	Integrity
Cross-layer information	Yes	Yes
Communication data	Yes	Yes
UDP Source/Destination Ports	Yes	Yes
IP Source/Destination Addresses	No	No

According to Table 3.9, cross-layer information bits are packed into the UDP payload in the clear, therefore no privacy is provided. However since the entire UDP payload is being hashed, including the cross-layer information bits, therefore integrity is provided. The 256-bit key generator is used to feed the digital signature (ECDSA-256) and a 128-bit key generator is used for the AES-128 mechanisms.

The communication data bits are used as the data input to the AES-GCM-AFEC entity. This functional entity and hashing provide both privacy and integrity for the communication data.

Both UDP and IP source/destination port/address information are all sent in the clear, therefore neither privacy nor integrity are provided for them. Therefore if no underlying (Network, MAC, and/or Physical layer) security is provided, an illegitimate user can have access to the packed information and is able to make unwanted changes.

3.2.3.1.2 Method 2

In this method, Suite-B security algorithms are applied to the Transport Layer Security (TLS) [179]. In this method, the only mechanism provided at the application layer is the FEC. The cross-layer information bits are exported in the clear from the application layer to the transport layer.

This method provides confidentiality and integrity to the UDP payload, as well as the source/destination port numbers. Table 3.10 summarizes the provided security functions.

The two other methods are discussed with network and MAC layer security options.

3.2.3.1.3 Method 3

In this method, Suite-B security algorithms are present for IPSec [180]. In this method, the only mechanism provided at the application layer is the FEC. There is no security mechanism provided at the transport layer. The cross-layer information bits are exported

from the application layer to the transport layer and from the transport layer to the network layer, all in the clear. At the network layer, Suite-B mechanisms provide confidentiality and integrity to the entire IP information, including; UDP payload, UDP source/destination port numbers, and IP source/destination addresses. Table 3.11 summarizes the provided security functions.

Table 3.11 Security profile for method 3

IPSec Payload Functional Entity	Privacy	Integrity
Entire UDP Payload (FEC and Communication Data)	Yes	Yes
Cross-layer information	Yes	Yes
UDP Source/Destination Ports	Yes	Yes
IP Source/Destination Addresses	Yes	Yes

3.2.3.1.4 Method 4

Suite-B cryptographic algorithms were discussed at application, transport, and network layers. However there is no standard specification for Suite-B at the MAC layer. Therefore we combine the cryptographic measures of Method 1 and the MAC layer security scheme available at the MAC layer. This method is based on MAC layer security options, therefore both parties have to be on the same WLAN unless these packets are routed using additional network layer facilities (i.e., IPSec). For this method we will consider WPA-AES.

3.2.4 Functions at the Receiving-End

In this section we assume the traffic has been transmitted successfully to Party B, where appropriate mechanisms (reversed actions) are presented to decode the information accordingly. These reverse actions include: Decapsulation, decipher, reverse hash, signature verification, and multimedia content decode. The reverse functions will be discussed in the next chapter where results are presented.

3.3 Conclusion

In this chapter, the mechanisms concerning both QoS and security models were presented and studied. We studied the deployment of Suite-B cryptographic algorithms in application, transport, and network layers.

3.3.1 Cross-Layer QoS Parameters

Cross layer information bits used in this system are comprised of four fields (24 bits): *RSSI (Received Signal Strength Indicator), Signal to Noise Ratio (SNR), WMM (Wireless Multimedia) DSCP (Differentiated Services Code Point).* The information bits in these fields are adjustable by the main or through a third-party application, therefore the application can change these values for a higher or lower QoS handling, depending on the channel conditions, where setting these values to high can also be an indication for a good channel quality, which can accommodate higher throughputs with higher than average number of bytes per packet. As indicated in section 3.1, the average number of bytes in a wireless packet was close to 200 bytes and depending on the channel condition, in good channel conditions, higher than 200 bytes of packet can be transmitted without encountering degradations (e.g., retransmissions, packet drops, packet loss, etc.). However for a relatively bad channel quality, the multimedia encoder can adapt to lower encoding quality schemes to reduce the number of bytes per packet transmission.

3.3.1.1 Bounded Delay

A thorough end-to-end analysis presents a bounded delay figures for the multimedia communication in the presence of security elements. These delay figures, which are discussed in chapter 4 show the minimum and maximum (bounded) delay figures, which play a major role in the VoIP-based communication performance.

According to the literature, a 150 msec end-to-end delay is the maximum tolerable delay figure for VoIP communications. Through analytical, as well as experimental

results, it would be shown that the system under consideration will perform well within the permitted delay range even for the worst case scenario. The worst case scenario features a payload with the encapsulation of highest delay intensive elements.

3.3.1.2 Minimum Throughput

Guaranteed minimum throughput is a vital QoS parameter that is required for a high quality VoIP communication. Throughput and delay figures are usually conversely proportional to one another. Therefore a maximum bounded end-to-end delay and a bounded overhead are often vital requirements for a minimum throughout. Through experimental results presented in chapter 4 it will be shown that the performance of the system under consideration will perform well from the delay and throughput points of views even for the worst case scenario.

3.3.1.3 Bounded Overhead

Overhead and delay are usually directly proportional to one another and overhead and throughput are mostly inversely proportional. Through analytical, as well as experimental results it would be shown that the system under consideration will perform well within the permitted delay range even for the worst case scenario. The worst case scenario features a payload with the encapsulation of lengthiest multimedia/security elements.

Table 3.12 Layered and cross-layer security feedback

Layer	Cross-Layer Parameter Export to Application Layer / Scheme Availability	Schemes
Application	Application-Suite-B	AES (128)-GCM-FEC SHA-(256, 384, 512) ECDH - ECDSA (256)
Transport	TLS-Suite-B	Suite-B-TLS
Network	IPSec-Suite-B	Suite-B-IPSec
MAC	WPA2-AES + Application-Suite-B	-

3.3.2 Cross-Layer Security Parameters

Half of the cross-layer information bits (24 bits) are used to indicate the availability of security mechanisms in various layers, including: IPSec/VPN/Suite-B-IPSec at the network layer, WPA2 at the MAC layer, and TLS-Suit-B-TLS at the transport layer. A security profile is discussed, based on the Suite-B cryptography. This security profile, which is applicable to various layers, covers necessary data and functional entities (e.g., hashing, digital signature, cross-layer information, and communication codeword data) and is the preferred security mechanisms for protected wireless applications. Table 3.12 sums all layered security model parameters.

The next chapter will contain thorough analytical and experimental results.

Chapter 4
Analytical and Experimental Results

In this chapter, we present the analytical and experimental results based on the QoS and security models that were presented in chapter 3. We will show that the performance parameters (e.g., end-to-end delays, packet overheads, etc.) are within acceptable ranges.

In chapter 3, QoS and security models were introduced and discussed. The security model included the deployed security algorithms (based on Suite-B) and the security protocols in multilayer scenarios. The security algorithms included: encryption and privacy, integrity, and non-repudiation. The experimental results are based on the Crypto++ Library 5.6.0 cryptographic algorithms [3] running on an Intel® T2500-2GHz CPU and a 2 GB RAM on a Windows XP Service Pack 2 platform. The presence of these algorithms and the added overhead, introduce additional delay and cause an overhead increase, which should be considered in the performance evaluation in this chapter. The QoS model, involves a multimedia-based communication system offering security-related services. The inclusion of both security and QoS parameters increases the overhead in terms of the number of bytes per packet and delay. As mentioned in Chapter 3, digital signatures are not required on every message/packet, however we will consider the worst case scenario where all security algorithms are performed while the maximum amount of overhead is incurred in terms of payload bytes, which are transmitted and we show that this system is able to handle such a load and perform well in terms of acceptable delay/payload size values.

Figure 4.1 shows the functional entities and algorithms in Application, Transport, Network, MAC, and Physical layers. The detail of the security protocol will be discussed later in this section.

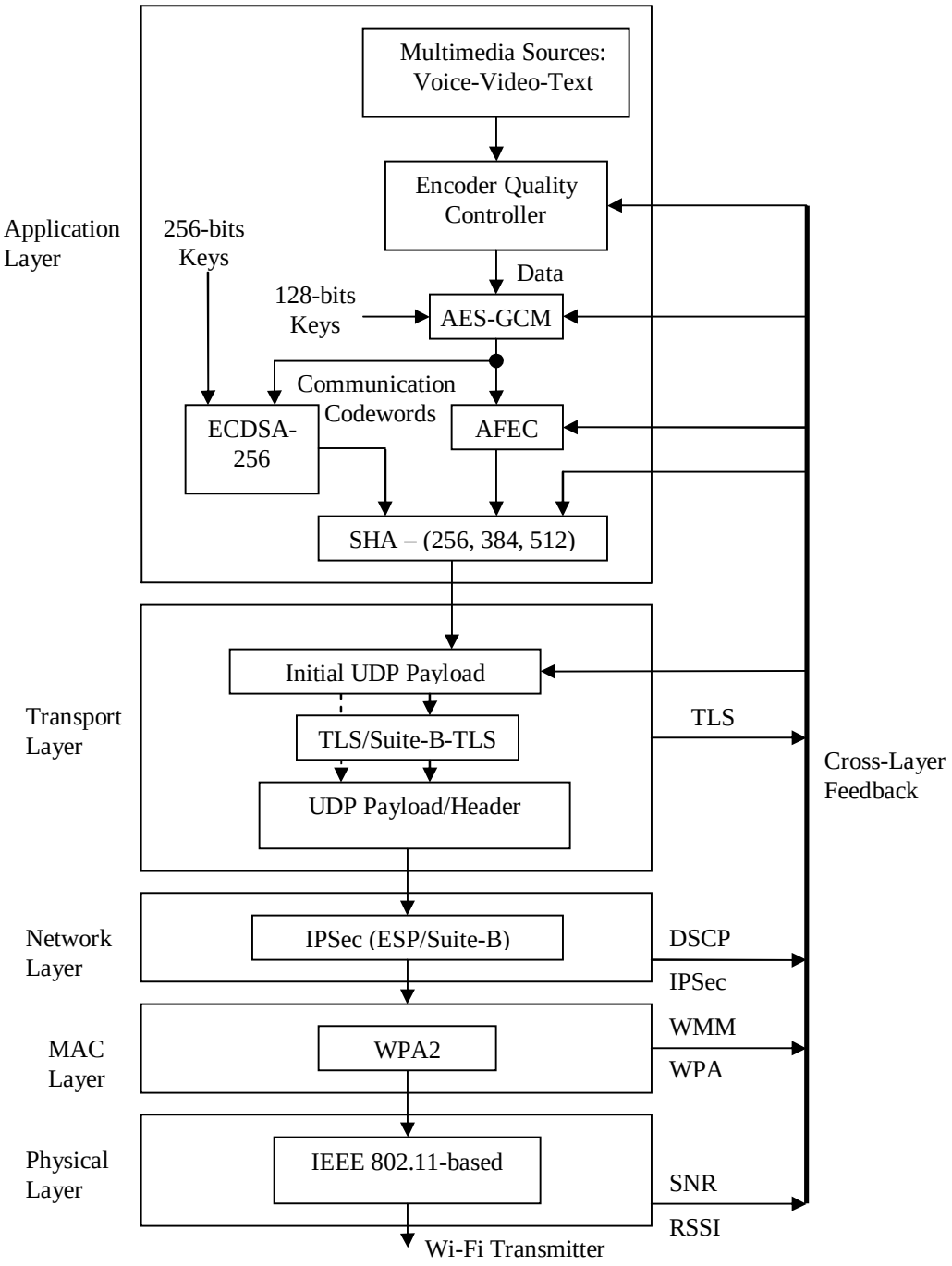

Figure 4.1 Cross-layer QoS/security models

4.1 Security/QoS Model Discussions

In this section, we discuss security/QoS performance details, including processing delays, overheads, and other comparison measures.

4.1.1 Security Model Discussions

We introduced the security model in chapter 3. In this section, the security model will be discussed in more detail. As mentioned, the security model is based on the security algorithms and the security protocols. In this section, the security algorithms are discussed, which are based on Suite-B algorithms adopted throughout various layers. However we first need to discuss the relevant scenarios.

4.1.1.1 ECDH

The key exchange algorithm is the first step before party A and B transmit data to one another. This is done through the ECDH scheme, which is based on the scheme described in subsection 3.2.2.2.1.

For evaluating the performance of ECDH protocol, the Crypto++ Library 5.6.0 [3] is used, which features a program that computes the time required for “Alice” and “Bob” to exchange key information based on Elliptic Curve Cryptography (ECC). Table 4.1 shows the ECDH-256 public key operation delay cost based on the mentioned platform.

Table 4.1 ECDH performance measures

Scheme	Key Pair Generation	Key Agreement
ECDH-256	2.62 (msec)	2.58 (msec)

4.1.1.2 ECDSA – 256

In this thesis, ECDSA-256 is being used for the digital signature purpose, which has the same security strength compared to DSA-3072 and RSA-3072, with lower overhead.

As mentioned in Chapter 3, ECDSA-256 has a signature size of 512 bits (64 bytes). The signature generation and verification delays are mentioned in Table 4.2 [3, 181, 182], based on the mentioned platform.

Table 4.2 ECDSA-256 Performance Measures

ECDSA-256	Overhead/Delay
Signature Size	512 bits (64 bytes)
Signature Delay	2.63 msec
Signature Verification	3.89 msec

4.1.1.3 AES – GCM

The AES-GCM (Galois/Counter Mode) (Figure 4.2, adapted from [183]), is an authenticated encryption block cipher mode, as mentioned in chapter 3. Though AES is a part of Suite-B algorithms, however AES-GCM has been considered in the Suite-B framework [184, 185], which have not been standardized yet.

GCM combines a well-known encryption counter mode with the Galois authentication mode. The key feature of GCM is that parallel multiplications are easily computed in GCM, thus achieving very high throughput with a fast chaining mode authentication technique. Figure 4.2 (adapted from [183]) shows the GCM encryption and decryption modules. As mentioned, AES-GCM is suitable for high throughput communications. Therefore the overheads and delays in its implementation are relatively low [186].

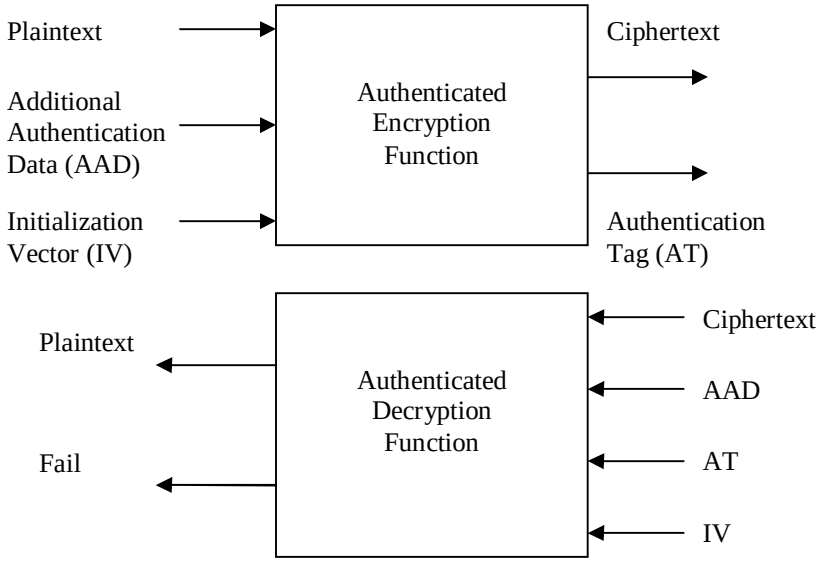

Figure 4.2 GCM encoding and decoding

Table 4.3 The AES-GCM performance

Encryption Scheme	Block Size	Performance	Delay
AES-GCM	16 bytes (128 bits)	18 Cycles per byte 106 MiB per second	9.44 nsec/byte

The multiplication complexity of GCM encryption algorithm involves a *GF* (2^{128}) field. The decryption algorithm offers the same amount of delay as both algorithms are based on the same structure.

The GCM's main overhead is due to the authentication tag, which can be 128, 120, 112, 104, 96, 64, and 32 bits, depending on type of application in use. The ciphertext is the same length as the plaintext, therefore there is no overhead in the ciphertext. For the highest security strength, 128-bit tag length should be used.

Though AES-128 is based on a symmetric structure, however the decryption algorithm of AES is slightly slower than the encryption algorithm, due to the complexity of the inverse matrix calculations involved [187].

The performance of the AES-GCM algorithm is based on Crypto++ Library 5.6.0 [3] C++ code running on an Intel Core 2 CPU at 2 GHz is mentioned in Table 4.3. Note that MiB stands for MeBiByte or Mega Binary Byte and 1 MiB is 1,048,576 bytes.

4.1.1.4 Adaptive FEC based on Reed-Solomon (RS)

The FEC mechanism features the inclusion of redundant bits into the input data for error correction purposes. The FEC scheme used in our approach (Table 4.4) uses four different values of *k*. All four cases have the same output of 255 bytes and depending on the value of *k*, with variable input data length. However since the input of the FEC algorithm is fed from the AES-GCM algorithm, we need to change the contents of Table 4.4 in a way to show the input data length is fixed at 255 bytes. This way the output data

lengths will become variable. This is shown in Table 4.5., which shows that, for example, if code 2 is used, we need to use 264 bytes (255 bytes data + 9 redundant bytes) for every 255 input data bytes and the relative overhead compared to (255, 255) scenario (without R-S coding) is 4%.

The performance of R-S codes depends on the minimum correction capability (*T*) and the codeword length (*W*). It follows the following equation (adapted from [188]):

$$Error\ Decoding\ Processing\ Delay = 9 \times T + \frac{W}{2} + 14$$

The latency for various values of *T* and *W* would require an average cycle count of 6359, which corresponds to maximum 3.25 μsec based on our platform. According to Table 4.6, different delay/overhead figures for various FEC schemes are shown based on the same platform.

Table 4.4 R-S FEC scheme with fixed output

Code Number	R-S (*n*, *k*)
Code 1	(255, 251)
Code 2	(255, 247)
Code 3	(255, 239)
Code 4	(255, 223)

Table 4.5 R-S FEC scheme with fixed input

Code Number	R-S (*n*, *k*)	Relative Overhead
Code 1	(260, 255)	2%
Code 2	(264, 255)	4%
Code 3	(273, 255)	7%
Code 4	(292, 255)	15%

Table 4.6 The adaptive Reed-Solomon (ARS) performance

ARS Scheme	Delay	Overhead
RS (260, 255)	11.30 nsec/byte	5 bytes
RS (264, 255)	11.47 nsec/byte	9 bytes
RS (273, 255)	11.82 nsec/byte	18 bytes
RS (292, 255)	12.69 nsec/byte	37 bytes

Table 4.7 SHA algorithms comparisons

Hash Function	Number of Rounds	Hash Size (Bytes)	Block Size (Bytes)	Processing Delay (cycles/byte)	Total Delay (CPU 2 GHz)
SHA-256	64	32	64	16	8.34 nsec/byte
SHA-384	80	48	128	17.2	8.97 nsec/byte
SHA-512	80	64	128	17.8	9.28 nsec/byte

4.1.1.5 SHA – (256, 384, 512)

Data integrity is offered through the deployment of Secure Hash Algorithm (SHA) functions. In Chapter 1 it was mentioned that SHA-1 may not be secure enough therefore other variations of SHA-2 family (i.e., SHA-256, -384, and -512) should be used. It is also important to note that SHA-2 family members have been mandated in Suite-B.

The number of arithmetic calculations required to perform SHA hash functions varies according to the processor and the running application. For instance, based on the same platform, SHA-256 requires 16 cycles per byte to perform secure hashing algorithm. Table 4.7 captures the performance of the SHA-2 family [189].

4.1.2 Multimedia Communication

The following scenarios are considered in this thesis's security profile:

1. Method 1: In this method, Suite-B algorithms (ECDH, ECDSA, AES, and SHA) are performed at the application. No security options at the Network and MAC layers are available.

2. Method 2: In this method, Suite-B at the transport layer is used.
3. Method 3 In this method, Suite-B at the network layer is used based on IPSec.
4. Method 4: In this method, core UDP payload (mentioned in Method 1) is used with a MAC layer encryption technique (i.e., WPA-AES).

Therefore it is essential to study security features applied to the Application, Transport, Network and MAC layers. First we start from application layer security entities. At the application layer, the following schemes are dealt with: AES-128-GCM, Adaptive Reed-Solomon; ARS (255, *k*), ECDSA-256, and SHA (256, 384, and 512), therefore we will consider these schemes in a later subsection.

The main communication data between two parties are in form of text, video, and voice, which are discussed in this subsection.

*Text***:** This can be non-interactive (email) or interactive (text-chat). In this case, keyboard key strokes are captures and placed in the UDP payload along with CLPs.

Interactive typing requires a very low bandwidth. An average professional typist reach 50 to 70 Word per Minute (WPM), usually less than 100 WPM. Therefore an interactive (two way) typing session transmitting 100+100 = 200 WPM (for half duplex PHY/MAC services) will require about 4 words per second, which requires less than 40 bps. In terms of the UDP payload per packet, this translates to only 1 byte of information on average.

Voice: This is usually an interactive VoIP call or a normal voice-chat. In both cases, voices are captured and coded into a voice codec. For the best case scenario where the signal/channel conditions are very good, G.711 codec is used to encode the voice information and pack them into the UDP payload. G.711 packets run at 64 kbps. A typical G.711 packet runs for 30 msec with a payload of 240 bytes; (240 bytes X 8bits) / 30 msec = 64 kbps.

For the worst case scenario where signal/channel conditions are poor, according to Table A.2, the data rate of G.723.1 can be as low as 5.3 kbps, with a 7.5 msec look-ahead and 30 msec inter-frame delay and 20 bytes per packet. Therefore: 20 bytes x 8 bits / 30 msec = 5.3 kbps. The summary of the voice codec delays and overheads are mentioned in Table 4.8.

Video: Video can also be an interactive videoconferencing communication between two (point-to-point) or more parties (multicasting). This is often accompanied by voice and text. In this case, for the best case scenario, one of the well-known video codecs; H.262, H.263, or H.264 could be used. All three mentioned codec could be running at the minimum rate of 64 kbps.

H.264, which is a newer and preferred codec compared H.263 and H.262, is able to generate video information starting from 15 fps (frame per seconds) at 177 X 144 (QCIF) resolution. H.264 (similar to MPEG-4) often uses RTP (Real Time Transport Protocol [190]) for video data transport. H.264 requires approximately 6500 bits per frame [191, 192] with a minimum 10 msec of frame size [193], in which the result is 65000 bits per second or 64 kbps. There are 15 frames per second (fps) with 66 msec inter-frame delays, which will results 15 packets per second with each containing 546 bytes of H.264 encoded video information (Table A.2).

For the worst case scenario, H.263 with 120 x 90 frame size and 10 fps with 100 msec inter frame delay, running at 24 kbps, are used. In this case, 307 bytes are sent per packet. Table 4.9 summarizes the video codec data (based on our platform).

Table 4.8 Voice codecs datasheet

Codec	Minimum Bit Rate	Channel/Signal Condition Usage	Encode Time	Packet Inter-Frame Delay	Bytes per Packet
G.711	64 kbps	Best	0.125 ms	30 ms	240 bytes
G.723.1	5.3 kbps	Worst	7.5 ms	30 ms	20 bytes

Table 4.9 Video codecs datasheet

Codec Type	Minimum Bit Rate	Resolution (pixels) Frame Per Seconds	Encode Time	Inter-packet Time	Bytes per Packet
H.264	64 kbps	177X144 - 15 fps	7.36 ms/f	66 ms	542 bytes
H.263	24 kbps	120X90 - 10 fps	5 ms/f	100 ms	307 bytes

4.2 Security Model Evaluation

Before evaluating the security model, the entire protocols/algorithms need to be considered into various components and functional segments. For the security mechanisms, three sets of functions are considered, which are: Security handling at the sender, transmission of the security enabled payload, and the security handling at the receiver. Some security mechanisms, such as encryption, are relatively less computational intensive at the sender, however require more computational power at the receiver. In this case, delay figures at the receiver may contribute more to the overall end-to-end delay measurements. In other instances, if multimedia-enriched traffic payloads are being transmitted, then large payload sizes and other factors, such as numerous fragmentations, retransmissions, and other issues may also contribute to delay figures. In these scenarios, the transmission of security enabled payloads will have considerable effects on the performance, delay, and jitter figures.

Table 4.10 offers delay and overhead figures for various mechanisms and algorithms operating at the application layer. It should be noted that for voice and video categories, two codecs are mentioned for both the best case and worst case (channel/signal qualities) scenarios. The delay figure for text is negligible, due to the fact that text information is buffered and used in the payload of the UDP packet. The amount of text per packet will be discussed in the protocol section.

The delay figures for video codes have been given per frame since every UDP packet may include at most one video frame data.

Table 4.10 Suite-B algorithms delays and overhead figures at the sender

Mechanism/Data	Algorithm	Delay	Overhead
Text	N/A	≈ 0	1 byte
Voice	G.711	0.125 msec	240 bytes
	G.723.1	7.5 msec	20 bytes
Video	H.264	7.36 msec/frame	542 bytes
	H.263	5 msec/frame	302 bytes
Digital Signature	ECDSA-256	2.63 msec	64 bytes
Block Cipher Encryption	AES-128 GCM	9.44 nsec/byte	0
Adaptive Forward Error Correction	RS (255, 251)	11.30 nsec/byte	5 bytes
	RS (255, 247)	11.47 nsec/byte	9 bytes
	RS (255, 239)	11.82 nsec/byte	18 bytes
	RS (255, 223)	12.69 nsec/byte	37 bytes
Hashing	SHA-256	8.34 ns/byte	32 bytes
	SHA-384	8.97 ns/blk	32 bytes
	SHA-512	9.28 ns/blk	64 bytes
Cross-Layer Data	CLPs	≈ 0	6 bytes

The delay and overhead figures for encryption, forward error correction, and hashing mechanisms, have been given per block. Therefore to find the total amount of delay/overhead figures, depending on the specific payload size, those figures vary. The cross-layer parameters (CLPs) are also assumed to be readily available when a UDP packet is being constructed. Therefore no delay is incurred, however 48 bits (6 bytes) are the overheads.

4.2.1 Method 1 Details

Now we will discuss the details of the security protocol. At the application layer (section 3.2.3.1.1, Figure 3.17), the multimedia communication data is encoded and fed into the AES-GCM module. This will encrypt and authenticate the data. The output of the AES-GCM module will be fed to the Adaptive FEC (AFEC) module and also will be signed. Therefore non-repudiation mechanism is provided for the encrypted/authenticated multimedia communication data. The output of the AFEC and the QoS/Security cross-

layer information are being used as inputs to the Hash function, providing integrity. The outputs of the AFEC, Hash, ECDSA, and the QoS/Security cross-layer information are transferred to the transport layer, where they are packed in the UDP payload.

Based on Table 3.9, privacy is provided for the multimedia communication data only. Integrity is provided for both the cross-layer information and the multimedia communication data. Non-repudiation is provided for the output of the AES-GCM module (i.e., multimedia communication data). In this method, there is no protection for the UDP port/IP addresses.

The overhead and delay measures indicated in Table 4.11 are calculated based on the measurements on every individual functional entity. For instance, the delay figure for best voice quality is comprised of; 0.125 msec and 240 bytes for G.711, 2.63 msec for ECDSA-256, for AES-GCM, the related delay is calculated from 9.44 nsec per byte.

In the overhead calculation, in the worst case scenario, the instantaneous communication involves the simultaneous transmission of both multimedia and security data. In a scenario where the multimedia data contains a low rate codec information, such as in G.723.1, the data associated to the voice codec contains 20 bytes per frame. In this scenario the entire 256 byte ARS block does not have to be dedicated to carrying the G.723.1 codec only, therefore other data types may also be used in the ARS block. This is especially important to increase the efficiency of the ARS coding.

For the G.711 data packet, which contains 240 bytes, less than two AES-GCM 128 byte blocks are needed, therefore the AES-GCM delay will be 2.266 μsec. For the best case scenario, RS (260, 255) scheme is used and to preserve efficiency, G.711 data packets are fed into the RS coder in a pro-rata basis. Therefore for a 240 byte G.711 packet, the RS (260, 255) portion will be 244 bytes, 4 bytes overhead. The delay figure of the AES-GCM scheme is 9.44 nsec per byte, therefore for the G.711 case, 2.303 μsec delay is incurred. The SHA-512 hashing delay is 9.28 nsec per byte, however all other data (e.g., output of AES-GCM-ARS, ECDSA signature, and CLPs) input to the SHA algorithm

will be 375 bytes with the delay of 3.48 μsec. The total overhead is: 240 (G.711) + 64 (ECDSA) + 0 (AES-GCM) + 4 (ARS) + 64 (SHA) + 3 (CLPs) = 375 bytes and the total delay figure is (in μsec): 125 (G.711) + 2630 (ECDSA) + 2.303 (AES-GCM) + 2.712 (ARS) + 3.480 (SHA) = 2763.49 μsec or rounding to 2.764 msec. Note that the most time consuming algorithm is the ECDSA digital signature, which 800 to 1000 more time costly than the other algorithms. Therefore if digital signature is excluded from the calculation, the delay figure will be: 0.1335 msec.

Table 4.11 Packet-based application layer Suite-B algorithms delays and overhead figures

Channel/Signal Condition	Communication Type	Overhead (bytes)	Delay (msec)
Best	Text	133	2.631
	Voice	375	2.764
	Video	684	10.008
	Text+Voice+Video	930	10.144
Worst	Text	133	2.631
	Voice	154	10.132
	Video	483	7.641
	Text+Voice+Video	506	10.142

For the scenario where voice, video, and text are being transmitted, as mentioned, we assume text is already available. To calculate the delay for video and voice transmission, we consider the most time consuming algorithm to be the delay figure. In this case between 0.125 msec and 7.36 msec, the resulted delay will be 7.36 msec. This is due to the fact that both video and audio codecs run simultaneously and during the time that a video frame is being created, the voice codec has already performed its function.

For the case where G.723.1 is being transmitted, only 20 bytes of data will be produced for each frame, as mentioned. If no other data was transmitted, a smaller FEC block size is more efficient, such as an RS (32,28) [194]. However the assumption is that the rest of the RS (XXX, 255) block can be fed by other simultaneous data being transmitted, which preserves the efficiency of the scheme. It should be noted that according to traffic classifications performed in Chapter 3, average packet size in the wireless medium was 197 bytes, therefore according to Table 4.11, only text and worst case audio fall within

this range. The rest of the items require higher packet sizes, which may trigger errors and retransmissions in noisy or busy channels.

4.2.2 Method 2 Details

In this method, Suite-B cryptographic mechanisms are considered for the Transport Layer Security (TLS). The most current TLS version is 1.2 [178]. The basic functionality of TLS includes data encryption and integrity. TLS provides encapsulation of application layer contents and has three basic properties: Identity authentication using public key or asymmetric key cryptographic systems, a mutual authentication scheme, and anonymous communication. Reference [178] proposes Suite-B cryptographic version of TLS, including the AES-128 scheme for the encryption algorithm, ECDSA scheme for the digital signature mechanism and ECDH (Elliptic Curve Diffie-Hellman) for the key exchange scheme. The one of the main differences between the calculations in methods 1 and 2 is that in method 2, UDP header (8 bytes) is also included in the calculation, therefore with a good approximation, the overhead calculation figures of Table 4.11 hold for this method as well [195].

4.2.3 Method 3 Details

Suite-B cryptographic algorithms have so far been considered for the application and transport layers. In the network (IP) layer, they are being considered to interact with the IPSec mechanism. The current Suite-B IPSec draft has been introduced in RFC 4869 [179] covering Suite-B-GCM and Suite-B-GMAC-128 to provide ESP integrity protection and confidentiality through the usage of AES-GCM-128 (RFC4106). The usage of Galois Message Authentication Code (GMAC) for IPsec ESP and AH has also been mentioned in RFC 4543 [197]. We already discussed details about ESP and AH in section 2.7.4.2.1 in which we decided to use ESP as the preferred IPSec mode.

Suite-B-GMAC is a new entity in the Suite-B mechanisms applied in ESP mode, where each ESP configuration requires a minimum of ten bytes of extra overhead in addition to its payload. In the structure of GMAC, an eight-byte IV is used in every individual packet,

therefore each GMAC-ESP packet requires an additional 18 bytes of overhead in addition to the 12-byte tag. Therefore a total of 30 bytes overhead is incurred. The header sizes for UDP and IP are 8 and 20 bytes respectively. Therefore in addition to the overheads calculated in Table 4.11, there are 58 bytes additional overheads.

According to table 3.11, in method 3, the entire UDP payload, UDP source/destination port addresses, and IP source/destination addresses are protected thought privacy, integrity, and non-repudiation.

At the network layer, we are dealing with the flow of IP packet between two logical addresses on two different networks. Using IPSec, we are dealing with a protected tunnel session between two ends of the IPSec tunnels. Therefore for one active IPSec tunnel, a digital signature is required to provide non-repudiation. Again it is essential to use a traffic classification analysis to calculate the average session duration of an IPSec tunnel in order to find the average frequency of applying a digital signature to a session.

Reference [198] reveals that the IPSec latency depends on the packet size. The testbed involves the IXIA Traffic Generator linking two end-points using two security gateways via 1 Gbps Ethernet links, simulating the traffic scenario between the RNC (Radio Network Controller) and NodeB in UMTS (Universal Mobile Telecommunications System) architecture. The IPSec is based on the Encapsulating Security Payload (ESP), in a Tunnel Mode, using AES encryption and HMAC-SHA authentication algorithms.

The simulation results show that the larger the packet size, the lower the latency, therefore based on the platform used, Table 4.12 shows the number of cycles per byte versus the packet size, using IxChariot end-points, running a Tunnel Mode IPSec VPN, in an IEEE 802.11g network, based on T2500 Intel Core 2 platform.

Table 4.13 provides the delay/overhead figures of the Suite-B mechanisms in the presence of IPSec.

Table 4.12 IPSec performance measures versus packet size

Packet Size (bytes)	Cycles per Byte (delay/byte)	Packet Delay (μsec)	Functional Throughput (Mbps)
64	352	11.26	1.3064
128	346	22.14	1.3294
256	338	43.26	1.3616
384	202	38.78	2.2770
512	67	17.15	6.8655
640	64	20.48	7.1875
768	62	23.81	7.4198
1024	58	29.70	7.9304
1280	55	35.20	8.3628
1520	52	39.52	8.8458

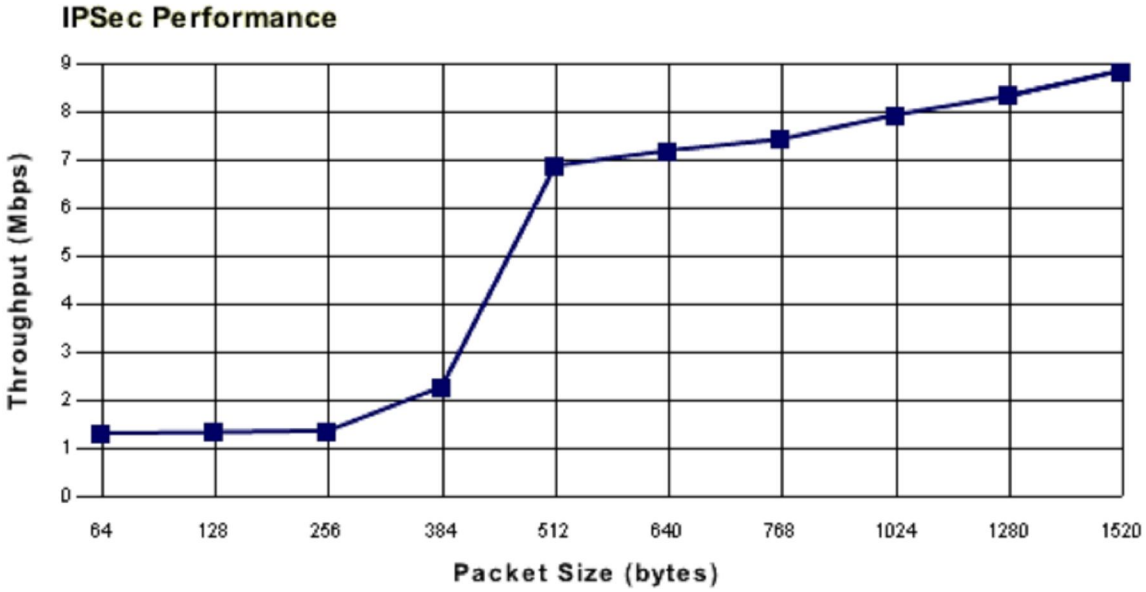

Figure 4.3 IPSec performance measures versus packet size

4.2.4 Method 4 Details

In this section, we consider MAC layer security delay/overhead in addition to the figures calculated in Table 4.11. We consider WPA-AES and its overhead and delay figure, which are given in Table 4.14.

Table 4.13 IPSec (network layer) Suite-B algorithms delays and overhead figures

Channel/Signal Condition	Communication Type	Overhead (bytes)	Delay (msec)
Best	Text	191	2.663
	Voice	433	2.918
	Video	742	10.031
	Text+Voice+Video	988	10.298
Worst	Text	191	2.663
	Voice	212	10.168
	Video	541	7.659
	Text+Voice+Video	564	10.160

Table 4.14 Summary WPA security delays/overheads

Security Method	Encryption Algorithm	Authentication Method	Number of Cipher Keys	Overhead (Bytes)
WPA-AES	AES + CTR	MAC	256	16

The WPA-AES mechanism involves AES-CTR and CBC-MAC schemes. Based on the platform used, AES-CTR uses 18.2 and CBC-MAC uses 15.2 cycles per byte. Therefore WPA-AES requires 33.4 cycles per byte. Note that the overheads mentioned in Table 4.11 need to be increased by 28 bytes (IP + UDP headers = 28 bytes).

The delay and overhead figures based on MAC layer WPA-AES are mentioned in Table 4.15.

4.2.5 Summary of Layered Security Schemes

We studied various implementations of Suite-B cryptographic algorithms in various layers, namely; application, transport, and network. For MAC layer, the current WPA mechanism is used in combination with the Suite-B-application layer algorithms. Table 4.16 shows the layer security profiles. The only layers that do not have complete implementations of the Suite-B algorithms are MAC and PHY layers, therefore when it comes to the MAC layer, MAC and application layers are used together.

Table 4.15 Application layer Suite-B algorithms and MAC security delays and overhead figures

Channel/Signal Condition	Communication Type	Overhead (bytes)	Delay (msec)
Best	Text	161	2.634
	Voice	403	2.896
	Video	712	10.019
	Text+Voice+Video	958	10.285
Worst	Text	161	2.634
	Voice	182	10.135
	Video	511	7.650
	Text+Voice+Video	534	10.151

Table 4.16 Suite-B layered mechanisms

Layer	Security profile	Mechanisms
Application	Suite-B-Application-Layer-AFEC	AES-128-GCM, RS (*k*, 255) SHA-512, ECDSA-256, ECDH
Transport	Suite-B-TLS-AFEC	AES-128-GCM, RS (*k*, 255) SHA-512, ECDSA-256, ECDH
Network	Suite-B-IPSec-AFEC	AES-128-GCM, RS (*k*, 255) SHA-512, ECDSA-256 GMAC-ESP, ECDH
MAC	WPA + Suite-B-Application-Layer	WPA2-AES

4.2.5.1 Minimum/Maximum Delay/Overhead Figures

In this section, based on the four methods described, minimum and maximum delay and overhead figures are compared. Based on delay figures given in Tables 4.11 and 4.12, the minimum and maximum delay figures can be shown on a single delay graph (Figure 4.4). Comparing the overhead and delay figures in all four methods, the application layer delay/overhead figures have relatively the lowest numbers and the network layer delay/overhead figures have relatively the highest numbers compared to the delay/overhead figures in other methods. Therefore the results obtained in methods 1 and 3 (i.e., application and network layers) are to be considered for calculating the delay/overhead max/min figures. According to Tables 4.11 and 4.13, the Suite-B

algorithms delays and overhead figures are calculated for both the application (App) and network (IP) layers under the best (B) and worst (W) channel/signal conditions for Text, Voice, Video, or Text+Voice+Video (TVV). Therefore for each method, there are eight calculated values, four of which for the best and the other four for the worst channel/signal conditions. Therefore there are 16 calculated values for overhead and 16 calculated values for delay figures. However since the delay/overhead values for communications based on text only are equal in both the best and worst channel/signal conditions, therefore there are actually 14 calculated values to be used for comparisons. These comparisons are shown in Figures 4.4 and 4.5 for min/max delay/overhead figures respectively. According to the Figure 4.4 the lowest delay figure is related to the transmission of text using method 1 (application layer) and the highest delay figure is related to the transmission of TVV using method 3 (network layer) for the best quality.

The following table shows the list of abbreviations used in Figures 4.4 and 4.5:

Abbreviation	Compete Definition
App-Text	Communication involving **Text** only at the **Application layer**, for both best and worst channel/signal conditions
App-B-Voice	Communication involving **Voice** only at the **Application layer** for the **Best** channel/signal condition
App-W-Voice	Communication involving **Voice** only at the **Application layer** for the **Worst** channel/signal condition
App-B-Video	Communication involving **Video** only at the **Application layer** for the **Best** channel/signal condition
App-W-Video	Communication involving **Video** only at the **Application layer** for the **Worst** channel/signal condition
App-B-TVV	Communication involving full multimedia content (**Text-Voice-Video**) only at the **Application layer** for the **Best** channel/signal condition
App-W-TVV	Communication involving full multimedia content (**Text-Voice-Video**) only at the **Application layer** for the **Worst** channel/signal condition
IP-Text	Communication involving **Text** only at the **Network (IP) layer**, for both best and worst channel/signal conditions
IP-B-Voice	Communication involving **Voice** only at the **Network (IP) layer** for the **Best** channel/signal condition

IP-W-Voice	Communication involving **Voice** only at the **Network (IP) layer** for the **Worst** channel/signal condition
IP-B-Video	Communication involving **Video** only at the **Network (IP) layer** for the **Best** channel/signal condition
IP-W-Video	Communication involving **Video** only at the **Network (IP) layer** for the **Worst** channel/signal condition
IP-B-TVV	Communication involving full multimedia content (**Text-Voice-Video**) only at the **Network (IP) layer** for the **Best** channel/signal condition
IP-W-TVV	Communication involving full multimedia content (**Text-Voice-Video**) only at the **Network (IP) layer** for the **Worst** channel/signal condition

Based on overhead figures given in Tables 4.11 and 4.12, the minimum and maximum overhead figures can be compared on a single delay graph (Figure 4.5). As shown in Figure 4.5, Network layer contributes to the highest overhead (988 bytes per packet) compared to the overhead figures calculated in other methods. The lowest overhead (133 bytes per packet) occurs when text is transmitted using method 1 (application layer).

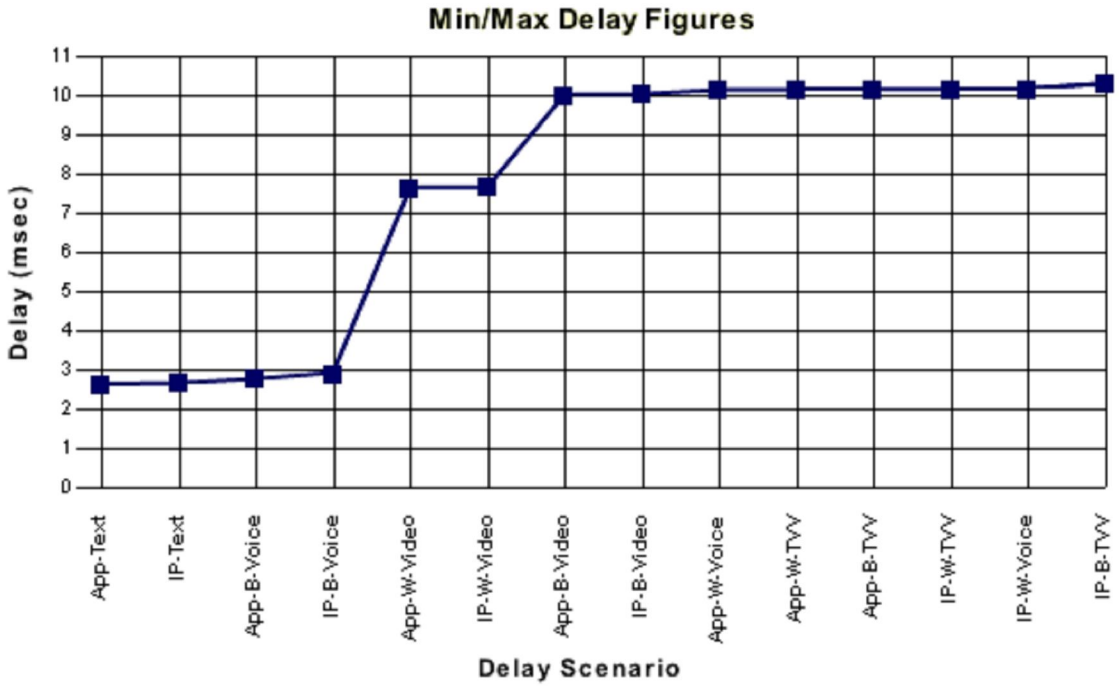

Figure 4.4 Max/Min delay figures

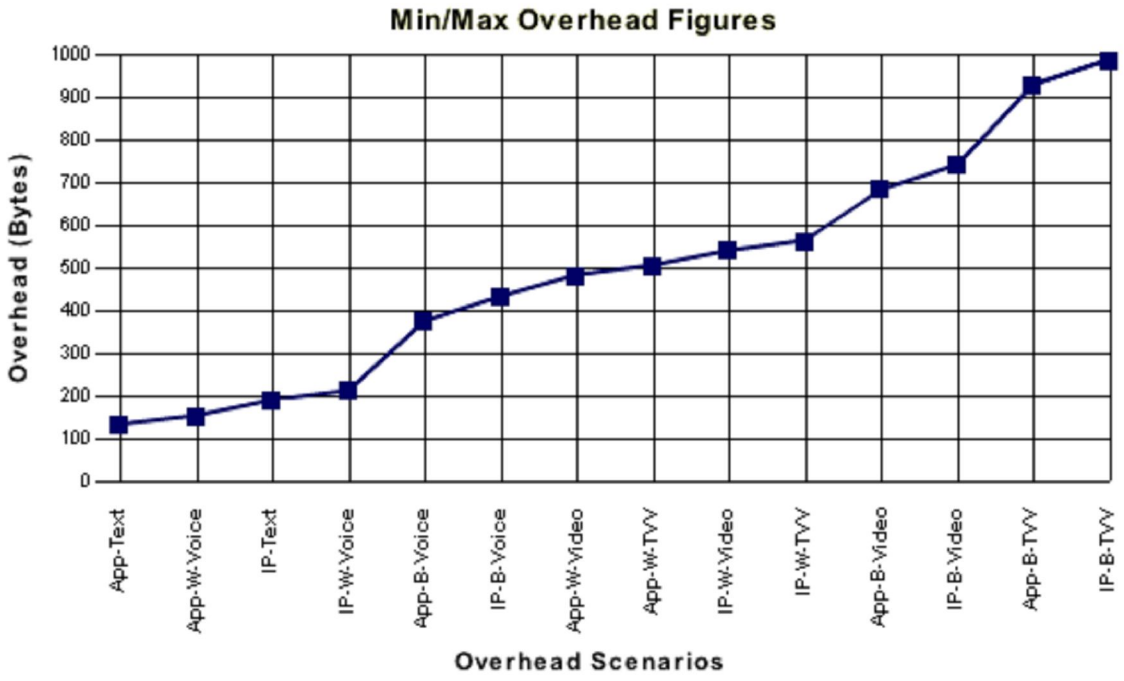

Figure 4.5 Max/Min overhead figures

4.2.6 Detail of the Security Protocol Handshakes and Flows

We have so far studied the various security components at each layer. These components and algorithms should be utilized through a set of handshakes and protocol flows, which are discussed in this section.

4.2.6.1 Application Layer Security Protocol Handshakes

In section 3.2.2.2.2, the key exchange protocol (based on ECDH) was introduced. It was also discussed that ECDH is prone to man-in-the-middle attack, therefore an authenticated Diffie-Hellman key agreement protocol is required, For this we assume both parties have authenticated themselves to each other using public-key certificates.

Once the key is exchanged, *A* and *B* are able to transmit information to one another. Assume *A* is transmitting to *B*. As mentioned, at the application layer, we are dealing with messages and depending on the application in use and the level of the security of the transmitted information, a digital signature is applied to the ongoing traffic.

In regards to the frequency of digital signature deployment, ECDSA is able to sign messages as large as 2^{64} byte in length, therefore to conserve battery power, the security protocol is set to sign a group of blocks instead of signing each block of data. The number of blocks used for one signature depends on the context of the multimedia data. If high critical data are being transmitted and the validity of the sender/data needs to be checked, the frequency of the digital signature deployment can be relatively higher, such as every minute or every 10 minutes once. On the other side if a low critical data is being transmitted, a digital signature can be deployed less frequently, such as once a day.

In a bursty type of data transmission, there are periods of time that no data is being transmitted. To maintain data freshness each transmitted block of data should be accompanied by an encrypted time-stamp coupled to a sequence number and the actual time of all entities in the network should be synchronized. The sequence number is used to track every transmitted block of multimedia data.

In an extreme case, a signature can also be used per packet. In this scenario, a more accurate delay/overhead calculation is required. It should be noted that both AES-GCM and ARS schemes operate in block-based algorithms, 16 and 256 bytes respectively.

Figure 4.6 shows the application layer security protocol and functional entities. Figure 4.7 focuses on the security protocol interactions between party *A* and party *B*. The protocol starts with both parties (*A* and *B*) authenticating themselves to each other, using public key certificates. Once authenticated, ECDH handshakes will start. Following the authenticated key exchange, adaptive FEC schemes is performed. Then a digital signature will be applied to the block of data, then encryption and a second hash function is applied to the data until the final code-word is transmitted and termination request is sent and accepted.

The application-layer provisioning Suite-B algorithms provide:

Authentication: This is provided for parties A and B through the usage of public and private keys and for the multimedia data through the deployment of GCM algorithm.

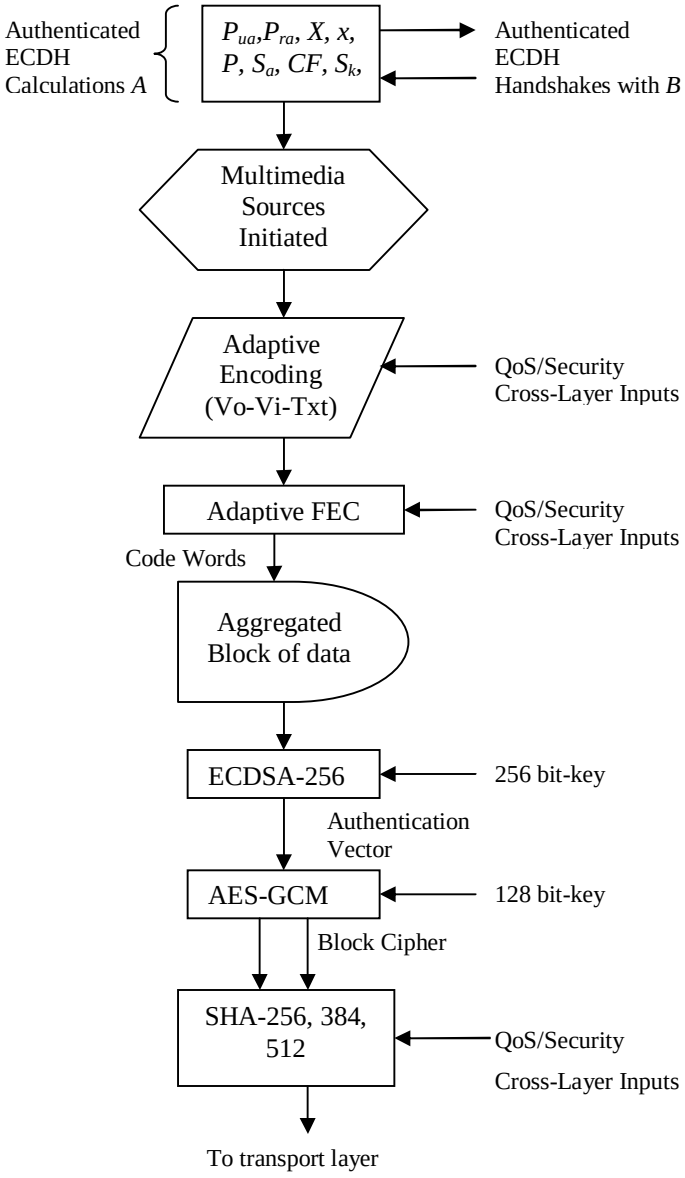

Figure 4.6 Application layer security protocol/functional entities

Privacy: Privacy is provided through the deployment of AES algorithm. Following the cross-layer adaptive treatment of multimedia data, AES is applied to provide a strong encryption mechanism.

Adaptive encoding: The authenticated and encrypted multimedia data and the QoS/security cross-layer information are used as inputs to the adaptive FEC scheme.

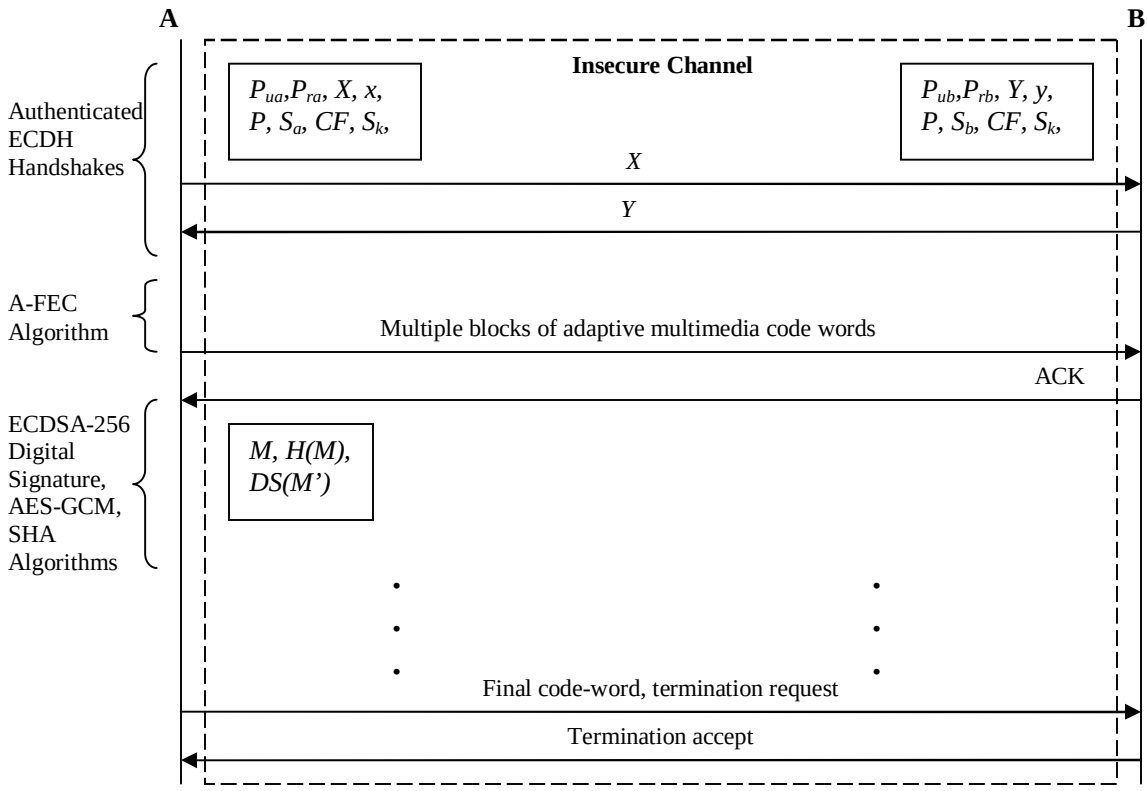

Figure 4.7 Application layer security protocol handshakes

Non-Repudiation: Non-repudiation is provided through the deployment of the digital signature algorithm; ECDSA. ECDSA is applied to the output of the adaptive FEC code words. The deployment of ECDSA at the application layer provides a non-repudiation service for messages containing multiple blocks of data with a time-limit that is being imposed by the criticality of the data. For a very critical data, a signature can be applied to 1 to 10 minutes of data blocks. For a non critical data, a digital signature can be used for 1 entire day worth of data blocks.

Integrity: Integrity is provided through the deployment of SHA-256, -384, and 512. The output of the adaptive FEC code words, the ECDSA digital signature, and the QoS-Security cross-layer information are all used as the SHA algorithm inputs.

4.2.6.2 Transport Layer Security Protocol Handshakes

Transport-layer Suite-B algorithms deployment involves treating sessions instead of messages. AES-GCM provides data integrity and encryption for session flows. ECDSA is used in TLS, however applies only for authentication handshake and public keys, not for session flows.

According to Table 3.10, cross-layer information, multimedia communication data, and UDP source/destination addresses are protected by privacy and integrity mechanisms. The IP source and destination addresses are not protected in this method. As mentioned, we are dealing with sessions at this layer. A session is an ongoing transmission of data between two ports on both the sender and receiver devices. Therefore since ECDSA is used in the session handshake, therefore as long as the session is active, one digital signature is sufficient.

Suite B implementations of TLS has been mentioned in RFC 5246 [178] and RFC 5430 [179]. For the 128-bit security level, the TLS-Suite-B scheme offers TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256. During the TLS handshake protocol, Suite B algorithms utilize the form of key-agreement based on ECDHE_ECDSA (using the P-256 Elliptic Curve). This is the Ephemeral Unified Model key-agreement scheme where the server authenticates its part through the deployment of an ECDSA signature. The server must supply an X.509v3 certificate to the client, which contains an acceptable (accepted by the client) ECDSA key that will be used to verify its signature. That certificate must be signed using ECDSA by a CA. Figure 4.8 shows the protocol handshake flows between Ports A and B. Based on the certificates and the ECDSA signature, Ports A and B authenticate each other and the data flow starts (AES_128_GCM_SHA256) and continues until the termination REQ/ACK are transmitted. Optional functions are marked with “*”.

The same process repeats for the client, which is required be authenticated during the TLS handshake (mutual authentication), then it must possess an X.509v3 certificate

containing an acceptable (accepted by the server) ECDSA key, signed using ECDSA by a CA. The client authentication is initiated by the issuance of “ECDSA_sign” certificate request message, which is issued by the server, requesting the client for an appropriate ECDSA certificate.

In terms of overhead, the digital signature is the costliest algorithm in Suite-B and for TLS, one digital signature is required per active session. The duration of a session depends on various parameters, including: Link quality and session cease/start triggered by the application. For the application-layer security protocol, as mentioned, a critical application may require a digital signature every minute. For the same application running, the protocol for the transport-layer security is still dependent on the ongoing session. As long as the session is valid, one digital signature is sufficient. A session traffic analysis is required to find the average time of a session, which may be more than 1 minute (assuming a good link quality). Therefore for critical applications, the overhead for the Suite-B TLS deployment may be less than the application-layer Suite-B deployment. However for non-critical applications, the sessions may be terminated and reestablished multiple times per day, therefore a digital signature may be required more than once a day, which increases the overhead for the Suite-B TLS deployment compared to the application-layer Suite-B deployment.

4.2.6.3 Network Layer Security Protocol Handshakes

Before an IPSec tunnel is established and data is transmitted through the VPN tunnel, the two parties at the two end of the tunnel (VPN client and server) should authenticate each other and exchange ephemeral ECDH keys. This is considered in the Internet Key Exchange (IKEv2) Protocol, specified in RFC 4306 [196]. IKE performs mutual authentication among two parties, by which an IKE security association (SA) is established, which includes shared secret information used to establish SAs for Encapsulating Security Payload (ESP). A set of cryptographic algorithms, including Suite-B algorithms may also be set up. IKEv2 specifies the following services for the ESP mode:

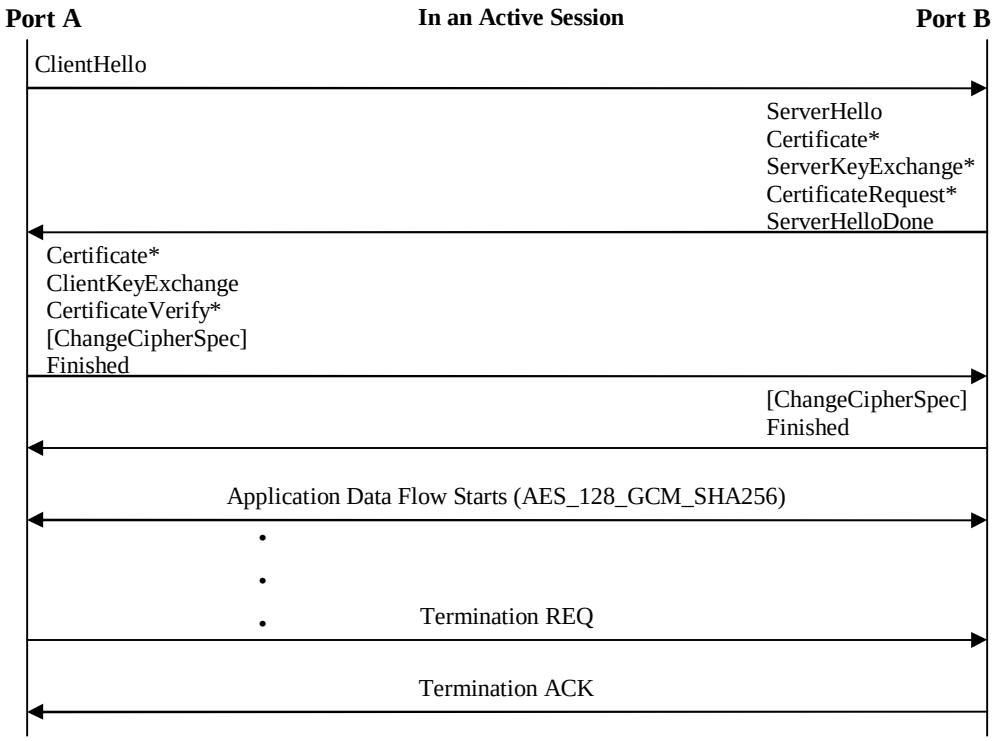

Figure 4.8 Transport layer security protocol handshakes

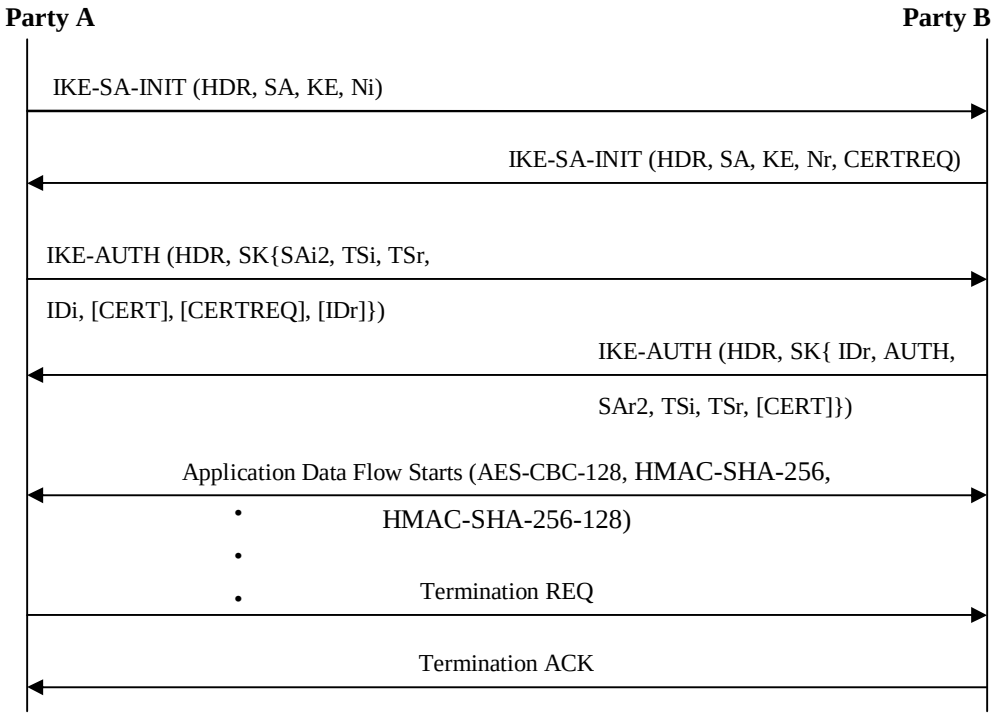

Figure 4.9 IPSec (IKEv2) security protocol handshakes

Encryption: This is done by AES with 128-bit keys in CBC mode (AES-CBC-128).

Data Authentication and *Integrity:* IKEv2 uses the Pseudo-Random Function (PRF) for generating keying material for authentication of the IKE SA and integrity. This is done through the deployment of HMAC-SHA-256 and HMAC-SHA-256-128.

Authentication: IKEv2 Elliptic Curve authentication is based on ECDSA-256.

Figure 4.9 shows the IKEv3 protocol handshake flows.

The deployment of Suite-B algorithms at the IP layer provides various security services (RFC 4869 [180]). For instance ECDH-ECDSA is used in the authentication handshake however does not provide non-repudiation services for the tunnel traffic.

In terms of overhead, since the digital signature is the costliest algorithm in Suite-B and since for IPSec, one digital signature is required once per active tunnel and the duration of an active tunnel depends on various parameters, including link quality. For the application-layer security protocol, as mentioned, a critical application may require a digital signature every minute. For the same application running, the protocol for the network will require only one digital signature is sufficient. A tunnel traffic analysis is required to find the average time of a tunnel connection, which is expected to be more than 1 minute (assuming a good link quality) and it is also expected to be more than the session duration at the transport layer. Therefore for critical applications, the overhead for the Suite-B IPSec deployment may be less than both the application-layer and transport layer Suite-B deployments. For non-critical applications, the tunnel may be terminated and reestablished multiple times per day, however it is expected to last longer than a session, thus a digital signature may be required more than once a day. Therefore the overhead associated to the IPSec Suite-B deployment is expected to be more than the deployment of Suite-B at the application layer and less than the Suite-B deployment at the transport layer.

4.2.6.4 Cross-Layer QoS/Security Parameters' Liveliness

The liveliness of the imported cross-layer QoS/Security parameters at the application layer determine how fast the system can adapt to environment changes. The variation of the environment's quality is probabilistic in nature. Both slow and rapid changes to the quality are dependent on many parameters, including: The number of wireless clients/APs, the operational channels, the average distances between clients/APs, and so many other parameters. The protocol is designed to start probing the QoS/security cross-layer information once every second. An adaptive scheme may be deployed to reduce the frequency of probing in an overall stable channel quality condition and to increase the frequency of probing in a rapid quality changes of the channel. Therefore the liveliness period of the cross-layer QoS/security parameters is set at once every second.

4.3 Functions at the Receiving-End

In this section we assume the traffic has been transmitted successfully to Party B, where appropriate mechanisms engage in the reversed algorithms take place. These reverse actions include: Decapsulation, decipher, reverse hash, signature verification, and multimedia content decode. Symmetric algorithms are schemes, in which both the sender and receiver have to perform identical functions both in the encoder and decoder. Therefore the incurred delay in both encoder and decoder is equal, such as in G.711 and G.723.1 codecs. Other schemes, in which the encoder and the decoder run different functions, are asymmetric schemes. For instance ECDSA signature (sender) is less computationally intensive than ECDSA verification (receiver), therefore it takes relatively less amount of time for the sender to sign. On the other hand H.264 and H.263 require relatively less amount of time to decode at the receiver side. These are given in Table 4.17.

Table 4.17 Maximum decoding latency at the receiver

Scheme Group	Scheme Algorithm	Relative Decoder-Encoder Delay	End-to-End Delay
Multimedia	G.711	≈	500 μsec
	G.723.1	≈	15000 μsec
	H.264	<	12300 μsec
	H.263	<	8500 μsec
Key Exchange	ECDH	<	5200 μsec
Digital Signature	ECDSA-256	>	6520 μsec
Reverse hash	SHA-512	≈	18.56 nsec per byte
Decryption	AES-GCM	>	23.46 nsec per byte
Error Correction	ARS	>	41.24 nsec per byte
IPSec Mode	Tunnel	≈	variable
	WPA-AES	≈	33.4 nsec per byte

Table 4.17 summarizes the functions used in this thesis and the relative encoder-decoder and end-to-end delay figures. For instance G.711 encoder and decoder have the same delay of 0.125 msec. Therefore the end-to-end delay is 0.5 msec. The ARS decoder is 2.25 times more time consuming than the ARS encoder.

4.4 Security Analysis

This section is concerned with the security strength of this system based on the security strength of the involved algorithms.

The security mechanisms used in this thesis are based on Suite-B cryptographic algorithms. Numerous references [168, 171, 172, 174, 176, 177] provide formal security analysis and proof for the security strengths of the Suite-B algorithms, which are considered secure based on today's computational power. These are explained below:

Key-Size: The 128-bit key size when used with a strong encryption algorithm, such as AES, coupled with a message authentication technique, provided by the Galois-Counter Mode (AES-GCM), offers a very strong security algorithm based on today's

computational power and is expected to remain unbreakable until 2030 [199, 200]. The 256-bit key size is used in the operation of ECDSA-256, which provides a strong digital signature algorithm. The AES-GCM and ECDSA are two of algorithms used in Suite-B.

Key Exchange: The ECDH scheme is based on Elliptic Curve Cryptography (ECC) and Diffie-Hellman (D-H) algorithm. Based on today's computational power, Diffie-Hellman is secure against eavesdropping (passive attack), however without a strong mutual authentication mechanism, Diffie-Hellman is insecure against Man-in-the-Middle attack (active attack). Therefore the ECDH algorithm is considered secure based on today's computational power [199, 200].

Digital Signature Security Analysis: The digital signature scheme used in this thesis is based on ECDSA-256, which is cryptographically comparable to RSA-3072 and DSA-3072. ECDSA is based on the DSA algorithm and despite the lack of a complete security proof, extensive attempts at cryptanalysis on either DSA or ECDSA have yet not yield any success [201]. Reference [202] has demonstrated that ECDSA is secure against forgery by adaptive chosen-message attack if the hash function is collision resistant and the elliptic curve group is modeled by a generic group.

Hash Function Security Analysis: Though SHA-256, SHA-384, and SHA-512 are not considered collision resistant functions, however they have still not been broken yet and are considered computationally secure with today's computational power [203, 204].

IPSec Security Analysis: As mentioned, IPSec can be used in two flavors; Encapsulating Security Payload (ESP) and Authentication Header (AH). ESP protocol is the preferred choice as it provides authentication, confidentiality, and integrity. The ESP security strength is based on the security strength of AES, which is considered secure.

Wi-Fi Security Option: At the MAC layer, the preferred security option is WPA-AES with key sizes above 128 bits, which are considered secure.

4.4.1 Possible attacks on the system

The attack on the system can be in two forms; the attack on the Suite-B cryptographic algorithms and the attack on the security protocol.

4.4.1.1 Possible Attacks on the Suite-B Algorithms

The possible attacks on the Suite-B algorithms are directly related to the security model and may include the followings: Dictionary-, brute-force-, birthday-, replay-, Man-in-the-Middle-, forged-certificate-, IP-fragmentation-, ping-of-death-, UDP-session-hijacking-, traffic-analysis, and other types of attacks. The security strengths of the security algorithms deployed by this system help withstand the combination of the mentioned attacks. These will be discussed in the following contexts:

Attack on the Suite-B algorithms: Suite-B algorithms used in this thesis, which are: ECDSA-256, AES (128)-GCM, and SHA-(256, 384, and 512), are all considered to be computationally secure. The digital signature (ECDSA) is based on the Elliptic Curve Cryptography (ECC) and one of the known classical attacks on the ECC algorithms is based on an exponential-search-algorithm with a complexity of $O(2^{\frac{n}{2}})$, which represents the square-root attack. The largest ECC-based cryptographic algorithm, which has publicly been broken, is the 109-bit ECC algorithm. Therefore a scheme with 256-bit key-size is still considered secure [205].

Attacks on the Digital Signature Algorithm: Though there are numerous attacks applicable to ECDSA, including birthday and brute-force attacks. However, as mentioned in this section, ECDSA-256 is considered unbreakable [206].

Attacks on the Hash Algorithm: All hash algorithms are subject to birthday attacks, which include all versions of SHA family. SHA-256, SHA-384, and SHA-512 have still not been broken yet and are considered computationally secure with today's computational power [203, 204, 207]

AES used in IPSec and WPA-AES: Both IPSec and WPA-AES use AES encryption algorithm. AES-128 is expected to remain secure till 2030 [208], however the method of attacking it is again through brute-force and birthday attacks.

4.4.1.2 Possible Attacks on the Security Protocols

The strength of the security protocol is required to be high enough to counter any possible attacks. Including:

Man-in-the-Middle-Attack (MitMA): MitMA is when an adversary is located between legitimate parties; *A* and *B* and is able to intercept the communication. The ECDH key-exchange algorithm is prone to MiTMA and to combat this security weakness, a mutual authentication scheme is required. An alternative scheme is to deploy ECMQV (Elliptic Curve Menezes-Qu-Vanstone), which is based on the parties preexisting static public keys. Once keys are safely exchanged, the ongoing communication will be encrypted for privacy, which limits the ability of the adversary.

Forged Certificate: Through ARP poisoning, the illegitimate party; *C* can pose as *B* and direct the initial communication from *A*. Then *C* can reply to *A* by forging B's certificate. The deployment of a trusted Certificate Authority (CA) solves this problem.

UDP Session Hijack: Due to the fact that UDP does not deploy synchronizing and packet sequencing, it may be prone to session hijack. The adversary is simply required to forge a server reply to a UDP client request before the actual server responds. The deployment of TLS will however prevent such a security threat through the built-in session security of TLS.

Data Block Injection or Data Block Deletion: As the transmitted data may become non-streaming type, the flow of the information could encounter frequent breaks and sudden bursts. In order to provide data freshness and avoid any malicious activities, which may lead to deliberate deletion of information, an encrypted synchronized time-stamp coupled

to a sequence number should be accompanied with every transmitted block of data. This way the receiver is able to put together different blocks and execute various checks to ensure data has not been altered by any illegitimate party.

Denial-of-Service (DoS) Attack: DoS attack in a multilayer attack and could involve a continual transmission of queries targeting a specific application, session, port, MAC or IP address. At the application layer, any attempt to hamper the function of the protocol will result in a DoS attack. For example an adversary could target *B* and send many forged queries masquerading *A*. If each query is transmitted along with the digital signature of the sender, then the receiver can filter out the queries with digital signatures of any illegitimate parties. At transport and network layers, the same issue may take place and again the receiver (port/party) can filter-out queries with unidentifiable senders.

4.5 An Experimental Setup and Results

In this section, an experimental test is setup and results will be evaluated.

4.5.1 QoS Analysis

Table 4.18 shows QoS parameters for the QoS model. Table 4.19 presents the zones of operations based on the feedback received from lower layers. For instance Zone 8 indicates very bad channel/signal qualities, which forces the selection of the lowest encoders for both voice and video, which are G.723.1 (6.4 Kbps) and H.263 4 (the lowest rate; 24 Kbps). On the same front, since the degrading channel/signal qualities may result in a relatively high number of errors and RF issues, a lower value for k is chosen (k = 223), which may increase the error correction capability. On the other side, Zone 1 indicates very good channel/signal qualities.

The QoS and security models presented in this thesis are summarized in Table 4.18.

All parameters mentioned in Table 4.18 contribute to the Zone selection. To consider the impact factor of each parameter, we assign numbers to each zone, based on Figure 4.10. Since the normal curve is used in many physical measurements (e.g., RF effects, traffic characteristics, etc.), we use this curve to represent the Zone variations. Therefore Table 4.20 summarizes the zones and their equivalent value ranges.

Table 4.18 Cross-Layer QoS feedback to the application-layer

Cross-Layer Parameter	Layer
RSSI	PHY
SNR	PHY
WMM	MAC
DSCP	Network
IPSec/VPN/Suite-B-IPSec	Network
WPA2-AES	MAC
TLS-Suite-B-TLS	Transport

Table 4.19 Zone-based encoder, FEC variations

Zone	Encoder (Voice-Video) - Data Rate	FEC, *k* value
Zone 8	G.723.1 – 5.3 Kbps H.263 – 24 Kbps	*k* = 223
Zone 7	G.729a – 8 Kbps H.263 – 28 Kbps	*k* = 223
Zone 6	G.726 – 32 Kbps H.263 – 32 Kbps	*k* = 239
Zone 5	G.728 – 16 Kbps H.263 – 36 Kbps	*k* = 239
Zone 4	G.729 – 8 Kbps H.263 – 42 Kbps	*k* = 247
Zone 3	G.722.1 – 48 Kbps H.263 – 48 Kbps	*k* = 247
Zone 2	G.722 – 64 Kbps H.263 – 56 Kbps	*k* = 251
Zone 1	G.711 – 64 Kbps H.264 – 64 Kbps	*k* = 251

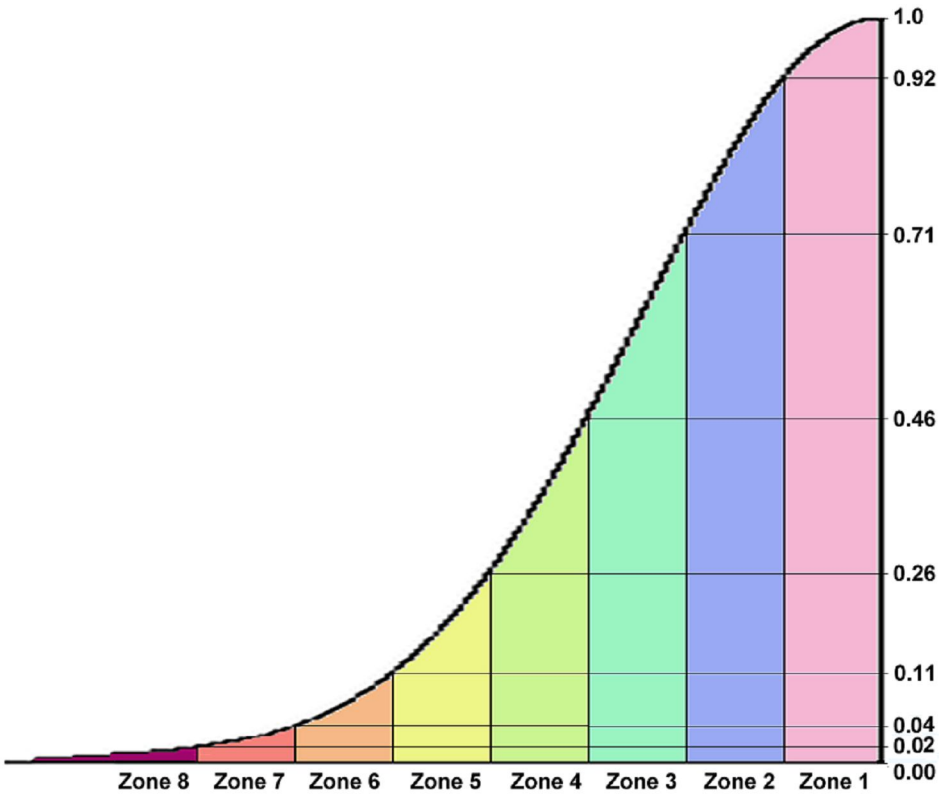

Figure 4.10 FEC's zone ranges

Table 4.20 Zones and equivalent ranges

Zone	Range
1	[1.0 - 0.92)
2	[0.92 - 0.71)
3	[0.71 - 0.46)
4	[0.46 - 0.26)
5	[0.26 - 0.11)
6	[0.11 - 0.04)
7	[0.04 - 0.02)
8	[0.02 – 0.00]

Each of the parameters specified Table 4.18 has a specific Impact Factor (IF) in the calculation of the Zone. For instance the value of the RSSI has a higher IF on the Zone determination compared to DSCP. These IF values are mentioned in Tables 4.21 to 4.27.

Table 4.21 Impact Factor for RSSI ranges

RSSI Range	Impact Factor (IF)
Below -90 dBm	0.0
-90 dB to -80 dBm	0.1
-80 dB to -70 dBm	0.4
-70 dB to -60 dBm	0.6
-60 dB to -50 dBm	0.8
-50 dB to -40 dBm	0.9
Above -40 dBm	1.0

Table 4.22 Impact Factor for SNR ranges

SNR Range	Impact Factor (IF)
0 - 2 dB	0.0
2 dB to 7 dB	0.1
7 dB to 15 dB	0.3
15 dB to 20 dB	0.5
20 dB to 30 dB	0.7
30 dB to 40 dB	0.8
40 dB to 50 dB	0.9
Above 50 dB	1.0

Table 4.23 Impact Factor for WMM ranges

WMM Range	Impact Factor (IF)
000	0.7
001 - 101	0.8
110	0.9
111	1.0

Table 4.24 Impact Factor for DSCP ranges

DSCP Range	Impact Factor (IF)
00000000 – 00011111	0.7
00100000 – 10111111	0.8
11000000 -11011111	0.9
11100000 – 11111111	1.0

Table 4.25 Impact Factor for VPN/IPSec settings

VPN/IPSec	Impact Factor (IF)
IPSec – ESP (Tunnel)	0.5
IPSec – ESP (Transport)	0.6
IPSec – AH	0.7
None	1.0

Table 4.26 Impact Factor for MAC-layer security settings

MAC-Later Security	Impact Factor (IF)
WPA2	0.7
WPA	0.8
WEP	0.9
None	1.0

To calculate the combined IF value, let's assume that the schemes in Table 4.28 are all present simultaneously. The combined IF value is: 0.9 X 0.8 X 0.9 X 0.9 X 1.0 X 0.7 X 0.5 = 0.204, which according to Table 4.20, it corresponds to Zone 5 and according to Table 4.19, the codec selected are: G.728 (16 Kbps), H.263 (36 Kbps), and k = 239.

Table 4.27 Impact Factor for transport-layer security settings

Transport-Later Security	Impact Factor (IF)
Suite-B-TLS	0.5
TLS	0.7
None	1.0

Table 4.28 Cross-layer QoS feedback to the application-layer

Cross-Layer Parameter	IF
RSSI (- 45 dBm)	0.9
SNR (38 dB)	0.8
WMM (110)	0.9
DSCP (11000000)	0.9
IPSec/VPN Capability (None)	1.0
MAC-Layer Security (WPA2)	0.7
Transport-Layer Security Suite-B)	0.5
Combined IF value:	**0.204**

4.5.1.1 Impact Factor – Zone – MOS Value Mapping

The effect of each individual Impact Factor (based on the imported cross-layer parameters) impacts the total IF. This leads to the selection of a specific Zone. Each Zone will enforce specific encoders to be used for both voice and video. Voice codecs being selected are: G.723.1, G.729a, G.728, G.729, G.726, G.722.1, G.722 and G.711. According to Tables 2.5, 4.19, and 4.20 and Figure 4.10, the relationship between the final IF, Zone, and MOS values are shown in Figure 4.11.

As the cross-layer feedback indicates an improving channel/signal quality, lower Zones are selected, which then better voice codecs with higher MOS score values are selected.

As mentioned in section 2.1.1, QoE measurements may use non-intrusive voice quality analyses for predicting MOS value using passive voice clarity evaluations. Once a MOS value is calculated this way, it will relate to specific Zone and Impact Factor, where were discussed in this section and also depicted in Figure 4.11.

Figure 4.11 can be used to match a targeted MOS value to a Zone and eventually achieve an Impact Factor. For example if in a VoIP scenario, a MOS score of 4.10 or above is required, according to Figure 4.10, Zone 3, 2, or 1 can be used (Impact Factor of 0.71 or better). This means on average, every individual Impact Factors should be 0.71 or above.

4.5.2 Security Analysis

For the security part, Table 4.29 sums all layered security model parameters.

4.6 Experimental Results

In this section we construct an experimental testbed using Labview 8.5 system design. Labview is capable of generating real-time traffic based functional elements and producing graphical and visual performance diagrams.

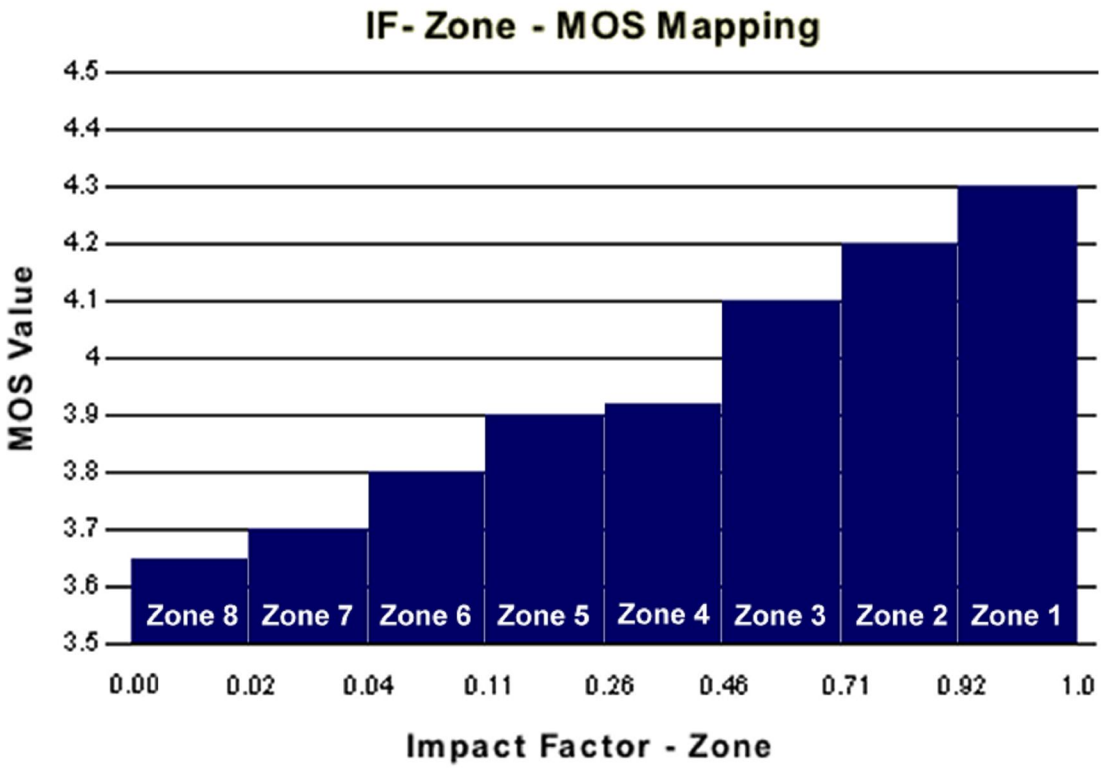

Figure 4.11 Total Impact Factor, Zone, and MOS value mapping

Table 4.29 Suite-B layered mechanisms

Layer	Cross-Layer Parameter Export to Application Layer / Scheme Availability	Schemes
Application	Application-Suite-B	AES (256)-GCM-FEC SHA-(256, 384, 512) ECDH - ECDSA (256)
Transport	TLS-Suite-B	Suite-B-TLS
Network	IPSec-Suite-B	Suite-B-IPSec
MAC	WPA2-AES + Application-Suite-B	-

The delay and overhead effects of multimedia codes (i.e., G.711, G.723.1, H.263, and H.264), Suite-B functional codes (i.e., ECDH, ECDSA, AES-GCM, and SHA) and the R-S code will be tested using a Labview testbed for evaluating the experimental results based on the analytical results found in Table 4.11. The end-to-end delay figures are also going to be evaluated.

The transport protocol conveying multimedia and security-enabled traffic is based on UDP and IP. Both sender and receiver have individual interfaces installed on the end-user PCs. The sender interface has selectors to select the type of digital signature (ECDSA-256), encryption (AES-GCM), hashing (SHA-512) and etc.. At the receiver, the reverse processes will take place and the payloads are extracted with the processing delays.

On the receiver side, following the correct decode of the payload, the Labview interface calculates and shows the following graphs (Figure A.2): Instantaneous and average packet delay versus time, instantaneous and average throughout, and instantaneous and average jitter versus time.

The system was run numerous times, during a month period, under various channel conditions, and each time over 30 minutes of run time. The system is able to show instantaneous as well as average results in a discrete manner.

Tables 4.30 and 4.31 show the CLP values in one of the testbed runs and the corresponding Zone and multimedia codecs in use.

Therefore the sender selectors will automatically choose the parameters mentioned in Table 4.32.

Table 4.30 Testbed CLPs values

CLP value	Impact Factor (IF)
RSSI: -38 dBm	1.0
SNR: 52 dB	1.0
WMM: 111	1.0
DSCP: 11100000	1.0
IPSec: None	1.0
MAC-Security: None	1.0
Transport-Layer-Security: None	1.0
Total IF	1.0

Table 4.31 Zone 1 multimedia settings

Multimedia settings (Zone 1)
Video codec: H.264 (64 kbps)
Audio code: G.711 (64 kbps)
k (R-S code): 251

Table 4.32 Zone 1 sender's selectors

ECDSA-256
SHA-512
AES (256)-GCM (128)
R-S (260,255)
G.711
H.264

The average analytical value for end-to-end delay is 19.64 msec on average on 930 byte overhead when video, audio, and text are transmitted simultaneously.

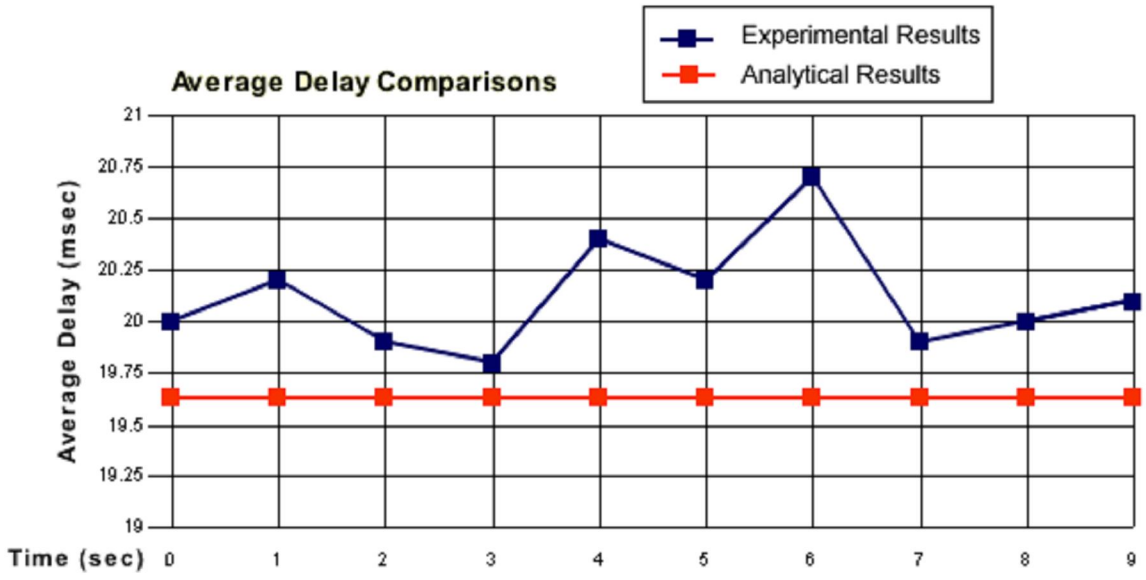

Figure 4.12 Experimental and theoretical results for delay comparisons

According to the experimental results, given in Figure 4.12, for 500 test trials, the average delay is 20.12 ± 0.001 msec (99% CI) with the standard deviation of 0.081 msec, which results an average 2.44% (and 5.40% maximum) differences between analytical and experimental results, which validates our assumptions. The fluctuations in the experimental results are due to channel variations and retransmissions.

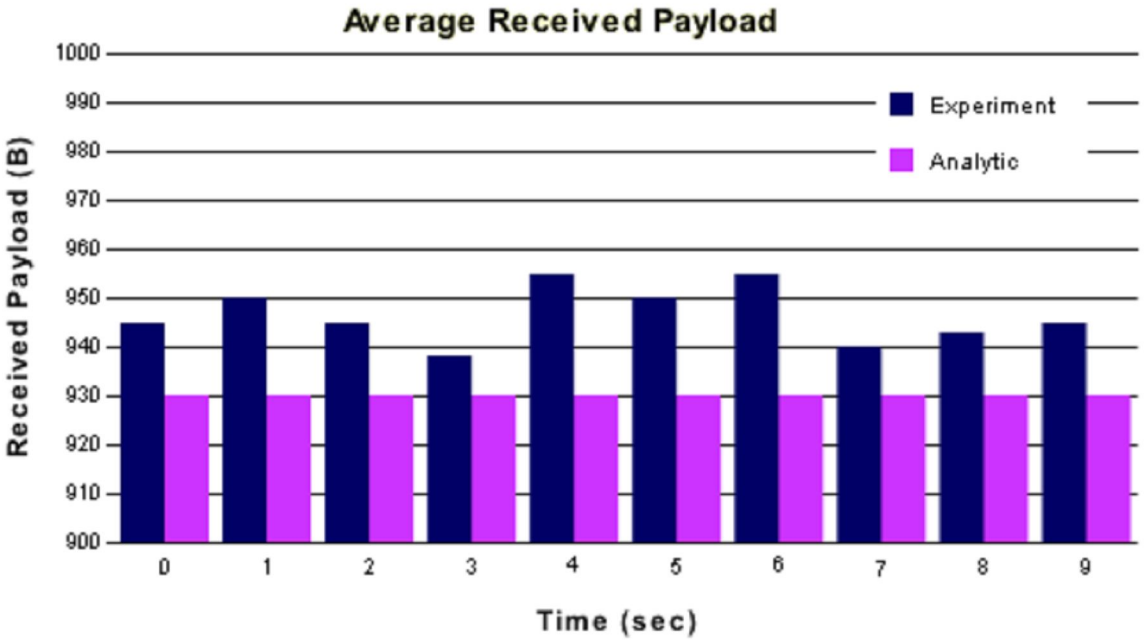

Figure 4.13 Experimental and theoretical results for overhead comparisons

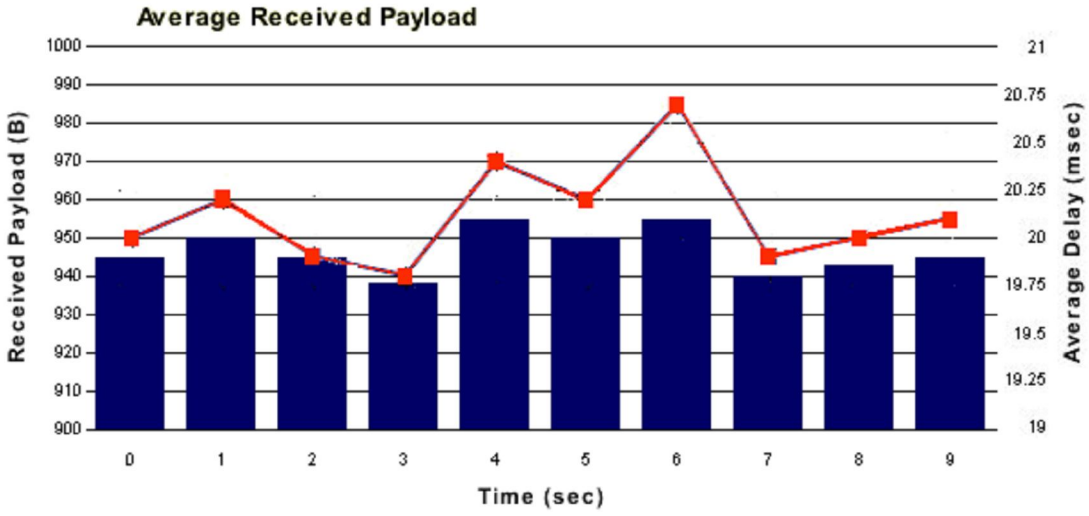

Figure 4.14 Experimental results for overhead and delay figures comparisons

Figure 4.13 presents the overhead comparisons between the analytical and experimental results. It can be seen that the analytical results yield 930 bytes of total overhead, whereas the experimental results show an average of 946.5 ± 0.3667 bytes (99% CI) with 3.19 bytes standard deviation and an average of 1.77% (and 2.69% maximum) increase compared to the analytical results.

Based on Figure 4.14, which is an overlap of payload size and incurred delay figures, there is a direct correlation between the increase and decrease of payload size and the incurred delay figures, which depend on channel quality and retransmission occurrences. In both cases, delay and overhead figures are within acceptable ranges.

4.6.1 The Effectiveness of the Cross-Layer Design

In this section a few experiments are run to compare the results of this system in the presence and absence of the cross-layer design. For this, two different sets of tests are conducted. In the first set, total delay figures are compared between normal cross-layer operation and a scenario where cross-layer feedback is disabled. The same test is done for the Impact Factor index being set to 1.0 and 0.45. The IF index 1.0 selects k = 251, G.711 for the audio and H.264 for the video codecs. The IF index 0.45 is associated to Zone 4 which selects k = 247, G.729 for the audio and H.263 for the video codecs.

To show the effectiveness of the cross-layer a new Labview module is added to calculate the jitter at the receiver. Since the calculation of delay was already part of the receiver's building-block, therefore for calculating jitter (variations in the receiver's delay), a slight modification is applied at the receiver. As mentioned, jitter is an important parameter that affects the quality of VoIP systems.

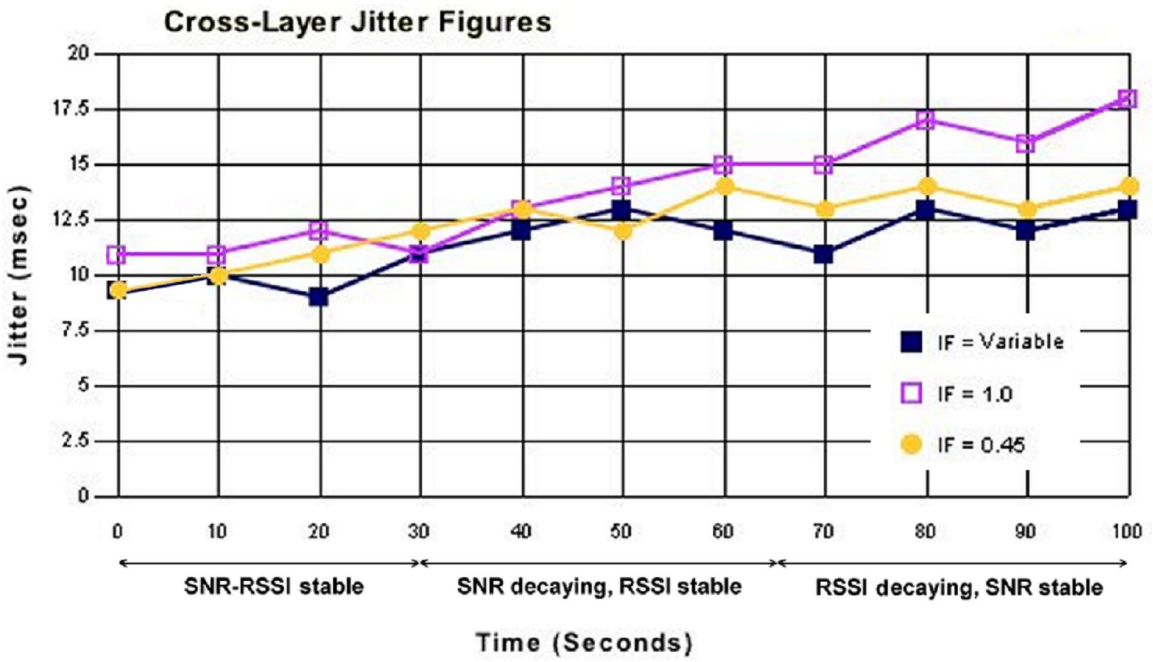

Figure 4.15 Experimental results for cross-layer jitter figures

Figure 4.15 shows a sample of the jitter figures in one test run, gathered from the three experiments; normal cross-layer function (causing variable IF), disabling the cross layer feedback, and setting IF to 1.0 and 0.45. During the 100 second run of the experiments, in the first 30 seconds, both SNR and RSSI are kept within relatively stable conditions (keeping channel noise level and the distances between the sender/receiver laptops from the AP constant). Then from 30th second to 65th second, RSSI is kept steady and SNR is decreased by increasing noise levels in the channel (e.g., turning on various wireless propagation sources). From the 65th second to the 100th second, SNR is kept constant and the RSSI is decreased by gradually increasing the distances between AP/sender/receiver.

The experiment is repeated for 500 times and yields the following average results: For variable IF, the average jitter is 11.47 ± 0.005 msec (99% CI), for IF = 1.0, the average jitter is 14.15 ± 0.007 msec (99% CI), and for IF = 0.45, the average jitter is 12.62 ± 0.006 msec (99% CI). The experimental results shown in Figure 4.15 show that for the duration in which SNR and RSSI are kept stable, for all three experimental scenarios (IF = variable, 1.0, and 0.45), the performances of the system, from the received jitter point of view, are consistent. As the SNR is degraded, the performance for the IF =1.0

deteriorate more noticeable than that of IF = 0.45 and for IF = variable, it is least impacted. The reason for the highest impact on the IF = 1.0 scenario is the fact that the system starts loosing the capability of handling constant deployment of G.711 and H.264 when the channel noise is on the rise. As the RSSI is degraded, similar outcomes are noticed as well and the best jitter values are seen for the IF = variable scenario where appropriate codecs are selected according to the channel quality conditions, which indicates the effectiveness of the cross-layer design deployed by this system.

4.7 Power Consideration

Power consumption is an important factor, especially for wireless systems, where the wireless devices are running on battery powers for prolonged period of times. Therefore every effort has to be considered to keep the power consumption within specific levels.

4.7.1 Suite-B on Intel T2500 Platform

The Suite-B algorithms were used in this thesis and their performances were tested using Crypto++ Library 5.6.0 cryptographic algorithms running on an Intel® T2500-2GHz CPU and 2 GB RAM using a Windows XP Service Pack 2 platform (laptop). We will show Suite-B's cryptographic algorithms power consumption figures based on the same platform, captured in the literature [209, 210, 211] and also discuss the power consumption figures on other handheld wireless systems running on limited battery powers, as their computational powers and batter resources may far be less than a laptop.

Table 4.33 includes the energy required to perform Suite-B cryptographic algorithms [209, 210, 211].

Table 4.33 Suite-B energy consumption figures

Algorithm	Energy Consumption	
ECDSA	154.3 (mJ) - Signing	207.33 (mJ) - Verifying
SHA-512	0.55 μJ/Byte	
AES-GCM	0.72 μJ/Byte - Encryption	0.93 μJ/Byte – Decryption
ARS (Adaptive RS)	0.66 μJ/Byte - Encryption	0.98 μJ/Byte – Decryption

For instance in the case of G.711 with 240 byte payload, the Suite-B algorithms energy consumption figure is: 362.68 mJ in 6 msec or 2.176 mW.

4.7.2 Handheld Processors and Platforms

Handheld devices usually deploy relatively less computationally complex processors to save power. The rule of thumb is, the less the computational power, the less the power consumption figures may be and since the wireless handheld devices run on limited batter powers, therefore it's vital to deploy mechanisms that require less amount of power for their operations.

ARM processors are the predominant processors used in many wireless handhelds [212]. The ARM features a 32-bit RISC (Reduced Instruction Set Computer) chip. ARM cores include family members of ARM 1 to ARM 11, StrongARM, XScale, and Cortext. Many vendors use one of the specific families of ARM processors in their systems. For instance, Blackberry Bold (9000 series) uses PXA270, which is an XScale ARM processor. Apple iPhone uses ARM1176JZF processor, which is a member of ARM11 family, and Nokia N-Gage uses the ARM946E processor, which is a member of ARM9E family.

Due to the processing and power restrictions of handheld devices, platforms with limited functionalities should run on these devices, including Windows CE, Windows Mobile (Pocket PC, v5, v6, and v7), and Linux (v2.6.x, etc.).

4.7.2.1 Suite-B on Handheld Platforms

Handhelds, such as the Blackberry Bold, which use PXA270 processors, have become increasingly popular, not only for their ubiquitous versatility, but also because of their security strengths. Security mechanisms are often involved with extra battery consumptions, delay, and overheads. The PXA270 ARM XScale family processor is

capable of performing 800 MIPS (Million Instructions per second) at 624 MHz clock speed, which is comparable to the 733 MIPS Pentium III [213, 214].

Reference [215] uses a testbed in which the PXA270 Processor is used (624 MHz) with 192 MB total RAM memory on Microsoft Windows Mobile 2003 platform. The comparisons are made to a PC platform (Acer TravelMate 4020 with Pentium M 725, 1.6 GHz, 400 MHz FSB, and 256 MB RAM). The cryptographic algorithms were created using C++. The encryption/decryption delays for an ECDSA 256 and the equivalent encryption scheme (AES-128), requires four times more time on the PXA270 compared to that of the Pentium M 725.

Reference [216] runs a test on Dell Axim X30 Pocket PC, which uses the same processor (624 MHz PXA270) on a 950 mAh battery and the MS PPC 2003 SE as the OS. For a file size of 2^{15} bytes, the encryption latency is 6.54 μs/byte and the decryption is 6.51 μs/byte, which is about 1000 times slower than the performance measure compared to our T2500-2GHz platform. Reference [216] also provides measurements for the HP iPAQ 4150 Pocket PC, which uses 400 MHz PXA255 processor on a 1000 mAh battery cell. The AES-128 scheme based on the block ciphers operated in an Electronic Code Book (ECB) mode and based on a 2^{15} byte file, requires 10.02 μs/byte and the decryption scheme requires 9.94 μs/byte. The power consumption figures are only provided for HP iPAQ 4150 Pocket PC, where AES on the same file size requires 2.524 μJ/byte and 2.505 μJ/byte for the decryption, which consumes approximately 4 times more power compared to our T2500-2GHz platform.

ARM7TDMI is another ARM-based processor used for cryptographic purposes [217]. There are numerous editions of ARM7TDMI, including: ARM7TDMI(-S), ARM710T, ARM720T, ARM740T, and ARM7EJ-S. The most powerful edition is the ARM7EJ-S, which operates at 133 MHz and its power consumption is at 0.14 mW/MHz [218].

Reference [219] uses a testbed in which ARM7TDMI is running at 60 MHz. The cryptographic codes were created and compiled by the tester and the results show that at 60 MHz clock speed, AES 128 and 256 had the throughput of 926 kbps and 680 kbps

respectively. SHA-256 and SHA-512 had the throughput of 640.8 kbps and 401.6 kbps and the delay of 1.524 μs/byte and 2.431 μs/byte respectively, almost 180 times slower than our T2500-2GHz implementation.

The ECDSA 224 signature generation time on the ARM7TDMI platform requires 142.6 msec, which is 54 times slower than our T2500-2GHz implementation.

4.7.2.2 Power tradeoffs of Suite-B on Handheld Platforms

As mentioned, the deployment of security mechanisms is always involved with extra battery consumption figures and in the case of wireless handheld devices that are expected to run on a single battery for a few days without being recharged, such extra power consumption figures may become critical and a few milliWatts difference between two cryptographic algorithms or the frequency of the deployment of two different algorithms, could have major impacts on the wireless handheld devices battery life-times.

Here are a few observations in terms of power consumption figures:

Limited memory and processing power: Due to power consumption limitations, many handheld devices are equipped with relatively weaker processors compared to those used in laptops and desktops. Most of these processors operate between the 20 – 800 MHz CPU clock speeds [215, 216, 217, 218, 219], therefore the number of instructions executed per unit of time are also lower than those observed in laptops and desktops. The same applies to the onboard memory systems. Most wireless handheld systems feature small RAM modules, often less than 256 MB [215].

Suite-B algorithms power consumptions versus file/message sizes: The size of the plaintext affects the performance (speed) of the Suite-B operation and as a consequence, the power consumption figures become dependent on the file/message size. As the file/message size increases, the consumption power figures per byte increases as well, however the average consumption power figures for the message per byte decreases.

Therefore in long run, the system consumes less power for performing Suite-B algorithms on larger file/message sizes. The actual dependency of the power consumption figures to the file size depends on the hardware platform (e.g., processor, CPU speed, RAM) and other software and protocol related parameters.

Battery life-time versus Suite-B algorithms deployment: The deployment of Suite-B cryptographic algorithms has a negative effect on the battery life-time. Again depending on the platform in use, the effect can be different. For instance; HP iPAQ 4150 uses a 3.7 V, 1000 mAh Lithium Ion battery and can work up to 12 hours on a full-charged battery. If we assume AES-128 is deployed (part of the Suite-B algorithms) and we assume AES is used 20% of the times. Such a deployment will reduce the battery-life from 12 hours to 9 hours 11 seconds. The same calculation would result 8 hours and 6 minutes of battery-life for the deployment of SHA-256. For ECDSA-256 signature that is used once every 10 second (as an example) on the ongoing traffic, it will reduce the batter life from 12 hours to 8 hours 9 minutes. As you notice, ECDSA-256 is a relatively more computational intensive algorithm and its deployment impacts the batter life greatly.

Suite-B algorithms deployment impacts on the throughput: Throughput is the performance parameter that will be impacted the most by the deployment of Suite-B cryptographic algorithms. As it was investigated in this thesis, the deployment of Suite-B did not impact the security/QoS performance of the system based on the laptop platform. However this is not the case for wireless handhelds, as it was mentioned in this section, Suite-B algorithms (i.e., AES, ECDSA, SHA) may be up to 1000 times slower than those operating on a laptop platform. For instance, reference [220] shows that the deployment of SHA-256 can decrease the throughput of the deployed platform (Actel CoreMP7 Fusion FPGA based on ARM7) almost 20 times. SHA-512 impacts the throughput almost 27 times. Using a 2 MB RAM, AES-128 and AES-256 reduce the throughput 11.37 and 15.43 times respectively.

Minimum handheld system requirement for Suite-B algorithms deployment: The delay, power consumption, and performance reductions figures associated to the deployment of

Suite-B algorithms in wireless handheld devices should be considered based on the battery-life and minimum performance requirements (e.g., mandated by the running applications). Therefore the required minimum system depends on the type of application, minimum bandwidth, maximum delay, and maximum extra consumption power that the system should be able to handle.

4.8 Existing Solutions

The system designed and discussed in this thesis is unique and there is no system to-date that provides the same services/functionalities. However in this subsection, existing systems and solutions with similar functionalities will be considered.

Reference [221] introduces a web-based application system (WebIBC), which integrates the a public-key cryptographic scheme into the ongoing traffic. WebIBC is a software-based system that provides encryption, integrity, and integrates public key cryptography and integrates them into the web applications. The public key system in WebIBC is provided by an identity-based cryptography. This system features a 512-bit Elliptic Curve Cryptography based on ECDH and ECDSA, which are implemented using a JavaScript Crypt Library. The WebIBC computation engine features other cryptographic building blocks, including AES-128, SHA-1, and a big integer module (MULtiply "MUL") algorithm. This system accepts 3 different block sizes of data; 0.5, 2.0, and 10.0 KB. The resulted overheads produced by the system are all over 1000 bytes per packet, which are higher than the overhead figure in the worst-case scenario in system discussed in this thesis. A more detailed comparison is given in Table 5.1.

Reference [222] proposes a secured lightweight architecture (SA2pMP), which is also software-based (implemented using Java ME "Micro Edition") and deploys a lightweight cryptographic scheme for a two-party mobile secure payment system. This system is a combination of a symmetric key and a public key systems based on ECDSA-192, AES-256 and other multi-factor authentication mechanisms. This system offers non-repudiation, authentication, integrity, and confidentiality services and is efficient when

implemented in a mobile platform without causing additional delays and is well-suited to protect two-party payment transactions over a resource-limited mobile network. The SA2pMP is emulated using three different software systems, including: Java Wireless Toolkit 2.5.2 for CLDC, Nokia S60 rd Edition, and Sony Ericsson SDK 2.5.0.3 Z800 emulators. The simulations were running on the IBM IntelliStation M Pro PC, with Pentium 4 CPU 2.80GHz and 2GB RAM and the operating system is Windows XP Professional SP3. The hardware platform used in these simulations has higher benchmark values compared to the hardware platform used for in this thesis (Intel® T2500-2GHz CPU and a 2 GB RAM on a Windows XP Service Pack 2 platform) [223].

All three emulators exhibit signature sign/verify times of more than 700 msec, which are much higher than the delay figures observed in the system discussed in this thesis. A more detailed comparison is given in Table 5.1.

Reference [224] discusses a secure architecture (software-based implementation) for Vehicular Ad-Hoc Network (VANET) applications that is built around an authentication protocol, including the related computational and experimental analyses. This system is based on a security protocol that is optimized for VANET scenarios and is not only capable of providing communication means between ad-hoc entities, it also offers strong security mechanisms. The performance measures are done on a Centrino machine with the clock speed set at 1.5 GHz and the security algorithms deployed are based on ECDSA-192, ECDSA-256, and AES-128. For ECDSA-256, the signing time is 3 msec and the verification time is 4.2 msec, which are higher compared to the signing and verification delays measured in our system (2.63 and 3.89 msec, respectively). According to the Reference [223], Centrino-based platforms have higher benchmark values compared to T2500 platform.

Depending on various authentication schemes used in reference [224], the message sizes may change from 100 to 800 bytes. These overhead figures are relatively lower compared to those measured in the system considered in this thesis due to the fact that no multimedia data are included in the message structure discussed in reference [224]. The

maximum delay figures are also variable, ranging from 20 ms to 50 ms, which are higher than the maximum delay figure calculated in our system (i.e., 20.12 msec). A more detailed comparison is given in Table 5.1.

Therefore by analyzing the existing systems and comparing the performance measures, delay/overhead figures, and/or the supported capabilities, our system outperforms the existing solutions in various levels and criteria. Table 5.1 sums up the comparison measures and criteria between the existing solutions and our system.

Table 5.1 Comparisons between the existing solutions and our system

	Suite-B Compatibility	Delay Figures (msec)	Overheads (Bytes/Packet)	Security Features	Multimedia Compatibility
WebIBC [221]	ECDH-ECDSA-512, AES-128, SHA-1	U/A	Overhead > 1000	Authentication, Authorization, Integrity, and Non-Repudiation	HTTP- and Web-applications
SA2pMP [222]	ECDSA-192 AES-256	E2E Delay: U/A Signing Delay: > 700	Information Unavailable	Authentication, Integrity, Non-Repudiation, Confidentiality	Not Applicable
VANET [224]	ECDSA-192, -256	E2E Delay > 20, Signing Delay > 7.2	Overhead: 100-800	Authentication, Integrity and Non-Repudiation	Not Applicable
Our Solution	ECDH-ECDSA-256, AES-128, SHA-256	E2E Delay < 20.1 Signing Delay: 6.5	Overhead < 955	Authentication, Authorization, Integrity, and Non-Repudiation	Full Multimedia (Voice and Video)

4.9 Conclusions

In this chapter, we reflect the analytical and experimental results based on the QoS and security models that were presented in chapter 3. We showed that the performance parameters specified in chapter 3 (e.g., end-to-end delays, packet overheads, etc.) are within acceptable ranges and the analytical and experimental data match.

In particular, we discussed four operational modes:

1. Method 1: In this method, Suite-B at the application layer is used packed in the core UDP payload. No security options at the Network and MAC layers are available.
2. Method 2: In this method, Suite-B at the transport layer is used.
3. Method 3 In this method, Suite-B at the network layer is used based on IPSec.
4. Method 4: In this method, core UDP payload (mentioned in Method 1) is used with a MAC layer encryption technique (e.g., WPA-AES).

The thorough analyses showed that when applying application layer Suite-B algorithms on full multimedia traffic (voice, video, and text), the payload will be 986 bytes and the end-to-end delay is 19.64 msec, which is acceptable for VoIP purposes.

Experimental results using Labview setup show similar results with a maximum 2.44% tolerance for the end-to-end delay and 1.77% tolerance for the received payload overhead.

We also considered Suite-B algorithm deployment in handheld devices, which mostly use one of the ARM processors families. Due to the limitations in power and computational capabilities, the performances of these handheld have proven to be much less compared to the performance of our laptop T2500-2GHz platform. However the delays and power consumption figures resulted from the Suite-B deployment are within acceptable ranges and are not causing computational and delay bottlenecks.

Chapter 5
Conclusions

In Chapter 2, we considered current and past approaches to wireless multimedia transporting security and QoS services of the following areas: QoS, Security Traffic classifications, Wireless multilayer approach, Wireless cross layer design, and Non repudiation multimedia-based wireless systems., which paved the path to introducing the QoS and security models.

In Chapter 3, we introduced QoS and security models and we discussed the design and implementation of an application layer non-repudiation wireless system, which featured a cross-layer technique validated by a traffic classification analysis. The main feature was to support full-multimedia capabilities (e.g., text, voice, and video) while offering highly secure mechanisms, supporting various algorithms, including a non-repudiation technique. We used real traffic sniffing tools to capture and analyze both wireless and wired traffic payloads. Then we applied classical traffic classification tools to find confidence interval for the flow durations and packet sizes for both wired and wireless traffic payloads. The average packet size for wired traffic was 652 and for wireless traffic was 197 bytes.

The end-to-end communication between end-point entities (e.g., patient, doctor, etc.) starts with a key agreement scheme. The deployed cryptographic algorithms are based on Suite-B and the key agreement algorithm offered by Suite-B is Elliptic Curve Diffie-Hellman (ECDH) scheme. Following the key agreement algorithm, traffic flows between Party A and Party B will include several Suite-B security parameters, which including: AES-128 for encryption, AES-GCM (Galois/Counter Mode) for authentication and encryption, which is a 128-bit block cipher. ECDSA-256 offers digital signature functionality, and SHA-512, offers a hashing mechanism. These mechanisms are initially deployed at the application layer, which are accompanied by a cross-layer adaptive Forward Error Correction scheme, based on Reed Solomon algorithm at the application

layer. Figure 4.1 shows the cross-layer QoS/Security model used in this system. The Suite-B algorithms were also considered at the transport and network layers.

In the application layer, multimedia communication data (i.e., voice, video, and text) are in terms of messages and are fed into the adaptive encoder, which is also fed from the lower layer security/QoS modules. The encoded multimedia data is fed into the AES-GCM and the output is used as an input to the ECDSA algorithm and SHA function.

The system operates using cross layer parameters (CLPs) feedbacks from lower layers. These include RSSI (Received Signal Strength Indicator) and Signal to Noise Ratio (SNR) at the physical layer, WMM (Wireless Multimedia) at the MAC layer, DSCP (Differentiated Services Code Point) at the network layer, and 24 bits related to security capabilities, which adds up to 48 bits for the CLPs (Cross Layer Parameters).

The CLPs were used in the Adaptive Forward Error Correction (AFEC) based on the Reed Solomon schemes (ARS).

The experimental results were based on the Crypto++ Library 5.6.0 cryptographic algorithms run on an Intel® T2500-2GHz CPU and a 2 GB RAM using a Windows XP Service Pack 2 platform and on a Labview testbed, which showed similar results with a maximum 2.44% tolerance for the end-to-end delay and 1.77% tolerance for the received payload overhead.

The rest of the thesis talked about the security analysis that discussed the strengths and weaknesses of the Suite-B algorithms used and the possible attacks on the system.

Table 4.11 shows the application layer delay and overhead figures related to Suite-B algorithms. Figure 4.11 shows the experimental results for the overhead and delay figures.

Therefore the theoretical analysis, as well as experimental results, show consistent outcomes in terms of overhead and delay figures. The end-to-end delay figures are

bounded to approximately 20 msec, which works well for most VoIP systems and are within the acceptable range. The overhead figures are between 100 to 1000 bytes, which work well in most wireless networks, however high overheads (above 200 bytes) may pose as a challenge for networks undergoing heavy traffic periods or congestions, in which case the cross-layer system will start to affect the codec schemes to reduce the overhead accordingly.

We also studied the Suite-B algorithm deployment in handheld devices, which mostly use one of the ARM processors families. Due to the limitations in power and computational capabilities, the performances of these handheld have proven to be much less compared to the performance of our laptop T2500-2GHz platform. However the delays and power consumption figures resulted from the Suite-B deployment are within acceptable ranges and are not posing as bottlenecks.

5.1 Details of the Contributions

The contributions of this thesis are listed in the following categories:

Application Layer Provisioning: In this thesis, we designed a system in which various QoS and security related parameters are exported from different layers and imported at the application layer. In this thesis, we covered details on a well organized QoS and security enabled application layer-based system, which incorporated multimedia-enriched traffic in the presence of various security-enabled data with a non-repudiation support.

QoS-based Cross Layer Design: In the structure of the proposed system, a cross-layer design approach was deployed, which served for both application layer-based QoS and security purposes. In such a cross-layer QoS-based system, a number of QoS-related parameters (e.g., Received Signal Strength Indicator “RSSI”, Signal to Noise Ratio “SNR”, Wireless Multimedia, “WMM”, and Differentiated Services Code Point “DSCP”) were gathered from three layers and imported at the application layer. The current values of these three parameters defined the quality of the wireless link and were used to

determine the quality of the multimedia encoder. This way, an adaptive multimedia encoder/decoder system was designed, which was used to adapt its coding quality to the quality of the environment and to adjust the encoder bit-rate and the integrated Forward Error Correction (FEC) scheme accordingly. The values of these three parameters were transmitted alongside the rest of the data to the receiver and the receiver used them to determine the correct decoding procedures.

Security-based Cross Layer Design: For this, a number of layered-based (e.g., physical, MAC, and network layers) security parameters were gathered from various layers, which were accompanied with other gathered QoS-related parameters and fed into an application layer security-based module. In particular, we deployed a multilayer Suite-B cryptographic system to address various security requirements in the application, transport, and network layers and to study the impacts to the overhead and delay figures of such a deployment.

Multimedia-based Adaptive Non-Repudiation System: As mentioned, the cross-layer parameters were imported at the application layer from lower layers. An adaptive mechanism was presented at the application layer, which processed the cross layer information and its outputs were directly used in the encoder (at the sender), decoder (at the receiver) systems, and in the Forward Error Correction (FEC) scheme, adapting to network conditions.

Following the functions of the encoder and the FEC module, Suite-B algorithms were applied to the multimedia traffic.

Finally, a thorough analysis was performed to investigate the performance of this system under various security and QoS conditions.

Security System Analysis: The security model of this system was discussed in chapter 3. This included a thorough consideration of the security schemes, mechanisms, and protocols used as well as security analyses of the system.

QoS Support: Through analyses and experimental results using C++ based cryptographic codes and Labview testbeds, QoS capabilities offered by the system were investigated under various traffic flows, containing; multimedia data, hashed data, cross-layer information, and digital signature.

Analytical versus Experimental Results: The average analytical value for end-to-end delay was shown to be 19.64 msec on average on 930 byte overhead when video, audio, and text were transmitted simultaneously. According to the experimental results, the average delay was 20.12 msec with the standard deviation of 0.081 msec, which resulted an average 2.44% (and 5.40% maximum) differences between analytical and experimental results, which validated our assumptions. The fluctuations in the experimental results were due to channel variations and retransmissions.

Figure 4.6 presented the overhead comparisons between the analytical and experimental results. It was shown that the analytical resulted yield 930 bytes of total overhead, whereas the experimental results showed an average of 946.5 bytes with 3.19 bytes standard variance and an average of 1.77% (and 2.69% maximum) increase compared to the analytical results.

Power tradeoffs of Suite-B on Handheld Platforms: The deployment of the Suite-B algorithms were investigated in wireless handheld platforms, mostly using ARM processors. It was shown that the reduced throughput, added power consumption figures, and the added delays where much more than that of the laptop platforms. Therefore depending on the application, delay, throughput, and power consumption limitations, minimum system requirements may be required to provide minimum services.

5.2 Discussions

This section is dedicated to the discussions on the applicability and other features of the designed system.

Applications – The two main features of this system are QoS (e.g., minimum bandwidth, maximum delay, etc.) and security (in particular, non-repudiation) and this system can be used in scenarios where the actions of any of the involved parties are to be logged for future reference. One of these scenarios, as mentioned, is in the medical field where a doctor may require such a system to record his/her actions in regards to the patients, as well as the patients (in particular, remote patients) who may require this system to ensure the authenticity of the medical instructions they receive.

Another application where this system can be used in is in the stock exchange market, where both the buyers and the brokers require such a service to provide assurance against the denial of the actions taken by the opponents.

The third application that can rely heavily on this system is in remote education, where students may be taking part in remote monitoring examinations and the administrative staff can handle remote examinations in a different geographical location by checking the digital signatures of the students taking part.

Adoptability – A major component of this system is based on an application layer provisioning scheme, which can be implemented purely using software codes with no hardware components. Therefore this system can be adopted by any wireless handheld (e.g., BlackBerry and iPhone) to provide non-repudiation services in the presence of QoS measures. There are already a few applications running on BlackBerry/iPhone devices offering similar functionalities (e.g., S/MIME Support Package, SolidPass, etc.) however they do not offer simultaneous QoS/security provisioning services.

Required Layered Hardware Modifications – To evaluate the required hardware modifications based on the four methods of deployments discussed in this thesis (application, transport, network, and MAC layers), a closer look to the components deployed in each method reveals the hardware modifications required in each method (at each layer). At the application layer (method 1), as mentioned in the adoptability subsection, all components (i.e., audio/video codecs, Suite-B algorithms, and A-FEC) are software-based, therefore no hardware-related modification is required.

At the transport layer (method 2), RFC 5246 (The Transport Layer Security "TLS" Protocol Version 1.2) and RFC 5430 (Suite B Profile for Transport Layer Security "TLS") [120, 179] cover the entire Suite-B security protocol handshakes and the application data transfer at the transport layer. Therefore no new modification is required beyond RFCs 5246 and 5430.

At the network layer (method 3), again RFC 4306 (Internet Key Exchange "IKEv2" Protocol) and RFC 4869 (Suite-B Cryptographic Suites for IPSec) cover the entire security protocol handshakes and application data the transfer at the transport layer. Therefore no new modification is required beyond RFCs 4306 and 4869.

Method 4 (MAC layer), as mentioned, uses method 1 and the conventional MAC layer security (WPA/WPA2) schemes, therefore no hardware modifications are required for this method either.

Therefore for the deployment of the entire system based on the methods described, no additional hardware-based modifications are required.

Limitations of the System – The following limitations are to be considered when deploying this system: *Codec bit-rate limits* – The lowest bit-rates for the audio and video codecs are 5.3 kbps and 24 kbps respectively and the highest bit-rates are 64 kbps for both codecs, which may be adequate for a relatively good quality voice reproduction, however may not result acceptable video reproductions if certain visual details are required. *Channel/signal qualities change rate* – In the current implementation, cross-layer parameters are probed once every second at the application layer. Therefore if the channel/signal qualities change more rapidly, the system may not operate as efficiently as less rapid signal/channel quality changes.

Security/QoS Interactions – The only interactions security and QoS have on one another is during the cross-layer probing. The cross-layer design imports both QoS and security related parameters at the application layer and the A-FEC and codec qualities are

determined based on the imported cross-layer parameters. However the security protocols and the Suite-B algorithms are performed independently from the QoS side of the system. For instance when the Bit Error Rate (BER) increases due to the increase of the noise level, this impacts the SNR value, in which case will decrease the SNR's individual Impact Factor and as a consequence, the total IF index will be impacted, which impacts the A-FEC scheme and deployed codecs. However this does not impact the operations of the security side of the system.

5.3 Future Work

For the future work, the following areas may be investigated:

Biometric Authentication Measures for Non-Repudiation Techniques: Biometric Authentication (BA) and Biometric Encryption (BE) schemes should be used to transform user initiated biometric data into untraceable biometric information. A few biometric measures used for authentication and encryption/decryption mechanisms, include: Face, fingerprints, iris colors and patterns, hand/finger geometries, voice, dynamic signature, keystroke dynamics, gait, and DNA schemes. The information contained in biometric measures is unique and irrevocable.

Therefore the future direction may involve binding physical characteristics of a biometric feature (e.g., voice, fingerprint) into encryption keys, which could be used to encrypt and decrypt messages.

Multimedia-based Non-Repudiation Services using Wireless Handhelds: More and more applications and services are being offered through wireless handheld devices. The same analytical and experimental study included in this thesis could be carried out involving a wireless handheld device instead of a laptop system. The results could be used to provide general guidelines for the deployments of such devices.

mLearning and mHealth on 3G networks: Following the same future direction for deploying wireless handhelds, Smartphones running on 3G (3rd Generation Mobile

Networks) could also be used and the same On the same analytical and experimental study included in this thesis could be carried out using such a platform/network. Currently there is a major interests in the academia to investigate the effectiveness of mLearning (Mobile Learning) and mHealth (Mobile Health), which could be integrated with a similar study that was conducted in this thesis.

Appendix A

A.1 Digital Signatures

A.1.1 Digital Signature Algorithm (DSA)

The DSA system is comprised of three parts: Key generation, signing, and verifying, which are discussed here [225]:

DSA key generation: A 160 bit prime number q is selected and a random number t is selected so that it satisfies the condition of $0 \leq t \leq 8$. A random number p is selected so that it satisfies these two conditions: $2^{511+64t} < p < 2^{512+64t}$ and q is divisible by (p -1). Select α as Z_p^* generator and set $g = \alpha^{(p-1)/q} \bmod p$ and select a so that it satisfies the condition: $1 \leq a \leq q$ -1 and compute $b = g^a \bmod p$. The public key is (p, q, g, b) and the private key is a.

DSA signing: Assume the message m is to be signed using DSA. The number k is so selected that it satisfies the condition: $0 < k < q$ and assume $H(m)$ is the hash function applied to the message m. For the hashing function, Federal Information Processing Standard (FIPS) recommends the use of SHA-1. Then r and s are computed as follow: $r = (g^k \bmod p) \bmod q$ and $s = k^{-1} (H(m) + a.r) \bmod q$. The signature is ($r$, s).

DSA verification: To verify a DSA signature, first the following condition should hold: 0 < r < q and 0 < s < q. Then we compute the followings: $w = s^{-1} \bmod q$, $z = w \cdot H(m) \bmod q$, $y = r \cdot w \bmod q$, $v = (g^z \cdot b^y \bmod p) \bmod q$. If $v = r$ it validates the signature. If not then the signature is invalid.

A.1.2 RSA

The RSA system is comprised of three parts: Key generation, signing, and verifying, which are discussed here [225]:

RSA key Generation: Two randomly large distinct same-size prime numbers; p and q are selected. We compute $n = p.q$ and $\phi = (p - 1).(q - 1)$ and select a random integer e, which satisfies the condition: $1 < e < \phi$, and gcd $(e, \phi) = 1$, where gcd is the greater common divisor finder function. Both n and ϕ are sufficiently large enough and e is randomly selected to ensure RSA function is a one-way trapdoor function (the private key as being the trapdoor). Then we compute d with the condition that: $1 < d < \phi$ and $e.d \equiv \text{mod } \phi$. The public key is (e, n) and the private key is a (d, p, q).

RSA signing: Assume the message m is to be signed using RSA. The condition that $0 < m < n$ must hold and then we computer $s = m^d \text{ mod } n$, where s is the signature. Then (s, m) are sent to the receiver.

RSA Signature Verification – The receiver uses the (e, n) public key and computes $s^e \text{ mod } n$, which is equal to $(m^d \text{ mod } n)^e \text{ mod } n = m$.

A.1.3 Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA Key Generation Algorithm: *Alice* follows the following steps:

1. Select a random integer number of d, which is a member of the set: $[2, n - 2]$.
2. Calculate $Q = P \times d$.
3. *Alice*'s private and public keys are: d and (E, P, n, Q)

ECDSA Signature Generation Algorithm: *Alice* signs the message m through the following algorithm:

1. Select a random integer number of k, which is a member of the set: $[2, n - 2]$.
2. Calculate $k \times P = (x_1, y_1)$, a point on the elliptic curve, and $r = x_1 \text{ mod } n$. Assuming x_1 is a binary number representation if $x_1 \in GF(2^k)$. If $r = 0$, then go to Step 1.
3. Calculate $k^{-1} \text{ mod } n$.
4. Calculate $s = k^{-1} (d.r + H(m)) \text{ mod } n$, where $H(m)$ is the SHA value of m. If $s = 0$, then go to Step 1.

5. (r, s) pair is the message m's signature.

ECDSA Signature Verification Algorithm: *Bob* verifies Alice's signature, message pair (r, s, m), through the following algorithm:

1. Calculate $f = s^{-1} \bmod n$ and $H(m)$.
2. Calculate $g_1 = H(m) \cdot f \bmod n$ and $g_2 = r \cdot f \bmod n$.
3. Calculate $P \times g_1 + Q \times g_2 = (x_1, y_1)$ and $v = x_0 \bmod n$.
4. If $v = r$, then the signature (r, s, m) is accepted and verified.

A.2 Traffic Classifications

A.2.1 Traffic Classification Parameters

In this section we introduce a number of network traffic parameters. These parameters are mostly considered in the study of packet and traffic classification techniques.

A.2.1.1 Packet Size

Packet size is one form of traffic classification. Most of the traffic volumes on the Internet can be categorized into either, very small (mouse) packets or very large (elephant or heavy tailed) packet sizes. The large packet size is usually associated with higher usage of a link. Basically 20% of the connections on the Internet are responsible for 80% of the traffic passing through a single point [226, 227, 228], mostly via elephant packets.

Zipf's law is used to specify a more generalized form of packet size traffic classifications. In the packet size scenario, Zipf's law characterizes the frequency of occurrence of certain packet sizes as a function of its rank in the frequency table [229]. This means that there exists an imbalance in the network due to the fact that 20% of the connections carry 80% of the traffic payloads and the rest of the 80% of the connections are for small packet traffic payloads.

Traffic Engineering (TE) [230] is a term applied to a systematic process in which traffic flows are arranged in "classified" groups to simplify the transmission management throughout networks and decrease the probability of congestions. TE, by nature, is well positioned to deal with very large volume traffic payloads through the aggregation of traffic. However TE tends not to perform as efficiently when dealing with mice flows. The drawback to TE in regards to traffic classification is the fact that traffic in large and random environments (e.g., the Internet) would exhibit volatility in several flow specifications, namely; volume and bandwidth [230]. Fluctuations of these network parameters reduce the efficiency of TE in the process of traffic classifications.

In many cases, flows exhibit inherent bandwidth fluctuations. As mentioned, this creates complications in the traffic classification process, leading to frequent reclassification, thus reduction in the classification performance. These fluctuations are due to the following factors [230]:

Connection termination following the link exhaustion - The duration of a connection can be modeled as a stochastic variable dependant on the following parameters [231, 232]: The protocol in use, the current (k^{th}) connection arrival time, the current connection (k^{th}) time duration, and client/server performance metrics (e.g., round-trip delay, client delay, server delay) for client/server based applications such as FTP.

The effects of these parameters contribute to the creation of a median time for the flow. This median time for elephant flows (also known as *heavy-hitters*) will be higher since according to reference [233], the longer the connection duration (heavy-hitters), the higher the probability for the link to continue its connection.

Burstiness Effect - Multimedia traffic, especially video data, can be affected by the burstiness of traffic flows, reflected by a number of parameters, such as [233]: Peak-to-average ratio (PAR) and the temporal auto-covariance function (ACF).

Burstiness is a time sensitive parameter, which can be modeled as a stochastic variable. It is more probable to be an issue in heavy-hitter connections compared to mouse flows.

Bandwidth Fluctuations - Bandwidth fluctuations occur relatively frequently in wireless networks compared to wired networks. In wired networks, bandwidth fluctuations may happen due to various reasons, such as, a sudden increase of user demands or a congestion period.

The reasons behind bandwidth fluctuations in wireless networks are mostly related to the PHY and MAC layers issues, including: handoff and handover between Access Points (APs), limitations of available bandwidth in multi-user environments, physical limitations

(e.g., reflections, refractions, multipath), vulnerability to various interferences, and the dependency of performance to the client's distance to the server (AP).

A.2.1.2 Heavy Hitters (Elephants) versus Mice Packets

Heavy hitters can be identified by both their large packet sizes and long duration connections. It has been presented in the literature [234, 235] that there are strong correlations between the rates of streams and their packet sizes mainly based on the protocols in use.

In wired connections, from a packet size point of view, packets are usually between a few tens of bytes up to 1514 bytes. Depending on the Maximum Transmission Unit (MTU), large files being transmitted are usually broken down into various fragments. Based on captured real traffic payloads, we notice that control packets (packets containing control commands), which do not usually have any data payloads, are less than 200 bytes. Data packets are usually above 200 bytes.

Wireless traffic starts from 14 bytes (e.g., ACKs, CTS) with no data payloads, up to 1530 bytes, in IEEE 802.11 a/b/g networks, which is a limit by which fragmentation occurs. Based on our real traffic analysis, we label packets with over 400 bytes in lengths as heavy hitters.

Mice Flows - Mice flows are those with relatively low sizes transmitting for a short duration. The duration limit is less than the time required for the accumulation of 10 KB data and the packet sizes are usually less than 500 bytes each.

Elephant Flows - Elephant flows on the other hand are flows, which usually last more than a few minutes carrying relatively large packet sizes (often larger than 1 KB each). Therefore for a typically elephant flow, more than 3 MB of data on average is accumulated compared to 10 KB in the mice flow case.

A.2.1.3 Duration

The time duration of packet streams is another form of packet classification. Depending on the application, a short lived packet stream can last from a few microseconds up to a few minutes. Long-lived packet streams, on the other hand, can last from a few minutes up to several hours. Statistics [234, 235] show that there are direct links between larger packet sizes and longer durations. Based on captured real traffic from connections conveying multimedia data, most control packets (e.g., Beacons, ACKs, CTSs) are light connections (tortoises) and other packets forming connections (e.g., connection requests, confirmations, data transmission, acknowledgement teardowns), are considered heavy hitters (dragonflies).

A.2.2 Traffic Analysis – Flow-based Approach

In an IP network, a *flow is defined* as a unidirectional series of IP packets with unique source/destination addresses, port numbers (assuming TCP or UDP to be the transport layer protocol) and protocol number [234, 236, 237].

The main focus of this section is to discuss application specific classes of traffic. However it is important to talk about a few basic and fundamental definitions first.

Four main parameters associated to every flow are: *size, duration, rate,* and *burstiness*. There is a correlation between size and rate and in regards to small/medium flow sizes, due to different timeout mechanisms, the strong correlation between size and rate is more likely a pervasive artifact. Such an argument might require the usage of a larger packet size or the deployment of a larger initial window to improve TCP performance. This will increase the chance that more data is sent in one round trip time “RTT” before the timeout occurs.

As mentioned, there is a strong correlation between flow size and rate. Therefore size can be chosen based on the availability of bandwidth [238].

A.2.2.1 Flow-Level Metrics

Reference [236] classifies flows according to their *sizes, durations,* and *inter-arrival times*. These are defined as followed [236]:

A.2.2.1.1 Flow Size

Flow size is the total number of bytes transferred between a server and a wireless client during a single connection. From the client point of view, it does not matter if a new server provides services (handover happens with a new IP address) while the connection is still ongoing. However this measurement is usually done per server/client pair [239, 240, 241].

Peer-to-Peer (P2P) networking has gained much popularity in the recent years. The statistical flows for both P2P and Internet have well been modeled and bounded between Pareto and Weibull distributions [236], and their probability density functions (pdf) can be derived as followed, as mentioned in Equations A.1 and A.2.

$$f_{WEB}(S) = \begin{cases} 0.26S^{-0.62}e^{-\left(\frac{S}{2.7}\right)^{0.38}} & : S \le 30KB \\ \frac{3.33}{S^{2.05}} & : 30KB \le S \le 5MB \\ \frac{600466}{S^{3.35}} & : S \ge 5MB \end{cases} \qquad \textbf{Equation A.1}$$

$$f_{P2P}(S) = \begin{cases} 0.63S^{-0.19}e^{-\left(\frac{S}{1.36}\right)^{0.81}} & : S \le 4KB \\ \frac{0.0548}{S^{0.35}} & : 4KB \le S \le 10MB \\ \frac{7034}{S^{2.42}} & : S \ge 10MB \end{cases} \qquad \textbf{Equation A.2}$$

The distribution for the web-flow sizes includes a long-tailed distribution. A probability distribution is called long-tailed when high probability regions are far from the median or mean.

A.2.2.1.2 Inter-Arrival Time between Flows

This is the time between any two consecutive flow arrivals. Inter-arrival times in flows are practically independent from each other and are distributed exponentially according to Poisson process. IP traffic on top of TCP or UDP, also has uncorrelated inter-arrival flow times (also true in regards to the flow lengths), therefore it can be modeled by a combination of algorithmic scaled normal distributions [242].

A.2.2.1.3 Flow Duration

This is calculated from the start of the initial handshake of the flow until the last data packet has left the sender followed by the tear-down of the link related to the flow. At this level we also deal with both mice and elephant flows.

In the section related to the flow size, a time range for both mouse and elephant flows is calculated. According to the definition, as mentioned, a typical mouse flow can be as short as a few microseconds (based on current 802.11 bandwidth limit of 54 Mbps) up to several minutes. A typical elephant flow lasts from an hour to several days and could transmit up to several thousand terabits of data in a single flow.

A.2.2.1.4 Flow Fluctuation Patterns

In general, one can categorize flow fluctuation patterns as: *Slowly varying continuous data flow, fast varying continuous data flow, traffic with common periodic trends, short-lived bursts,* and *noise*.

Slowly varying continuous data flows are long-lived connections generated from a steady source where relatively high correlation among successive data is observed. Therefore, only small variations are observed in a short period of time. An example would be the data transmitted from thermal sensors.

Fast varying data flows are long-lived flows with rapid fluctuates over a relatively short period of time. In these types of flows, high variations are observed with low correlations among successive data. An example of this would be data transmission across a busy LAN.

Common periodic trends are long-lived traffic patterns which are observed to be periodic in nature, such as web server traffic and scheduled backup data.

Short-lived bursts are also a part of most data network traffic. As mentioned before, a long established busy LAN connection may exhibit fast varying data flow, however over a short period of time, such a connection may include short-lived bursts resulting from rapidly fluctuating traffic payloads. A burst can be characterized as a fluctuating data stream over a relatively short period of time.

Background noise is an inevitable part of any network traffic. A high SNR (Signal-to-Noise Ration) value ensures relatively high level of signal and low level of noise.

It should be noted that the network traffic categories mentioned are applicable per flow and per aggregation of flows.

A.2.3 Traffic Control

Depending on the nature of the flows, either the majority being mice, elephant, or a combination of both, networks will deal with conditions differently. For instance if the majority of the flows are mice and the network has undergone congestion periods, dropping packets will do little in dealing with the congestion control. In general, such a

network will exhibit random behaviors with high adaptability to sudden changes, which can be a favorable issue for time-sensitive applications. Telnet and HTTP transfer streams tend to be of mice flow type [237].

In a network where most of the flows are elephant, depending on the protocol in use, it can be tolerant against congestion, in particular if the majority of the traffic is based on TCP, as TCP features a built in congestion avoidance mechanism. TCP (e.g., FTP applications) and UDP (e.g., video applications) flows are examples of elephant flows [237].

Flow duration increase may increase the Long Range Dependence (LRD, also known as long memory, measured by the Hurst parameter) as well. LRD is an autocorrelation value of a data stream, which approaches a constant value (normalized to 1) as the number of data bits increases. If the limit in equation A.3 exists for a real number of *r*, then α (*s*) is the autocorrelation function and X_t is the LRD stationary process (Figure A.1 (adapted from [243]), equation A.3).

$$\lim_{k\to\infty} \alpha(k) = r; \qquad k = 1, \quad 2, \quad \ldots \qquad \textbf{Equation A.3}$$

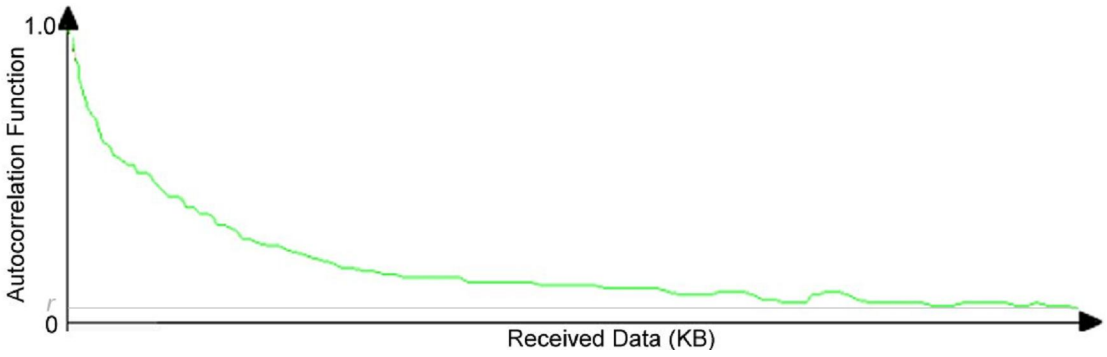

Figure A.1 Autocorrelation function in an elephant flow merging to the value *r*

Therefore for a typical elephant flow, equation A.3 should hold. The following definitions are related to LRD:

Hurst parameter - is an indicator parameter, which increases as the traffic volume and traffic burstiness increase.

Self similarity - is a statistical property, fractal-like, to examine produced data for similar patterns over a scale of time. Some of its properties include: slow decaying variance, long-range dependence, and Hurst effect.

A.2.4 Traffic Analysis – Application-Specific (QoS/Security)

The purpose of this section is to study traffic classifications from different QoS and security requirement perspectives. These types of classifications can be applied to different layers, including application, network, and lower layers (MAC and PHY), which makes it a fairly complex task to do. Therefore in this section we try to present different aspects of QoS and security from a traffic classification point of view.

A.2.4.1 QoS Classes of Traffic

QoS is an essential part of a non-best-effort traffic classification, which is important to ensure priority data, in particular; multimedia applications running on stringent wireless links, are handled with proper priority in a timely manner (limited upper delay). These multimedia applications (data containing both audio and video), based on the delay tolerability, can be grouped in the following categories (disregarding the security mandates) [244, 245]:

Streaming - Clients request audio/video files from servers and pipeline reception over the network. Streaming data can be interactive, in which case the user can control some operations (e.g., pause, resume, fast forward, rewind, etc.).

Unidirectional Real-Time (Half-Duplex): Functions similar to TV and radio devices (e.g., mobile-TV), however data delivery direction is from the network to the device. It is a non-interactive service, which involves only listening and/or viewing.

Interactive Real-Time (Full-Duplex): Two-way traffic, similar to a phone conversation and videoconferencing (e.g., talking/listening broadcasting/viewing at the same time). This class has a more stringent delay requirement compared to real-time streaming and unidirectional, which requires normally less than 150 msec of delay for both audio and video applications (in each direction).

A.3 Other Application Layer Schemes

Following any application layer process, appropriate action parameters are created and transported back to the related layer.

A.3.1 Deep Packet Classification (DPC)

One application layer mechanism that has recently gained quite a bit of interest among researchers is the deployment of the Network Based Application Recognition (NBAR) [246, 247] concept. NBAR is capable of recognizing packets with complex fields and attributes combinations. NBAR is able to identify if a packet belongs to certain traffic stream by performing a deep-packet inspection and with an appropriate policy scheme, specific packets can be dealt with accordingly. An entity that works with NBAR is the Protocol Description Language Module (PDLM), which is an application signature. Another entity that NBAR works with is the Cisco Express Forwarding (CEF) that cooperates with NBAR facilitating deep-packet classifications.

A.3.2 QoS API

The QoS API provides a standard interface to add new QoS components before dealing with layered QoS structures [248]. QoS API is able to request bandwidth requirements particular to a specific application at the application layer. This includes facilities to unnecessary latency for streaming audio and video applications. QoS API also deals with the network layer QoS (e.g., RSVP “Resource Reservation Protocol”).

A.3.3 Application Layer Dynamic Services

In the application layer QoS-based approach, one can set the default application and service QoS parameters for various application classification policies (user basis, service class basis, etc.). For application run in a broadband wireless access such as WiMAX, the application type; QoS parameters and the classification rule must be passed to the MAC

layer to initiate dynamic service creation process to send Dynamic Service Addition (DSA) (Request/Response) and Dynamic Service Change (DSC) messages to BS.

When the service flow creation is completed, whether it succeeded or failed, the MAC must send the result to the application layer. Upon receiving the report, the application may generate error message to the user or continue to run the application accordingly.

The application should be able to monitor the QoS status and may send a message to the MAC to initiate DSC-Request message to change the QoS parameters if necessary.

The application should receive a message to initiate Dynamic Service Deletion (DSD) request/response commands to delete the service flow when the application is terminated.

A.3.4 QoS Push and Pull

A non-push, non-pull QoS model is referred to a static scenario in which QoS is set irrespective to service status, which usually contributes to the waste of bandwidth. In such a scenario, when the session is initiated, it will be impossible to change QoS variable while the session is still active. Therefore in order to change QoS requirements, the session has to be torn-down first, which is called pre-provisioned service flow [249].

As a contrast, a non-static QoS system is called dynamic QoS in which QoS can be requested at the initiation of the service and deleted when session terminates, which makes optimal use of bandwidth. In dynamic QoS schemes, it is possible to renegotiate QOS in the middle of an active session. There are two types of dynamic QoS schemes: Push and pull schemes. In the pull model, QoS requirements are pulled by the user/user equipment, which are requested upon service initiation. In the push model, the application client triggers an application service which in turn pushes QoS request from the network, where network server requests QOS upon service initiation. In Push model, the user (UE) requires no knowledge of QoS, which is easy to implement and operate using dynamic QoS and renegotiation scheme. Since the Application Servers are trusted entities and UE/Application clients are authorized by the Application Servers, the QoS

request can be trusted, hence avoiding DoS attacks and theft of service. Many servers already support this model. However it could be relatively slow compared to the pull model [250]. The pull mode is however prone to DoS and theft of service attacks.

A.4 Graphs, Tables, and Charts

Table A.1 Different data rates for different video applications

Algorithm	Format	Format Specific Properties	Data Rate (kbps)
DPCM	H.120	625-line-50 field, 525-line-60 field	1544 (NTSC) 2048 (PAL)
DPCM, DCT, MC	H.261	88x72, 176x144, 352x288, 704x576 Comparable to MPEG-1	20, 2048
8x8 DCT, CIF, SIF	MPEG-1	352x288, 352x240, 25 fps (PAL), CBR, MPEG-1, Audio, Layer 2 VCD	32 (audio) – 1,536 (video)
8x8 DCT VLC	H.262	Similar to MPEG-2	60-2,048
8x8 DCT, CIF, SIF	MPEG-2	MPEG-1, Low (352x288), Main (720x476), SD (1440x1152), HD (1920x1152), SVCD, DVD	32-80,920
OBMC, DCT, SQCIF, QCIF, CIF, 4CIF, 16CIF	H.263	126x96, 176x144, 352x288, 704x576, 1408x1152 – up to 72 fps	10-64 (audio) 24-20,480
4x4 DCT, 8x8 DCT	H.264	Similar to MPEG-4	64-983,040
DCT, VBSMC	MPEG-4	Level 4, 720x1280 progressive 1080x1920 interlace	24-24,5760

Table A.2 VoIP codec bandwidth requirements

Codec	Data Rate (kbps)	Coding Technique	Bandwidth (kbps)
G.711 (A-, μ-law)	64	PCM	68-96
G.722	48, 56, 64	ADPCM	88
G.722.1	24, 32	ACELP	42, 52
G.722.2	23.85	ACELP	42
G.723.1	5.3, 6.4	ACELP/MPC-MLQ	26, 27
G.726	24, 32	ADPCM	48, 64
G.728	16	LD-CELP	78
G.729	6. 4, 8, 11.8	CS-ACELP	31.2
G.729a	8	CS-CELP	40
AMR-WB (G.722.2)	6.6-23.85	ACELP	36-49
AMR-WB+	5.2-48	ACELP	7.2-50
AMR-NB	4.75-12.2	ACELP	36-44
GSM EFR	12.2	ACLEP	30
GSM FR	13.3	RPE-LTP	31
iLBC	13.3, 15.2	FB-LPC	24, 32

Table A.3 Regional IEEE 802.11 *b*/*g* Frequency Channels

Channel	Center Frequency (GHz)	North America	Europe	Spain	France	Japan
1	2.412	X	X			X
2	2.417	X	X			X
3	2.422	X	X			X
4	2.427	X	X			X
5	2.432	X	X			X
6	2.437	X	X			X
7	2.442	X	X			X
8	2.447	X	X			X
9	2.452	X	X			X
10	2.457	X	X	X	X	X
11	2.462	X	X	X	X	X
12	2.467		X		X	X
13	2.472		X		X	X
14	2.484					X

Table A.4 Regional IEEE 802.11 *a* Frequency Channels

Channel	Center Frequency (GHz)	North America	Europe	Japan
34	5.170			X
36	5.180	X	X	
38	5.190			X
40	5.200	X	X	
42	5.210			X
44	5.220	X	X	
46	5.230			X
48	5.240	X	X	
52	5.260	X	X	
56	5.280	X	X	
60	5.300	X	X	
64	5.320	X	X	
100	5.500		X	
104	5.520		X	
108	5.540		X	
112	5.560		X	
116	5.580		X	
120	5.600		X	
124	5.620		X	
128	5.640		X	
132	5.660		X	
136	5.680		X	
140	5.700		X	
149	5.745	X		
153	5.765	X		
157	5.785	X		
161	5.805	X		
165	5.825			

Table A.5 Protocol Hierarchy

Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
⊟ Frame	100.00%	32147	21745794	0.422	0	0	0.000
⊟ Ethernet	99.96%	32133	21744534	0.422	0	0	0.000
⊟ Internet Protocol	87.62%	28166	21492336	0.418	0	0	0.000
⊟ User Datagram Protocol	8.36%	2686	478999	0.009	47	3290	0.000
Domain Name Service	1.00%	323	46514	0.001	323	46514	0.001
⊟ NetBIOS Datagram Service	2.59%	833	202419	0.004	0	0	0.000
⊟ SMB (Server Message Block Protocol)	2.59%	833	202419	0.004	0	0	0.000
⊟ SMB MailSlot Protocol	2.59%	833	202419	0.004	0	0	0.000
Microsoft Windows Browser Protocol	2.56%	824	199732	0.004	824	199732	0.004
Microsoft Windows Logon Protocol (Old)	0.03%	9	2687	0.000	9	2687	0.000
Data	0.70%	225	34065	0.001	225	34065	0.001
NetBIOS Name Service	1.61%	518	48589	0.001	518	48589	0.001
Cisco Hot Standby Router Protocol	1.35%	435	26942	0.001	435	26942	0.001
Session Initiation Protocol	0.00%	1	582	0.000	1	582	0.000
Bootstrap Protocol	0.75%	242	105521	0.002	242	105521	0.002
Routing Information Protocol	0.04%	14	1204	0.000	14	1204	0.000
Connectionless Lightweight Directory Access Protocol	0.13%	42	9387	0.000	42	9387	0.000
⊟ UDP Encapsulation of IPsec Packets	0.01%	4	306	0.000	3	180	0.000
Encapsulating Security Payload	0.00%	1	126	0.000	1	126	0.000
Network Time Protocol	0.01%	2	180	0.000	2	180	0.000
⊟ Transmission Control Protocol	78.81%	25335	20997909	0.408	24503	20366689	0.396
⊟ Hypertext Transfer Protocol	1.59%	511	296653	0.006	374	219475	0.004
Line-based text data	0.21%	66	35873	0.001	66	35873	0.001
Compuserve GIF	0.16%	53	24416	0.000	53	24416	0.000
JPEG File Interchange Format	0.01%	3	3650	0.000	3	3650	0.000
Media Type	0.02%	5	4340	0.000	5	4340	0.000
eXtensible Markup Language	0.02%	7	5053	0.000	7	5053	0.000
Portable Network Graphics	0.01%	3	3846	0.000	3	3846	0.000
TPKT - ISO on TCP - RFC1006	0.02%	5	2060	0.000	5	2060	0.000
Secure Socket Layer	0.97%	313	331534	0.006	313	331534	0.006
Data	0.01%	3	973	0.000	3	973	0.000
Internet Control Message Protocol	0.30%	97	11024	0.000	97	11024	0.000
Open Shortest Path First	0.13%	41	3690	0.000	41	3690	0.000
Generic Routing Encapsulation	0.02%	7	714	0.000	7	714	0.000
Address Resolution Protocol	11.38%	3658	222710	0.004	3658	222710	0.004
⊟ Logical-Link Control	0.76%	245	17132	0.000	16	960	0.000
Spanning Tree Protocol	0.64%	205	12300	0.000	205	12300	0.000
Dynamic Trunking Protocol	0.04%	14	840	0.000	14	840	0.000
Cisco Discovery Protocol	0.02%	7	2828	0.000	7	2828	0.000
Logical-Link Control Basic Format XID	0.00%	1	60	0.000	1	60	0.000
Cisco Wireless LAN Context Control Protocol	0.01%	2	144	0.000	2	144	0.000
⊟ Configuration Test Protocol (loopback)	0.13%	43	2580	0.000	0	0	0.000
Data	0.13%	43	2580	0.000	43	2580	0.000
ATAoverEthernet	0.02%	7	8568	0.000	7	8568	0.000
Cisco Wireless LAN Context Control Protocol	0.00%	1	60	0.000	1	60	0.000
⊟ Internet Protocol Version 6	0.02%	7	686	0.000	0	0	0.000
Internet Control Message Protocol v6	0.01%	4	344	0.000	4	344	0.000
⊟ User Datagram Protocol	0.01%	3	342	0.000	0	0	0.000
DHCPv6	0.01%	3	342	0.000	3	342	0.000
Data	0.02%	6	462	0.000	6	462	0.000
⊟ Cisco ISL	0.04%	14	1260	0.000	0	0	0.000
⊟ Ethernet	0.04%	14	1260	0.000	0	0	0.000
⊟ Logical-Link Control	0.04%	14	1260	0.000	0	0	0.000
Dynamic Trunking Protocol	0.04%	14	1260	0.000	14	1260	0.000

Table A.6 Ethernet flows ordered by heavy hitter end point bytes

Ethernet Conversations

Address A	Address B	Packets	Bytes *	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
xxxx	yyyy	15340	20079465	15340	20079465	0	0	2.381945000	404.9778	396653.12	N/A
xxxx	yyyy	10272	948450	0	0	10272	948450	1.801738000	405.7396	N/A	18700.66
xxxx	yyyy	2262	135935	2262	135935	0	0	0.116182000	411.4847	2642.82	N/A
xxxx	yyyy	80	47200	80	47200	0	0	32.577021000	376.9135	1001.82	N/A
xxxx	yyyy	172	29274	82	6541	90	22733	0.000000000	358.1720	146.10	507.76
xxxx	yyyy	209	21359	209	21359	0	0	53.326330000	358.1894	477.04	N/A
xxxx	yyyy	81	19598	81	19598	0	0	13.074541000	398.3006	393.63	N/A
xxxx	yyyy	86	18170	86	18170	0	0	13.074579000	398.2852	364.96	N/A
xxxx	yyyy	257	17350	257	17350	0	0	0.011066000	409.9104	338.61	N/A
xxxx	yyyy	224	13860	224	13860	0	0	1.592621000	408.4631	271.46	N/A
xxxx	yyyy	211	13082	211	13082	0	0	1.867859000	409.6521	255.48	N/A
xxxx	yyyy	205	12300	205	12300	0	0	0.103039000	411.0511	239.39	N/A
xxxx	yyyy	62	10332	62	10332	0	0	10.819365000	394.9700	209.27	N/A
xxxx	yyyy	58	10306	58	10306	0	0	7.070626000	401.4037	205.40	N/A
xxxx	yyyy	8	8660	8	8660	0	0	17.096301000	359.9898	192.45	N/A
xxxx	yyyy	114	7475	114	7475	0	0	3.449150000	309.2908	193.35	N/A
xxxx	yyyy	47	7427	47	7427	0	0	15.766646000	384.0010	154.73	N/A
xxxx	yyyy	43	7155	43	7155	0	0	2.139889000	399.6688	143.22	N/A
xxxx	yyyy	33	6587	33	6587	0	0	10.439151000	399.8668	131.78	N/A
xxxx	yyyy	35	6582	35	6582	0	0	43.387229000	342.7116	153.65	N/A
xxxx	yyyy	29	6530	29	6530	0	0	10.836523000	400.2669	130.51	N/A
xxxx	yyyy	29	6530	29	6530	0	0	4.457948000	402.3805	129.83	N/A
xxxx	yyyy	28	6470	28	6470	0	0	11.701854000	400.0232	129.39	N/A
xxxx	yyyy	28	6470	28	6470	0	0	1.811931000	399.9830	129.41	N/A
xxxx	yyyy	27	6410	27	6410	0	0	7.543851000	400.0421	128.19	N/A
xxxx	yyyy	27	6410	27	6410	0	0	8.420542000	400.0389	128.19	N/A
xxxx	yyyy	27	6410	27	6410	0	0	4.265939000	400.2831	128.11	N/A
xxxx	yyyy	27	6410	27	6410	0	0	8.083161000	400.2283	128.13	N/A
xxxx	yyyy	27	6410	27	6410	0	0	9.768650000	400.2297	128.13	N/A
xxxx	yyyy	27	6410	27	6410	0	0	11.563555000	400.2656	128.11	N/A
xxxx	yyyy	27	6410	27	6410	0	0	8.945987000	400.2367	128.12	N/A
xxxx	yyyy	26	6318	26	6318	0	0	4.477096000	400.3840	126.24	N/A
xxxx	yyyy	26	6318	26	6318	0	0	5.872089000	400.0084	126.36	N/A
xxxx	yyyy	26	6318	26	6318	0	0	8.202490000	400.2736	126.27	N/A
xxxx	yyyy	26	6318	26	6318	0	0	9.814400000	400.2656	126.28	N/A
xxxx	yyyy	26	6318	26	6318	0	0	10.294794000	400.0389	126.35	N/A
xxxx	yyyy	26	6318	26	6318	0	0	1.542215000	400.0226	126.35	N/A
xxxx	yyyy	27	6227	27	6227	0	0	4.809094000	399.6452	124.65	N/A
xxxx	yyyy	26	6167	26	6167	0	0	13.719238000	384.3963	128.35	N/A
xxxx	yyyy	25	6075	25	6075	0	0	0.000282000	399.6341	121.61	N/A
xxxx	yyyy	25	6075	25	6075	0	0	1.986426000	399.7799	121.57	N/A
xxxx	yyyy	87	6014	85	5930	2	84	2.106739000	405.2319	117.07	1.66
xxxx	yyyy	25	5924	25	5924	0	0	13.834749000	383.4658	123.59	N/A
xxxx	yyyy	36	5922	36	5922	0	0	17.396969000	348.0261	136.13	N/A
xxxx	yyyy	36	5278	36	5278	0	0	93.307040000	277.9280	151.92	N/A
xxxx	yyyy	35	4928	35	4928	0	0	12.172357000	399.0056	98.81	N/A
xxxx	yyyy	28	4500	28	4500	0	0	10.558705000	343.3400	104.85	N/A
xxxx	yyyy	47	4155	47	4155	0	0	13.504966000	378.1107	87.91	N/A
xxxx	yyyy	23	3750	23	3750	0	0	0.120452000	364.7750	82.24	N/A
xxxx	yyyy	41	3690	41	3690	0	0	5.138473000	400.0161	73.80	N/A
xxxx	yyyy	19	3396	19	3396	0	0	29.060473000	364.3001	74.58	N/A
xxxx	yyyy	37	3354	37	3354	0	0	24.625805000	320.9413	83.60	N/A
xxxx	yyyy	16	3002	16	3002	0	0	12.221941000	390.9727	61.43	N/A
xxxx	yyyy	42	2867	42	2867	0	0	14.386653000	391.3686	58.60	N/A
xxxx	yyyy	28	2865	28	2865	0	0	43.521578000	364.0251	62.96	N/A
xxxx	yyyy	2	2828	2	2828	0	0	21.251742000	0.0001	209481481.48	N/A
xxxx	yyyy	41	2803	41	2803	0	0	15.891203000	391.2518	57.31	N/A
xxxx	yyyy	42	2688	42	2688	0	0	2.675875000	400.1701	53.74	N/A
xxxx	yyyy	13	2472	13	2472	0	0	154.424451000	121.8761	162.26	N/A
xxxx	yyyy	41	2460	41	2460	0	0	7.895929000	400.2255	49.17	N/A
xxxx	yyyy	26	2431	26	2431	0	0	0.999890000	386.2026	50.36	N/A
xxxx	yyyy	21	2210	21	2210	0	0	83.998103000	312.7562	56.53	N/A
xxxx	yyyy	15	1967	15	1967	0	0	55.778928000	299.7265	52.50	N/A

Table A.7 Ethernet bidirectional flows ordered by heavy hitter end point bytes (based on Table A.2)

Ethernet Hosts

Address	Packets	Bytes ˆ	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
Address	25908	21066557	10393	958429	15515	20108128
Address	15359	20086326	15359	20086326	0	0
Address	10483	961532	211	13082	10272	948450
Address	5041	574747	0	0	5041	574747
Address	2262	135935	2262	135935	0	0
Address	80	47200	80	47200	0	0
Address	429	46624	347	40083	82	6541
Address	435	26942	0	0	435	26942
Address	336	23408	295	20948	41	2460
Address	209	21359	209	21359	0	0
Address	120	20638	0	0	120	20638
Address	319	19951	317	19867	2	84
Address	81	19598	81	19598	0	0
Address	86	18170	86	18170	0	0
Address	95	16471	95	16471	0	0
Address	86	14174	86	14174	0	0
Address	224	13860	224	13860	0	0
Address	205	12300	0	0	205	12300
Address	59	9672	0	0	59	9672
Address	8	8660	8	8660	0	0
Address	114	7475	114	7475	0	0
Address	47	7427	47	7427	0	0
Address	43	7155	43	7155	0	0
Address	33	6587	33	6587	0	0
Address	35	6582	35	6582	0	0
Address	29	6530	29	6530	0	0
Address	29	6530	29	6530	0	0
Address	28	6470	28	6470	0	0
Address	28	6470	28	6470	0	0
Address	27	6410	27	6410	0	0
Address	27	6410	27	6410	0	0
Address	27	6410	27	6410	0	0
Address	27	6410	27	6410	0	0
Address	27	6410	27	6410	0	0
Address	27	6410	27	6410	0	0
Address	27	6410	27	6410	0	0
Address	26	6318	26	6318	0	0
Address	26	6318	26	6318	0	0
Address	26	6318	26	6318	0	0
Address	26	6318	26	6318	0	0
Address	26	6318	26	6318	0	0
Address	26	6318	26	6318	0	0
Address	27	6227	27	6227	0	0
Address	26	6167	26	6167	0	0
Address	25	6075	25	6075	0	0
Address	25	6075	25	6075	0	0
Address	25	5924	25	5924	0	0
Address	36	5278	36	5278	0	0
Address	35	4928	0	0	35	4928
Address	28	4500	28	4500	0	0
Address	48	4232	48	4232	0	0
Address	41	3690	41	3690	0	0
Address	41	3690	0	0	41	3690
Address	19	3396	19	3396	0	0
Address	32	3209	32	3209	0	0
Address	16	3002	16	3002	0	0
Address	4	2948	0	0	4	2948
Address	42	2867	42	2867	0	0

Table A.8 IPv4 flows ordered by heavy hitter end point bytes

IPv4 Conversations

Address A	Address B	Packets ˆ	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
xxxx	yyyy	14978	13576738	5690	312084	9288	13264654	151.857165000	78.5998	31764.35	1350095.03
xxxx	yyyy	5452	4981695	2033	114741	3419	4866954	97.821744000	52.8085	17382.19	737298.26
xxxx	yyyy	384	244225	142	14700	242	229525	3.786964000	403.7544	291.27	4547.81
xxxx	yyyy	364	74148	217	34285	147	39863	72.234699000	173.3670	1582.08	1839.47
xxxx	yyyy	352	297661	133	11161	219	286500	70.682240000	86.1442	1036.49	26606.55
xxxx	yyyy	343	266270	137	21051	206	245219	97.714167000	157.6042	1068.55	12447.34
xxxx	yyyy	323	89897	169	27170	154	62727	176.160782000	78.8098	2758.03	6367.43
xxxx	yyyy	231	66901	126	51844	105	15057	71.684330000	225.1206	1842.35	535.07
xxxx	yyyy	224	13860	224	13860	0	0	1.592621000	408.4631	271.46	N/A
xxxx	yyyy	212	92378	105	52006	107	40372	277.289360000	64.5930	6441.07	5000.17
xxxx	yyyy	211	13082	211	13082	0	0	1.867859000	409.6521	255.48	N/A
xxxx	yyyy	209	21359	209	21359	0	0	53.326330000	358.1894	477.04	N/A
xxxx	yyyy	189	163250	73	5906	116	157344	123.655123000	131.6629	358.86	9560.41
xxxx	yyyy	172	29274	90	22733	82	6541	0.000000000	358.1720	507.76	146.10
xxxx	yyyy	157	130811	61	11257	96	119554	167.698619000	87.6200	1027.80	10915.68
xxxx	yyyy	152	17247	74	11243	78	6004	1.801738000	351.9408	255.57	136.48
xxxx	yyyy	140	68234	140	68234	0	0	13.073987000	398.3008	1370.50	N/A
xxxx	yyyy	123	51617	62	15434	61	36183	213.084099000	194.2759	635.55	1489.96
xxxx	yyyy	121	34499	66	23313	55	11186	170.964937000	72.7614	2563.23	1229.88
xxxx	yyyy	120	38938	59	13785	61	25153	90.437917000	7.3834	14936.26	27253.67
xxxx	yyyy	108	26385	57	12259	51	14126	271.934199000	24.3769	4023.16	4635.87
xxxx	yyyy	107	54917	52	14817	55	40100	275.976091000	22.0299	5380.68	14562.01
xxxx	yyyy	104	78132	41	5186	63	72946	97.888721000	63.9374	648.89	9127.18
xxxx	yyyy	97	71158	39	7915	58	63243	332.161318000	7.0007	9044.83	72270.65
xxxx	yyyy	92	72425	37	5045	55	67380	94.368076000	0.3796	106335.61	1420196.92
xxxx	yyyy	86	61593	38	6119	48	55474	64.721363000	271.1217	180.55	1636.87
xxxx	yyyy	86	49995	38	10697	48	39298	353.743737000	12.5070	6842.26	25136.69
xxxx	yyyy	83	5810	83	5810	0	0	2.106739000	405.2319	114.70	N/A
xxxx	yyyy	80	25251	44	12418	36	12833	169.088048000	76.3507	1301.15	1344.64
xxxx	yyyy	79	22181	45	9221	34	12960	273.425116000	57.7239	1277.95	1796.14
xxxx	yyyy	76	53423	30	4280	46	49143	340.589088000	65.6618	521.46	5987.41
xxxx	yyyy	75	43817	35	7068	40	36749	69.202014000	97.4208	580.41	3017.75
xxxx	yyyy	75	14416	43	8738	32	5678	292.124201000	60.7217	1151.22	748.07
xxxx	yyyy	64	14913	35	11007	29	3906	2.317305000	94.5296	931.52	330.56
xxxx	yyyy	58	13147	32	9135	26	4012	66.090441000	265.8444	274.90	120.73
xxxx	yyyy	54	36031	23	2485	31	33546	296.458490000	99.7920	199.21	2689.27
xxxx	yyyy	54	3204	0	0	54	3204	139.741283000	228.5816	N/A	112.14
xxxx	yyyy	49	15534	24	5854	25	9680	275.410645000	20.1225	2327.35	3848.43
xxxx	yyyy	47	17343	47	17343	0	0	13.074541000	398.3006	348.34	N/A
xxxx	yyyy	47	3290	47	3290	0	0	280.502317000	24.5830	1070.66	N/A
xxxx	yyyy	45	29156	19	2932	26	26224	349.526051000	16.7250	1402.45	12543.65
xxxx	yyyy	44	13686	24	10818	20	2868	334.160841000	7.1016	12186.55	3230.82
xxxx	yyyy	43	36042	16	1946	27	34096	297.743438000	64.1394	242.72	4252.73
xxxx	yyyy	42	15498	42	15498	0	0	17.387569000	374.9915	330.63	N/A
xxxx	yyyy	41	3690	41	3690	0	0	5.138473000	400.0161	73.80	N/A
xxxx	yyyy	38	9439	22	3507	16	5932	62.416601000	303.8346	92.34	156.19
xxxx	yyyy	37	3555	37	3555	0	0	13.504966000	378.1107	75.22	N/A
xxxx	yyyy	36	5922	36	5922	0	0	17.396969000	348.0261	136.13	N/A
xxxx	yyyy	36	3312	36	3312	0	0	24.625805000	320.9413	82.56	N/A
xxxx	yyyy	33	12399	17	6930	16	5469	279.008618000	16.7104	3317.70	2618.25
xxxx	yyyy	32	5314	32	5314	0	0	10.819365000	394.9700	107.63	N/A
xxxx	yyyy	32	5022	32	5022	0	0	101.252798000	269.9822	148.81	N/A
xxxx	yyyy	32	8119	16	3262	16	4857	122.308613000	0.2266	115170.40	171484.56
xxxx	yyyy	31	6964	11	3147	20	3817	69.294935000	45.2347	556.56	675.06
xxxx	yyyy	31	3001	16	1723	15	1278	270.664840000	30.4613	452.51	335.64
xxxx	yyyy	30	5018	30	5018	0	0	32.914875000	370.4624	108.36	N/A
xxxx	yyyy	30	5286	30	5286	0	0	7.070626000	401.4037	105.35	N/A
xxxx	yyyy	30	6282	30	6282	0	0	55.721919000	330.3769	152.12	N/A
xxxx	yyyy	30	7390	15	4301	15	3089	96.670488000	55.2008	623.32	447.67
xxxx	yyyy	28	5020	28	5020	0	0	61.595281000	342.7364	117.17	N/A
xxxx	yyyy	27	6410	27	6410	0	0	7.543851000	400.0421	128.19	N/A
xxxx	yyyy	27	6410	27	6410	0	0	8.420542000	400.0389	128.19	N/A
xxxx	yyyy	27	6410	27	6410	0	0	4.265939000	400.2831	128.11	N/A
xxxx	yyyy	27	6410	27	6410	0	0	8.083161000	400.2283	128.13	N/A
xxxx	yyyy	27	6410	27	6410	0	0	9.768650000	400.2297	128.13	N/A
xxxx	yyyy	27	6410	27	6410	0	0	11.563555000	400.2656	128.11	N/A
xxxx	yyyy	27	6410	27	6410	0	0	8.945987000	400.2367	128.12	N/A
xxxx	yyyy	27	6410	27	6410	0	0	4.457948000	402.3805	127.44	N/A
xxxx	yyyy	27	6410	27	6410	0	0	10.836523000	400.2669	128.11	N/A

Table A.9 TCP flows ordered by heavy hitter end point bytes

TCP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes ˆ	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
xxxx	yyyy	xxxx	yyyy	14975	13576552	5687	311898	9288	13264654	172.935077000	57.5219	43377.97	1844814.06
xxxx	yyyy	xxxx	yyyy	5449	4981509	2030	114555	3419	4866954	118.699987000	31.9303	28701.28	1219395.26
xxxx	yyyy	xxxx	yyyy	374	241991	136	13005	238	228986	4.976367000	402.5650	258.44	4550.54
xxxx	yyyy	xxxx	yyyy	190	168550	70	5083	120	163467	91.870863000	64.9556	626.03	20132.78
xxxx	yyyy	xxxx	yyyy	147	128965	56	4550	91	124415	193.914975000	61.4031	592.80	16209.61
xxxx	yyyy	xxxx	yyyy	144	124230	52	4537	92	119693	70.682240000	86.1441	421.34	11115.61
xxxx	yyyy	xxxx	yyyy	141	110039	56	9150	85	100889	97.714167000	64.1120	1141.75	12589.10
xxxx	yyyy	xxxx	yyyy	144	108574	59	10046	85	98528	97.714180000	64.1120	1253.56	12294.48
xxxx	yyyy	xxxx	yyyy	98	81126	38	7365	60	73761	167.698619000	62.3375	945.18	9466.03
xxxx	yyyy	xxxx	yyyy	63	56627	23	2257	40	54370	94.534794000	0.2128	84835.67	2043648.84
xxxx	yyyy	xxxx	yyyy	59	49685	23	3892	36	45793	236.924035000	18.3946	1692.67	19915.88
xxxx	yyyy	xxxx	yyyy	58	47657	22	1855	36	45802	184.930031000	70.3883	210.83	5205.64
xxxx	yyyy	xxxx	yyyy	52	45001	20	2177	32	42824	341.899464000	64.3514	270.64	5323.77
xxxx	yyyy	xxxx	yyyy	59	44201	23	2936	36	41265	97.888721000	63.9374	367.36	5163.18
xxxx	yyyy	xxxx	yyyy	45	37443	18	2043	27	35400	69.202014000	0.2218	73679.70	1276682.07
xxxx	yyyy	xxxx	yyyy	43	36042	16	1946	27	34096	297.743438000	64.1394	242.72	4252.73
xxxx	yyyy	xxxx	yyyy	42	34285	17	1356	25	32929	123.655123000	62.6054	173.28	4207.81
xxxx	yyyy	xxxx	yyyy	45	33931	18	2250	27	31681	97.888750000	63.9373	281.53	3964.01
xxxx	yyyy	xxxx	yyyy	52	32184	22	6214	30	25970	353.743737000	12.5068	3974.78	16611.71
xxxx	yyyy	xxxx	yyyy	39	31091	15	3083	24	28008	338.578703000	0.5833	42283.63	384132.32
xxxx	yyyy	xxxx	yyyy	39	29911	17	2512	22	27399	329.247670000	0.3817	52646.33	574226.46
xxxx	yyyy	xxxx	yyyy	38	29494	16	2178	22	27316	64.721363000	0.2932	59434.92	745419.75
xxxx	yyyy	xxxx	yyyy	45	29156	19	2932	26	26224	349.526051000	16.7250	1402.45	12543.65
xxxx	yyyy	xxxx	yyyy	128	28901	64	5168	64	23733	214.697490000	22.2290	1859.91	8541.26
xxxx	yyyy	xxxx	yyyy	35	27934	14	2827	21	25107	295.017687000	0.3574	63270.56	561915.13
xxxx	yyyy	xxxx	yyyy	32	22558	13	2594	19	19964	335.587790000	0.4276	48534.74	373534.16
xxxx	yyyy	xxxx	yyyy	35	20210	16	3866	19	16344	213.084099000	39.4189	784.60	3316.99
xxxx	yyyy	xxxx	yyyy	27	17997	12	2226	15	15771	342.389444000	15.0018	1187.06	8410.21
xxxx	yyyy	xxxx	yyyy	34	17811	16	4483	18	13328	355.594118000	10.6566	3365.43	10005.45
xxxx	yyyy	xxxx	yyyy	26	17509	11	2238	15	15271	332.161318000	0.6379	28067.40	191518.01
xxxx	yyyy	xxxx	yyyy	22	16106	9	892	13	15214	296.459181000	0.1765	40437.47	689703.63
xxxx	yyyy	xxxx	yyyy	40	14214	20	2828	20	11386	176.160782000	6.0781	3722.21	14986.23
xxxx	yyyy	xxxx	yyyy	20	14047	8	838	12	13209	296.458490000	0.0249	268966.90	4239598.80
xxxx	yyyy	xxxx	yyyy	20	13770	9	1502	11	12268	94.422307000	0.1344	89437.37	730504.42
xxxx	yyyy	xxxx	yyyy	19	10935	10	6792	9	4143	296.462272000	0.7906	68726.77	41922.11
xxxx	yyyy	xxxx	yyyy	20	9555	9	3164	11	6391	296.480988000	0.4594	55102.15	111301.47
xxxx	yyyy	xxxx	yyyy	34	9470	18	7010	16	2460	2.317876000	42.1383	1330.86	467.03
xxxx	yyyy	xxxx	yyyy	16	9365	8	2026	8	7339	273.791487000	0.3172	51091.46	185074.16
xxxx	yyyy	xxxx	yyyy	17	9221	8	2023	9	7198	275.410645000	0.6333	25556.72	90932.92
xxxx	yyyy	xxxx	yyyy	17	8968	8	1670	9	7298	95.604717000	0.1929	69271.97	302722.66
xxxx	yyyy	xxxx	yyyy	17	8830	8	3937	9	4893	295.051322000	0.3643	86447.44	107438.99
xxxx	yyyy	xxxx	yyyy	15	8575	7	1055	8	7520	203.980446000	0.2527	33393.47	238027.41
xxxx	yyyy	xxxx	yyyy	15	8571	7	1055	8	7516	183.376380000	0.3000	28134.08	200432.01
xxxx	yyyy	xxxx	yyyy	15	8568	7	1058	8	7510	193.411856000	0.1414	59839.51	424758.74
xxxx	yyyy	xxxx	yyyy	30	8092	16	2564	14	5528	251.382074000	3.5885	5715.97	12323.66
xxxx	yyyy	xxxx	yyyy	16	7926	8	1979	8	5947	341.777088000	0.1309	120920.50	363372.51
xxxx	yyyy	xxxx	yyyy	15	7904	7	1179	8	6725	296.469743000	0.4109	22952.59	130921.27
xxxx	yyyy	xxxx	yyyy	15	7876	7	1300	8	6576	91.659894000	0.1418	73365.50	371116.57
xxxx	yyyy	xxxx	yyyy	16	7607	8	4315	8	3292	339.079871000	0.5568	62001.36	47302.08
xxxx	yyyy	xxxx	yyyy	16	7599	8	4318	8	3281	339.176849000	0.4655	74203.60	56383.05
xxxx	yyyy	xxxx	yyyy	16	7347	8	3938	8	3409	340.968094000	0.9142	34459.19	29830.21
xxxx	yyyy	xxxx	yyyy	16	7065	8	3952	8	3113	340.385548000	0.4032	78412.50	61765.72
xxxx	yyyy	xxxx	yyyy	16	7027	8	3937	8	3090	340.930136000	0.3761	83749.70	65731.92
xxxx	yyyy	xxxx	yyyy	20	6963	10	3859	10	3104	279.008618000	7.5651	4080.84	3282.44
xxxx	yyyy	xxxx	yyyy	16	6860	8	3913	8	2947	332.936489000	1.0874	28787.54	21680.78
xxxx	yyyy	xxxx	yyyy	16	6820	8	3952	8	2868	335.948481000	0.4801	65847.19	47785.86
xxxx	yyyy	xxxx	yyyy	16	6805	8	3936	8	2869	293.770161000	0.3325	94687.94	69019.23
xxxx	yyyy	xxxx	yyyy	16	6804	8	3936	8	2868	277.289375000	0.3428	91850.49	66927.64
xxxx	yyyy	xxxx	yyyy	16	6742	8	3912	8	2830	332.907731000	1.0988	28481.62	20604.03
xxxx	yyyy	xxxx	yyyy	26	6606	14	2401	12	4205	238.971263000	4.6793	4104.87	7189.08
xxxx	yyyy	xxxx	yyyy	16	6343	8	3949	8	2394	335.924178000	0.5095	62009.66	37592.08
xxxx	yyyy	xxxx	yyyy	16	6343	8	3949	8	2394	340.376755000	0.4309	73320.06	44448.78
xxxx	yyyy	xxxx	yyyy	12	5878	6	755	6	5123	333.912842000	62.3377	96.89	657.45
xxxx	yyyy	xxxx	yyyy	14	5806	7	772	7	5034	352.739301000	13.5115	457.09	2980.57
xxxx	yyyy	xxxx	yyyy	13	5772	7	2713	6	3059	123.968400000	62.2920	348.42	392.86
xxxx	yyyy	xxxx	yyyy	12	5535	6	1456	6	4079	276.539183000	0.1290	90280.58	252922.03
xxxx	yyyy	xxxx	yyyy	16	5522	9	1721	7	3801	348.505531000	17.7457	775.85	1713.54
xxxx	yyyy	xxxx	yyyy	12	5519	6	1440	6	4079	294.793312000	0.0908	126875.04	359391.18
xxxx	yyyy	xxxx	yyyy	13	5436	7	3071	6	2365	295.286585000	0.4324	56815.92	43754.37

Table A.10 UDP flows ordered by heavy hitter end point bytes

UDP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
xxxx	yyyy	xxxx	yyyy	224	13860	224	13860	0	0	1.592621000	408.4631	271.46	N/A
xxxx	yyyy	xxxx	yyyy	211	13082	211	13082	0	0	1.867859000	409.6521	255.48	N/A
xxxx	yyyy	xxxx	yyyy	196	18320	196	18320	0	0	53.326330000	358.1894	409.17	N/A
xxxx	yyyy	xxxx	yyyy	140	68234	140	68234	0	0	13.073987000	398.3008	1370.50	N/A
xxxx	yyyy	xxxx	yyyy	47	17343	0	0	47	17343	13.074541000	398.3006	N/A	348.34
xxxx	yyyy	xxxx	yyyy	42	15498	0	0	42	15498	17.387569000	374.9915	N/A	330.63
xxxx	yyyy	xxxx	yyyy	36	3312	36	3312	0	0	13.504966000	378.1107	70.07	N/A
xxxx	yyyy	xxxx	yyyy	36	2088	0	0	36	2088	161.753368000	206.5695	N/A	80.86
xxxx	yyyy	xxxx	yyyy	36	5922	0	0	36	5922	17.396969000	348.0261	N/A	136.13
xxxx	yyyy	xxxx	yyyy	36	3312	36	3312	0	0	24.625805000	320.9413	82.56	N/A
xxxx	yyyy	xxxx	yyyy	32	5314	0	0	32	5314	10.819365000	394.9700	N/A	107.63
xxxx	yyyy	xxxx	yyyy	30	5018	0	0	30	5018	32.914875000	370.4624	N/A	108.36
xxxx	yyyy	xxxx	yyyy	30	5286	0	0	30	5286	7.070626000	401.4037	N/A	105.35
xxxx	yyyy	xxxx	yyyy	28	5020	0	0	28	5020	61.595281000	342.7364	N/A	117.17
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	4.265939000	400.2831	126.27	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	4.477096000	400.3840	126.24	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	5.872089000	400.0084	126.36	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	6.801711000	400.0367	126.35	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	7.543851000	400.0421	126.35	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	8.083161000	400.2283	126.29	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	8.202490000	400.2736	126.27	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	8.420542000	400.0389	126.35	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	8.945987000	400.2367	126.29	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	9.768650000	400.2297	126.29	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	9.814400000	400.2656	126.28	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	10.294794000	400.0389	126.35	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	10.836523000	400.2669	126.28	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	11.563555000	400.2656	126.28	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	11.701854000	400.0232	126.35	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	1.542215000	400.0226	126.35	N/A
xxxx	yyyy	xxxx	yyyy	26	6318	26	6318	0	0	1.811931000	399.9830	126.37	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	4.809094000	399.6452	121.61	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	10.439151000	399.8668	121.54	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	0.000282000	399.6341	121.61	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	13.719238000	384.3963	126.43	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	15.766646000	384.0010	126.56	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	1.986426000	399.7799	121.57	N/A
xxxx	yyyy	xxxx	yyyy	25	6075	25	6075	0	0	2.139889000	399.6688	121.60	N/A
xxxx	yyyy	xxxx	yyyy	24	5832	24	5832	0	0	13.834749000	383.4658	121.67	N/A
xxxx	yyyy	xxxx	yyyy	24	5730	24	5730	0	0	61.586028000	324.5128	141.26	N/A
xxxx	yyyy	xxxx	yyyy	23	3750	0	0	23	3750	0.120452000	364.7750	N/A	82.24
xxxx	yyyy	xxxx	yyyy	21	1932	21	1932	0	0	1.002784000	16.5304	935.00	N/A
xxxx	yyyy	xxxx	yyyy	18	1656	18	1656	0	0	83.998103000	312.7562	42.36	N/A
xxxx	yyyy	xxxx	yyyy	18	1260	18	1260	0	0	295.675066000	9.4102	1071.18	N/A
xxxx	yyyy	xxxx	yyyy	18	1656	18	1656	0	0	318.251467000	34.4897	384.11	N/A
xxxx	yyyy	xxxx	yyyy	14	3366	14	3366	0	0	101.252798000	269.9822	99.74	N/A
xxxx	yyyy	xxxx	yyyy	13	3039	13	3039	0	0	61.558268000	319.0550	76.20	N/A
xxxx	yyyy	xxxx	yyyy	12	1104	12	1104	0	0	167.957557000	128.3790	68.80	N/A
xxxx	yyyy	xxxx	yyyy	12	1104	12	1104	0	0	105.648773000	188.2393	46.92	N/A
xxxx	yyyy	xxxx	yyyy	12	840	12	840	0	0	280.602851000	4.6055	1459.11	N/A
xxxx	yyyy	xxxx	yyyy	10	920	10	920	0	0	65.309028000	259.3985	28.37	N/A
xxxx	yyyy	xxxx	yyyy	10	700	10	700	0	0	295.573730000	9.5113	588.77	N/A
xxxx	yyyy	xxxx	yyyy	10	920	10	920	0	0	342.735936000	12.7680	576.44	N/A
xxxx	yyyy	xxxx	yyyy	8	736	8	736	0	0	42.541685000	360.6525	16.33	N/A
xxxx	yyyy	xxxx	yyyy	8	2266	8	2266	0	0	12.221941000	390.9727	46.37	N/A
xxxx	yyyy	xxxx	yyyy	8	1800	8	1800	0	0	101.251772000	223.0913	64.55	N/A
xxxx	yyyy	xxxx	yyyy	7	1085	7	1085	0	0	42.489235000	359.8852	24.12	N/A
xxxx	yyyy	xxxx	yyyy	7	644	7	644	0	0	46.399822000	324.6865	15.87	N/A
xxxx	yyyy	xxxx	yyyy	7	490	7	490	0	0	280.502317000	4.7058	833.01	N/A
xxxx	yyyy	xxxx	yyyy	6	552	6	552	0	0	55.721919000	3.7547	1176.12	N/A
xxxx	yyyy	xxxx	yyyy	6	552	6	552	0	0	212.730405000	3.7707	1171.14	N/A
xxxx	yyyy	xxxx	yyyy	6	552	6	552	0	0	219.893525000	4.9298	895.78	N/A
xxxx	yyyy	xxxx	yyyy	5	1222	5	1222	0	0	18.958437000	157.2900	62.15	N/A
xxxx	yyyy	xxxx	yyyy	5	1165	5	1165	0	0	342.735775000	12.7731	729.66	N/A
xxxx	yyyy	xxxx	yyyy	4	344	4	344	0	0	314.834780000	86.9902	31.64	N/A
xxxx	yyyy	xxxx	yyyy	4	368	4	368	0	0	113.733295000	271.7616	10.83	N/A
xxxx	yyyy	xxxx	yyyy	4	368	4	368	0	0	141.377150000	151.8045	19.39	N/A
xxxx	yyyy	xxxx	yyyy	4	344	4	344	0	0	38.878838000	85.9882	32.00	N/A
xxxx	yyyy	xxxx	yyyy	3	709	3	709	0	0	26.939436000	355.1166	15.97	N/A

Table A.11 Wireless Network Component Statistics

Network	Total Packets	100.000%
Network	Total Broadcast	33.059%
Network	Total Multicast	1.922%
Network	Average Utilization (percent)	0.279
Network	Average Utilization (bits/s)	300,868.339
Network	Current Utilization (percent)	0.204
Network	Current Utilization (bits/s)	220,352.000
Network	Max Utilization (percent)	0.435
Network	Max Utilization (bits/s)	469,968.000
Errors	Total	2.510%
Errors	CRC	2.510%
Counts	Physical Addresses	356
Counts	IP Addresses	6
Counts	IPv6 Addresses	0
Counts	AppleTalk Addresses	0
Counts	DECnet Addresses	0
Counts	IPX Addresses	0
Counts	Protocols	36
Size Distribution	< 64	57.947%
Size Distribution	64-127	13.713%
Size Distribution	128-255	16.414%
Size Distribution	256-511	11.926%
Size Distribution	512-1023	0.000%
Size Distribution	1024-2047	0.000%
Size Distribution	2048-2346	0.000%
Size Distribution	>= 2347	0.000%
Wireless	Wireless Networks	11
Wireless	Weak Wireless Networks	28
Wireless	Ad Hoc Networks	0
Wireless	Access Points	13
Wireless	Clients	50
Wireless	Trusted Access Points	0
Wireless	Known Access Points	0
Wireless	Unknown Access Points	13
Broadcast	Storm	126
Severe	Broadcast Storm	67

Wireless - Too Many Physical Errors		114
Wireless Access Point - Broadcasting ESSID		11
Wireless Access Point - Missing		1
Wireless Access Point - Mixed Mode		2
Wireless Access Point - Physical Errors		111
Wireless Access Point - Too Many Retries		78
Wireless Access Point - Weak Signal		27
Wireless Authentication Denied		1
Wireless Channel Overlap		8
Wireless Client - No Response to Probe Request		13
Wireless Client - Physical Errors		9
Wireless Client - Probe Response Not Accepted		118
Wireless Client - Too Many Retries		3
Wireless Client - Weak Signal		5
Wireless Deauthentication Attack		17
Wireless Duration Attack		15
Wireless Excessive Probe Requests		18
Wireless Excessive RTS		5
Wireless Low Signal-to-Noise Ratio		19
Wireless Reassociation Denied		0
Wireless RF Interference		0
Wireless RF Jamming		0
Wireless Security Error		0
802.11 Analysis	Average Signal Strength	16.453
802.11 Analysis	Average Signal dBm	-83.358
802.11 Analysis	Average Noise	6.261
802.11 Analysis	Average Noise dBm	-93.739
802.11 Analysis	802.11 Data	45.29%
802.11 Analysis	802.11 Management	48.57%
802.11 Analysis	802.11 Control	3.63%
802.11 Analysis	Retry	53.78%
IP Analysis	ARPs Unanswered	825

From protocol point of view, the following protocols were involved in the wireless transmission:

Table A.12 Packets details

Protocol	Bytes	Packets
802.11 Null Data	879792	25901
Beacon	3781251	18506
Deauthentication	209400	6980
Probe Responses	958448	5173
Acknowledgment	21448	1532
Probe Requests	60297	1112
SSDP	311110	876
Request	57750	825
LSAP-01	27000	663
Clear to Send	8596	614
Name Svc	64950	542
Cisco Discovery	138355	371
802.11 QoS Null Data	9150	305
Request to Send	4520	226
802.11 TKIP Data	15703	110
802.11 WEP Data	7228	64
Authentication	756	22
Association Requests	1727	12
Association Responses	621	6
Contention-Free End	100	5
802.1x	736	4
Reassociation Responses	254	2
Reassociation Requests	268	2
EAP Response	48	1
UDP	0	0
SNAP	0	0
NetBIOS	0	0
IP	0	0
802.1x EAP-Packet	0	0
ARP	0	0
802.11 QoS Data	0	0
802.11 Management	0	0
802.11 Data	0	0
802.11 Control	0	0
IEEE 802.11	0	0

Table A.13 Line traffic parameters during various times/dates

Traffic Parameter	Parameter Values	Traffic Parameter	Parameter Values
Packets	32,147	Packets	19,524
Bytes	21,745,794	Bytes	15,501,610
Monitor Duration	411 sec	Monitor Duration	270 sec
Average packets/sec	78.059	Average packets/sec	72.294
Average Mbit/Sec	0.422	Average Mbit/Sec	0.459
Average bytes/sec	52,802.949	Average bytes/sec	57,399.443
Average packet size	676 bytes	Average packet size	793 bytes
Traffic Parameter	**Parameter Values**	**Traffic Parameter**	**Parameter Values**
Packets	16,956	Packets	13,794
Bytes	13,466,250	Bytes	11,507,313
Monitor Duration	429 sec	Monitor Duration	169 sec
Average packets/sec	39.506	Average packets/sec	81.371
Average Mbit/Sec	0.251	Average Mbit/Sec	0.543
Average bytes/sec	31374.942	Average bytes/sec	67,882.096
Average packet size	794 bytes	Average packet size	834 bytes
Traffic Parameter	**Parameter Values**	**Traffic Parameter**	**Parameter Values**
Packets	16,653	Packets	21,746
Bytes	13,179,228	Bytes	9,887,008
Monitor Duration	242 sec	Monitor Duration	499 sec
Average packets/sec	68.804	Average packets/sec	43.511
Average Mbit/Sec	0.436	Average Mbit/Sec	0.158
Average bytes/sec	54,451.554	Average bytes/sec	19,782.727
Average packet size	791 bytes	Average packet size	454 bytes
Traffic Parameter	**Parameter Values**	**Traffic Parameter**	**Parameter Values**
Packets	16,380	Packets	25,210
Bytes	6,052,500	Bytes	14,647,180
Monitor Duration	430 sec	Monitor Duration	488 sec
Average packets/sec	38.021	Average packets/sec	51.581
Average Mbit/Sec	0.112	Average Mbit/Sec	0.240
Average bytes/sec	14,048.944	Average bytes/sec	29,968.867
Average packet size	369 bytes	Average packet size	581 bytes

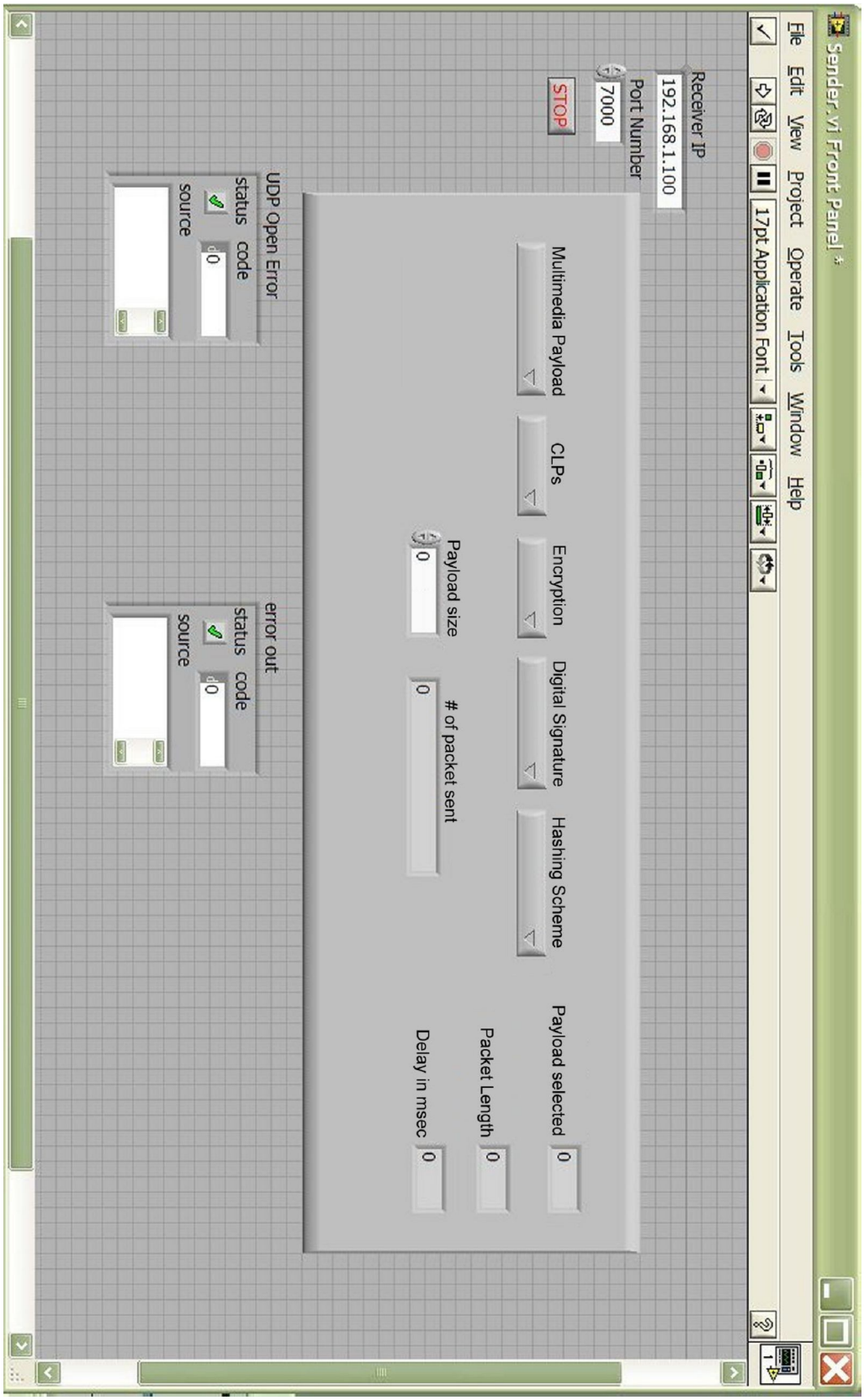

Figure A.2 Labview 8.5 Suite-B testbed graphical user interface

References

[1] "NSA Suite B Cryptography", National Security Agency, Central Security Service, Retrieved on December 14th, 2009, http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml

[2] Hydra PC FIPS File Encryption Module Security Policy, Revision Document No. 07, October 22, 2009, http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1179.pdf

[3] Crypto++® Library 5.6.0, Retrieved on August 24th, 2009, http://www.cryptopp.com/

[4] Labview, National Instrument, http://www.ni.com/labview/

[5] E D Puschita, T P Palade, "QoS Perspective for Wireless Scenarios", Broadband Europe 2005, Whitepapers, 5 pages, Bordeaux, 12-14 December 2005

[6] K. Kilkki, "Next Generation Internet and Quality of Experience", EuroFGI IA.7.6 Workshop on Socio-Economic Issues of NGI, Santander, Spain, June, 2007

[7] Tang, J. and Zhang, X., "Cross-layer design of dynamic resource allocation with diverse QoS guarantees for MIMO-OFDM wireless networks," IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, 2005, pp. 205-212

[8] S. Pasupuleti, D. Das, "Throughput and delay evaluation of a proposed-DCF MAC protocol for WLAN", Proceedings of IEEE INDICON 2004, 4 pages, December 2004, http://dit.unitn.it/~srinivas/INDICON_IIITB.pdf

[9] W. Grote, A. Grote, I. Delgado "IEEE 802.11 Goodput Analysis for Mixed Real-time and Data Traffic for Home Networks", Annals of Telecommunications; Vol. 63 Nº 9/10, pp. 463-471; 2008

[10] Pai-Hsiang Hsiao, H.T. Kung, and Koan-Sin Tan, "Streaming Video over TCP with Receiver-based Delay Control", IEICE Transactions on Communications, Vol. E86-B, No. 2. 2003, pp. 572-584.

[11] Aura Ganz, Zvi Ganz, Kitti Wongthavarawat, "Multimedia Wireless Networks: Technologies, Standards, and QoS", Prentice Hall Publisher, ISBN: 978-0130460998, September 18, 2003

[12] Odd Inge Hillestad, Bjornar Libak, Andrew Perkis, "Performance Evaluation of Multimedia Services over IP Networks", in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Amsterdam, The Netherlands, July 06-08, 2005, pp. 1464 – 1467

[13] Hans Olaf Rutger Thomschutz, "Security in Packet-Switched Land Mobile Radio Backbone Networks", Master of Science Thesis at Virginia Polytechnic Institute and State University, May 2005

[14] S. Adibi, G. B. Agnew, “Security Measures for Mobile Ad-Hoc Network (MANETs)”, Handbook of Research on Wireless Security, IGI Global Inc., ISBN 978-1-59904-899-4, March 2008

[15] D. Zhu, "Security Control in Inter-Bank Fund Transfer", Journal of Electronic Commerce Research, Vol. 3, No. 1, pp. 15-22, 2002

[16] Muhammad Sher, Thomas Magedanz, "Developing Network Domain Security (NDS) Model for IP Multimedia Subsystem (IMS)”, Journal of Networks, Vol. 1, No. 6, pp. 10-17, November/December 2006

[17] Lidong Zhou, Zygmunt J. Haas, "Securing Ad Hoc Networks", IEEE Networks Special Issue on Network Security, Vol. 13, No. 6, pp.24-30, November/December, 1999

[18] Pekka Kanerva, "Anonymous Authorization in Networked Systems: An Implementation of Physical Access Control System", Master’s Thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Espoo 2001

[19] Rong Ren; Deng-Guo Feng; Ke Ma, "A detailed implement and analysis of MPLS VPN based on IPSec", Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Vol. 5, Issue, 26-29 Aug. 2004 Page(s): 2779 - 2783

[20] I.Yiakoumis, M.Papadonikolakis, and H.Michail. Efficient small sized implementation of the keyed-hash message authentication code. In Proceedings of the IEEE Eurocon Conference 2005 Computer as a Tool, volume 2, pages 1875-1878, Belgrade, November 2005

[21] Soo-Young Kang and Im-Yeong Lee, "A Study on Low-Cost RFID System Management with Mutual Authentication Scheme in Ubiquitous", Managing Next

Generation Networks and Services, Springer Link Publication, ISBN: 978-3-540-75475-6, September 18, 2007

[22] G.B. Agnew, ECE628 Computer Network Security Course Notes, Winter 2006

[23] Ryusuke Koide, Tomoyuki Nagase and Takashi Araki, "QHF: A Quaternion-based a Multidimensional Hash Function", Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems, Xiamen, China, pp. 830 – 833, Nov.28-Dec.1, 2007

[24] Shoichi Hirose, "Weak Security Notions of Cryptographic Unkeyed Hash Functions and their Amplifiability", Special Section on Cryptography and Information Security, IEICE Transaction of Fundamentals, Vol. E88-A, No. 1 January 2005

[25] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", Rump Session, CRYPTO 2004, Cryptology ePrint Archive, Report 2004/199, pp. 1-4, IACR 2004 Cryptography Conference, Santa Barbara, USA

[26] Gaëtan Leurent, "Message Freedom in MD4 and MD5 Collisions: Application to APOP", In Proceedings of FSE, LNCS 4593, pp. 309-328, Springer, 2007

[27] Xiaoyun Wang and Hongbo Yu, "How to Break MD5 and Other Hash Functions", EUROCRYPT 2005, http://www.infosec.sdu.edu.cn/uploadfile/papers/How%20to%20Break%20MD5%20and%20Other%20Hash%20Functions.pdf

[28] A. Levitin, "Brute Force", Chapter 3, Introduction to the Design & Analysis of Algorithms, 2nd ed., Pearson Addison-Wesley, ISBN: 978-0321358288, 2007, http://www.cs.ucr.edu/~jiang/cs141/ch03n.ppt

[29] SHA hash functions, Wikipedia, http://en.wikipedia.org/wiki/SHA1, Retrieved on May 1st, 2009

[30] W. Diffie and M. E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, Vol. IT-22, Nov. 1976, pp: 644–654

[31] Carlos M. Gutierrez, Patrick Gallagher, "Digital Signature Standard (DSS)", DRAFT FIPS PUB 186-3, FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Information Technology Laboratory, National

Institute of Standards and Technology, U.S. Department of Commerce, November, 2008, http://csrc.nist.gov/publications/drafts/fips_186-3/Draft_FIPS-186-3%20_November2008.pdf

[32] Don Johnson, Alfred Menezes, and Scott Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom Corp., http://www.comms.scitech.susx.ac.uk/fft/crypto/ecdsa.pdf

[33] Marcel Medwed, Elisabeth Oswald, “Template Attacks on ECDSA”, WISA 2008, http://www.comms.scitech.susx.ac.uk/fft/crypto/ecdsa.pdf

[34] Alfred Menezes, “Evaluation of Security Level of Cryptography: RSA-OAEP, RSA-PSS, RSA Signature”, pp. 1-29, University of Waterloo, Dec 2001, http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1011_rsa.pdf

[35] G. B. Agnew, R. C. Mullin, S. A. Vanstone, “An implementation of elliptic curve cryptosystems over F_2155”, IEEE Journal on Selected Areas in Communications, 11(5):804-813, June 1993

[36] M. Aydos, B. Sunar, C. K. Koc , “An Elliptic Curve Cryptography based Authentication and Key Agreement Protocol for Wireless Communication”, The 2nd International Workshop on Discrete Algorithms and Methods for Mobility (DIALM 98), pp. 1-12, Dallas, Texas, October 30, 1998

[37] Stefaan Seys, "Lightweight Cryptography Enabling Secure Wireless Networks", Workshop on Security Issues in Mobile and Wireless Heterogeneous Networks, Brussels – December 6, 2004

[38] A. H. Lashkari, M. Mansoor, A. S. Danesh, "Wired Equivalent Privacy (WEP) versus Wi-Fi Protected Access (WPA)", IEEE International Conference on Signal Processing Systems, Singapore, pp. 445 – 449, May 15-17, 2009

[39] Thomas d'Otreppe de Bouvette, "WPA, what else?", Aircracking, UNAM, Mexico City, Nov. 27-28, 2008, http://intromision.fciencias.unam.mx/1_UNAM_08.ppt

[40] Joerg Gruenauer, "Wireless network security standard", June 2005, http://joerg.gruenauer.doesntexist.org/WLANSecurity_V2.ppt

[41] Murray Stokely, "The Insecurity of 802.11 WEP", CS6520 Cryptography, Aug 2003, http://www.mcs.csuhayward.edu/~pwong/cs6520_sum03/wep.pdf

[42] Levi Portillo, Zhan Liu, “WEP Flaws and Implementation Flaws of Authentication Protocols”, ELEN 689, Texas A&M University, April 2006, http://www.ece.tamu.edu/~reddy/ee689_06/levi-zhan.pdf

[43] L. Bernaille, R. Teuxeira, I. Akodkenou, A. Soule, K. Salamatian, “Traffic Classification on the Fly”, ACM SIGCOMM Computer Communication Review, Vol. 36, No. 2, April 2006

[44] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, “BLINC: Multilevel Traffic Classification in the Dark”, ACM Sigcomm 2005, pp. 229–240, Philadelphia, PA, USA, August 2005

[45] Andrew W. Moore and Konstantina Papagiannaki, "Toward the Accurate Identification of Network Applications", Passive and Active Measurements Workshop, LNCS 3431, pp. 41-54, Boston, MA, USA, March 31 - April 1, 2005

[46] Cisco IOS Documentation, “Network-Based Application Recognition and Distributed Network-Based Application Recognition“, (as of February 2005).

[47] K. Lan, J. Heidemann, “On the correlation of Internet flow characteristics”, Technical Report ISI-TR-574, USC/Information Sciences Institute, July, 2003

[48] Laurent Bernaille, Renata Teixeira, “Implementation Issues of Early Application Identification”, AINTEC 2007, LNCS 4866, pp. 150-166, 2007

[49] “App-ID™ Application Classification - Technology Overview”, Paloalto Networks, June, 2007, http://www.paloaltonetworks.com/literature/docloader.php?docURL=/literature/whitepapers/App-ID_overview.pdf&docName=PDF%3A+App-ID+Application+Classification+Technology

[50] Thomas Karagiannis, Andre Broido, Nevil Brownlee, Kc Claffy, “Is P2P dying or just hiding?”, Proceedings of IEEE Globecom 2004, Vol. 3, pp. 1532 – 1538, Nov 29 – Dec 3, 2004

[51] S. Sen, O. Spatscheck, D. Wang, “Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures”, 13th International World Wide Web Conference (WWW 2004), pp. 512–521, New York City, 17-22 May 2004

[52] M. Grega, L. Janowski, M. Leszczuk, P. Romaniak, Z. Papir, “Quality of Experience Evaluation for Multimedia Services”, Przeglad Telekomunikacyjny, Vol. 81, No. 4, pp. 142–153, 2008

[53] A. Madhukar and C. Williamson “A longitudinal study of P2P traffic classification”, Proceedings of the 14th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS’ 2006), pp. 179–188, Monterey, CA, September 11–13, 2006

[54] Denis Zuev, Andrew Moore, "Internet Traffic Classification using Bayesian Analysis Techniques", ACM SIGMETRICS 2005, pp. 50–60, Banff, Canada, June, 2005

[55] Denis Zuev, Andrew Moore, “Traffic Classification using a Statistical Approach”, Passive and Active Measurement (PAM ‘05) Workshop, pp. 321–324, Boston, USA, April 2005

[56] James P. Early, Carla E. Brodley, and Catherine Rosenberg, "Behavioral Authentication of Server Flows", Proceedings of the 19th Annual Computer Security Applications Conference (ACDAC ’03), pp. 46-55, Las Vegas, NV, USA, December 2003

[57] Jefferey Erman, Anirban Mahanti and Martin Arlitt, "Byte Me: A Case for Byte Accuracy in Traffic Classification", Proceeding of the 3rd annual ACM workshop on Mining network data (MineNet ’07), pp. 35-38, 2007

[58] M. Roughan, S. Sen, O. Spatscheck, N. Duffield, “Class-of-Service Mapping for QoS: A statistical signature-based approach to IP traffic classification”, ACM SIGCOMM Internet Measurement Workshop, pp. 135-148, Taormina, Italy, 2004.

[59] A. Soule, K. Salamatian, N. Taft, R. Emilion, and K. Papagiannaki, “Flow Classification by Histograms or How to Go on Safari in the Internet”, In ACM SIGMETRICS Performance Evaluation Review 32, pp. 49-60, New York, USA, June, 2004

[60] “PA-4000 Series Feature Overview”, Paloalto Networks, June 2007, http://www.koresecurity.com/content/papers/PA4000_Series_overview.pdf

[61] H. Chang, S. Jamin, Z. Mao, and W. Willinger, "An Empirical Approach to Modeling Inter-AS Traffic Matrices", Proceedings of ACM Internet Measurement Conference (IMC '05), pp. 139–152, 2005

[62] A. McGregor, M. Hall, P. Lorier, J. Brunskill, "Flow Clustering Using Machine Learning Techniques", Passive and Active Measurement Workshop (PAM '04), pp. 205-214, France, April 19-20, 2004

[63] Thuy T.T. Nguyen, Grenville Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning", IEEE Communications Surveys and Tutorials, Vol. 10 No. 4, pp. 56-76, 2008

[64] T. Dunnigan, G. Ostrouchov, "Flow Characterization for Intrusion Detection", Oak Ridge National Laboratory, Technical Report, November 2000, http://www.csm.ornl.gov/~ost/id/tm.ps

[65] Eamonn Linehan, "Internet Worm Detection as part of a Distributed Network Inspection System", Master Thesis in Computer Science Department, University of Dublin, 2004

[66] Laurent Bernaille, Renata Teixeira, "Early Recognition of Encrypted Applications", Passive and Active Measurement Conference (PAM), pp. 165-175, Louvain-La-Neuve, Belgium April, 2007,

[67] S. McCreary, K. Claffy, "Trends in Wide-Area IP Traffic Patterns - A View from Ames Internet Exchange", in the Proceedings of the ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management, Monterey, CA, USA, Sep 2000, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.6038&rep=rep1&type=pdf

[68] Alessandro Amirante, "Robust and Efficient traffic Classification in IP nEtworks", Recipe, December 2007, http://www.comics.unina.it/index.php?option=com_content&task=view&id=115&Itemid=164

[69] Ahsan Habib, Sonia Fahmy, and Bharat Bhargava, "Monitoring and controlling QoS network domains", International Journal of Network Management, Int. J. Network Mgmt 2005; 15: 11–29

[70] Martin Thielen, "Design and Development of a Quality of Service Framework for the Network-Integrated Multimedia Middleware (NMM), Master Thesis, Saarland University, May 30, 2007

[71] Aura Ganz, Zvi Ganz, Kitti Wongthavarawat, "Multimedia Wireless Networks: Technologies, Standards, and QoS", Prentice Hall Publisher, ISBN: 978-0130460998, September 18, 2003

[72] "Overcoming Barriers to High-Quality Voice over IP Deployments", Intel Whitepaper, 2003, http://www.clccorp.com/whitepapers/Overcoming%20Barriers%20VOIP.pdf

[73] "Voice Codecs", Voice over IP - Tested by Uninett AS, UNINETT – The Norwegian research network, January 14, 2004, http://forskningsnett.uninett.no/voip/codec.html,

[74] "Video Conferencing Standards", Tandberg, Application Notes, D10740, Rev 2.3, http://www.tandberg.com/collateral/white_papers/whitepaper_Videoconferencing_standards.pdf

[75] Thomas Wiegand, "Video Coding Standards", Digital Image Communication, http://iphome.hhi.de/wiegand/assets/pdfs/DIC_video_coding_standards_07.pdf

[76] Pankaj Topiwala, Munish Jindal, "H.264/AVC: Overview and Intro to Fidelity-Range Extensions", FastVDO, 2004, http://www.ti.com/asia/docs/india/tiidevconf2004/analog_symp/munish.pdf

[77] Peter Pocta, "Impact of Multimedia Services Interaction on Speech Quality in VoIP & VoWLAN", ETSI Workshop of Effects of transmission performance on Multimedia Quality of Service, Prague, Czech Republic, 17 - 19 June 2008

[78] Phil Karn, Craig Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", Proceedings of ACM SIGCOMM '87 11-13 August 1987, pp 2-7

[79] "Voice over IP - Per Call Bandwidth Consumption", Cisco Systems, Document ID: 7934, http://www.cisco.com/application/pdf/paws/7934/bwidth_consume.pdf

[80] Bandwidth Calculator for VoIP, AsteriskGUIDE, 2006, http://blog.asteriskguide.com/bandcalc/bandcalc.php

[81] “VoIP Bandwidth Calculation, Newport Networks”, http://www.newport-networks.com/whitepapers/voip-bandwidth3.html

[82] Sven Wiethoelter, "Virtual Utilization and VoIP Capacity of WLANs Supporting a Mix of Data Rates", Technical University Berlin, Technical Report, September 2005

[83] “Packetcable™ Audio/Video Codecs Specification”, Packetcable™ 1.0 Specifications, PKT-SP-CODEC-I05-040113, Cable Television Laboratories, Inc., 2004, http://www.cable-labs.net/specifications/archives/PKT-SP-CODEC-I05-040113.pdf

[84] Steve Heath, “Multimedia and Communications Technology”, Second Edition, Focal Press, ISBN: 978-0240515298, August 1999

[85] Jerry D. Gibson, Toby Berger, Tom Lookabaugh, Dave Lindgergh, Richard L. Baker, “Digital Compression for Multimedia”, Principles and Standards, Morgan Kaufmann, ISBN: 978-1558603691, January 1998

[86] Eric D. Siegel, “Designing Quality of Service Solutions for the Enterprise”, Wiley Computer Publishing, John Wiley & Sons, ISBN: 978-0471333135, October 1999

[87] Dimitrios Miras, "A Survey on Network QoS Needs of Advanced Internet Applications", omputer Science Department, University College London, November 2002, http://qos.internet2.edu/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.html

[88] Mean Opinion Score, Wikipedia, Retrieved on May 7, 2009, http://en.wikipedia.org/wiki/Mean_Opinion_Score

[89] G. Boccolini, M. Luise, B. Garnier, J. M. Merour, A. Brunelle, S. Titomanlio, V. Mignone, M.A. Sasse, "A two-way interactive broadband satellite architecture to break the digital divide barrier", 16th Ka and Broadband Communications Conference, pp 1-10, Sep. 24-26, 2007, Turin, Italy

[90] Jirka Klaue, Andreas Hess, “On the Impact of IPsec on Interactive Communications”, Proc. of 19th International Parallel and Distributed Processing Symposium (IPDPS), Workshop 17 – Vol. 18, Page: 291.1, April 2005

[91] Miroslav V, F. Hromek, “Analytic Model of Delay Variation Valid for RTP”, CESNET technical report number 16/2007, Nov. 2007

[92] Nikolay Scarbnik, “IP VPN”, http://www.it.lut.fi/kurssit/07-08/CT30A2300/Verkkomateriaali/Day3/Day3-luento-16_IP%20VPN.pdf

[93] Austin Godber, Partha Dasgupta, “Secure Wireless Gateway”, Proceedings of the 1st ACM workshop on Wireless security Atlanta (WiSe-02), pp. 41 - 46, GA, USA, 2002

[94] Jenne Wong, “Performance Investigation of Secure 802.11 wireless LANs”, Master of Commerce in Accountancy, Finance, and Information Systems, University of Canterbury, 2003

[95] Heiko Niedermayer, Andreas Klenk, and Georg Carle, "The Networking Perspective of Security Performance - a Measurement Study”, 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems, MMB 2006, http://www.net.in.tum.de/fileadmin/bibtex/publications/papers/niedermayer-mmb06.pdf

[96] Ken Masica, "Recommended Practices Guide Securing WLANs using 802.11i", Vulnerability & Risk Assessment Program (VRAP), Lawrence Livermore National Laboratory (LLNL), October 2006

[97] Thomas Otto, “Extensible Network Access Authentication", Master of Science Thesis, Institute of Operating Systems and Computer Networks, Technical University of Braunschweig, July 4, 2006

[98] Yao Zhao, Chuang Lin, Hao Yin, "Security Authentication of 3G-WLAN Interworking", Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA ’06), Vol. 02, pp. 429 – 436, 2006

[99] Ingo Riedel, “Security in Ad-hoc Networks: Protocols and Elliptic Curve Cryptography on an Embedded Platform”, Diploma Thesis, Ruhr-Universitat Bochum, March 2003

[100] Levente Buttyan, Jean-Pierre Hubaux, "Chapter 1: The security of existing wireless networks", Security and Cooperation in Wireless Networks, http://www.cs.gmu.edu/~setia/cs818/lectures/lecture3.pdf

[101] Sohail Hirani, “Energy Consumption of Encryption Schemes in Wireless Devices”, Master of Science in Telecommunications, University of Pittsburgh, 2003

[102] Sandrine Duflos, Brigitte Kervella, Valerie Gay “Considering Security and Quality of Service in SLS to improve Policy-based Management of Multimedia Services”, Proceedings of the Sixth International Conference on Networking (ICN ’07), Page: 39, 2007

[103] “ProCurve Wireless LAN Security”, Fundamentals Guide Technical Training Version 8.21, HP Press, May 2008

[104] Srivaths Ravi, Anand Raghunathan, Paul Kocher, Sunil Hattangady, "Security in embedded systems: Design challenges", ACM Transactions on Embedded Computing Systems (TECS), Volume 3 , Issue 3, Pages: 461 - 491, 2004

[105] Qiang Ni, Lamia Romdhani, Thierry Turletti, "A survey of QoS enhancements for IEEE 802.11 wireless LAN: Research Articles", Wireless Communications and Mobile Computing, Volume 4 , Issue 5, Pages: 547 - 566, 2004

[106] “List of WLAN channels”, Wikipedia, Retrieved on May 10, 2009, http://en.wikipedia.org/wiki/List_of_WLAN_channels

[107] “Testing Transmitted Signals for Compliance with IEEE 802.11a WLAN Standards”, Technical Brief, Tektronix WLAN Compliance Test, Retrieved on June 24th, 2009, http://www.nortelco.no/default.asp?FILE=items/1028/249/Tektronix%20WLAN%20compliance%20test.pdf

[108] “Wireless Video System Design - A Comprehensive Reference for the System Integrator”, Version 1.0, VERINT SYSTEMS INC., January 2008, http://demossolutions.com/video_solutions/file.cfm?id=350

[109] M. Haidar, R. Ghimire, H. M. Al-Rizzo, R. Akl, and Y. Chan, "Channel Assignment in an IEEE 802.11 WLAN based on Signal-to-Interference Ratio," 21st IEEE Canadian Conference on Electrical and Computer Engineering, Paper No. 1569092894, 6 pages, May 4-7, 2008, Niagara Falls, ON, Canada

[110] K. Navaie, S. Valaee, E. Sousa , “On the downlink interference in heterogeneous wireless DS-CDMA networks”, IEEE Transactions on Wireless Communications, pp. 384 – 393, Volume: 5 Issue: 2, March 2006

[111] Q. Wang, M.A. Abu-Rgheff, “Cross-Layer signaling for Next-Generation Wireless Systems”, Proceedings of the IEEE Wireless Communication and Network Conference (WCNC '03), Vol. 2 , pp. 1084-1089, New Orleans, LA, USA, March 16-20, 2003

[112] Fei Yu, Vikram Krishnamurthy, Victor C. M. Leung, “Cross-Layer Optimal Connection Admission Control for Variable Bit Rate Multimedia Traffic in Packet Wireless CDMA Networks”, IEEE Global Telecommunications Conference, (GLOBECOM '04), Vol. 5, pp. 3347 – 3351, Nov 29 – Dec 3, 2004

[113] Li. Xiaohua, J. Hwu E. P. Ratazzi, "Array Redundancy and Diversity for Wireless Transmissions with Low Probability of Interception", Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06), Vol. 4, Page: IV, May 14-19, 2006

[114] Y. P. Fallah, H. Alnuweiri, "A controlled-access scheduling mechanism for QoS provisioning in IEEE 802.11e wireless LANs", International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, the QoS provisioning in wireless multimedia session, pp. 122 - 129, Montreal, Quebec, Canada, 2005

[115] Ritun Patney, Raj Jain, "Scheduling in WiMAX Scheduling in WiMAX Networks", Presented at WiMAX Forum AATG F2F Meeting, Washington DC, April 26, 2007, http://www.cs.wustl.edu/~jain/wimax/ftp/sch704c2.pdf

[116] Shivaputrappa Vibhuti, "IEEE 802.11 WEP Concepts and Vulnerability", San Jose State University, CA, USA, CS265 Spring 2005, http://www.cs.sjsu.edu/faculty/stamp/CS265/projects/Spr05/papers/WEP.pdf

[117] IEEE 802.11i-2004, Wikipedia, Retrieved on May 15, 2009, http://en.wikipedia.org/wiki/WPA2

[118] Karthikeyan Mahadevan, "Security Considerations for IEEE 802.15.4 Networks", http://www.cs.umn.edu/research/sclab/Slides/10_01.ppt

[119] Peter J. Welcher, "Introduction to IPSec VPN's", Chesapeake Netcraftsmen, 2003, http://www.netcraftsmen.net/welcher/seminars/intro-ipsec.pdf

[120] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, Aug 2008, http://tools.ietf.org/html/rfc5246

[121] Ozcelebi, T., Sunay, M.O., Civanlar, M.R., Tekalp, A.M., "Application-Layer QoS Fairness in Wireless Video Scheduling", Proceedings International Conference on Image Processing (ICIP '06), Vol. 1-7, pp 1673-1676

[122] Nicola Cranley, Liam Murphy, "Handbook of Research on Wireless Multimedia: Quality of Service and Solutions", Information Science Reference, IGI Global, ISBN: 978-1599048208, July 28, 2008

[123] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, John M. Peha, "Streaming Video over the Internet: Approaches and Directions", IEEE Transactions on Circuit and Systems for Video Technology, Vol. 11, No. 3, pp. 282-300, March 2001

[124] J. Rosenberg, et al., "SIP: Session Initiation Protocol", RFC 3261, June 2002

[125] "Assuring Quality of Experience for IPTV — The Role of Video Admission Control", Alcatel-Lucent Application Note, September 2007

[126] Erno Alhoniemi, "Error Detection and Control in Data Transfer", Helsinki University of Technology, Nov 22, 1998, http://www.tml.tkk.fi/Studies/Tik-110.300/1998/Essays/error_detection.html

[127] "Cross-Layer Design of Ad-hoc Wireless Networks for Real-Time Media", http://www.stanford.edu/~zhuxq/adhoc_project/adhoc_project.html

[128] F. Liu, J. Kim, C.-C.J. Kuo, "Adaptive delay concealment for Internet voice applications with packet-based time-scale modification", Proceedings of IEEE International Conference on Communications, Proceedings of the Acoustics, Speech, and Signal Processing (ICASSP '01), Vol. 3, pp. 1461-1464, 2001

[129] Bong Ho Kim, Tom Kavanaugh, "QoS API for the Signaling between Upper layer and MAC in MS", WiMAX Forum Press, 2007

[130] F. Guo, "Traffic Analysis: from Stateful Firewall to Network Intrusion Detection System, " RPE Report, January 2004

[131] Time Szigeti, Christina Hattingh, "End-to-End QoS Network Design: Quality of Service in LAN's WANs, and VPNs", Cisco System Press, November 2004

[132] “Packetcable Dynamic Quality-of-Service Specification”, PKT-SP-DQOS-I12-050812, http://www.packetcable.com/downloads/specs/PKT-SP-DQOS-I12-I05-050812.pdf

[133] Alam M., Prasad R., and J. R. Farserotu, “Quality of service among IP-based heterogeneous networks”, IEEE Personal Communications, Vol. 8, No. 6, pp. 18-24, December 2001

[134] Andreas Kassler, Teodora Guenkova-Luy, Davide Mandato, Tomas Robles, "E2ENP: An End-to-End QoS Negotiation Protocol", International Workshop on Mobile-IP-based Network Development, pp 1-7, London UK, October 2002

[135] Sasu Tarkoma, “Network Application Frameworks and XML Summary and Conclusions”, T-110.5140, April 2008, http://www.tml.tkk.fi/Opinnot/T-110.5140/2008/Lectures/naf_lecture_220408.pdf

[136] Yuan Xue, "Email Security", CS 291 Network Security, Vanderbilt University, 2006, http://vanets.vuse.vanderbilt.edu/~xue/cs291fall06/email.pdf

[137] Tom Wennerstrom, Jens Jespersen, Magnus Lundquist, "Simple Object Access Protocol - A basic overview", University of Uppsala, September 2002, http://user.it.uu.se/~hsander/Courses/DistributedSystems/Reports/soap_report_2.pdf

[138] Juan J Vargas, "SOAP (Simple Object Access Protocol), An Introduction", University of Central Florida, CDA 5937 Fall 2002, http://www.cs.ucf.edu/~dcm/Teaching/ProcessCoordination/Fall02Class/ResearchPresentations/JuanVargas.ppt

[139] U. Kozat, A. Begen, "Pseudo Content Delivery Protocol (CDP) for Protecting Multiple Source Flows in FEC Framework", Internet Draft, July 7, 2008, ftp://ftp.mimuw.edu.pl/mirror/ftp.rfc-editor.org/internet-drafts/draft-kozat-fecframe-pseudo-cdp-00.txt

[140] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery", RFC 5053, October 2007

[141] B. Fong, P.B. Rapajic, G.Y. Hong, A.C.M. Fong, "Forward error correction with

Reed-Solomon codes for wearable computers", IEEE Transactions on Consumer Electronics, 49(4):917- 921, December 2003

[142] G. Tan, T. Herfet, "Application Layer Hybrid Error Correction with Reed-Solomon Code for DVB Services over Wireless LANs", 3rd International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2007), pp. 2952-2955, Shanghai, China, September 21-23, 2007

[143] Pankaj Bhagawat, Rajballav Dash, Gwan Choi, "Systolic Like Soft-Detection Architecture for 4x4 64-QAM MIMO System", IEEE Design Automation and Test in Europe (DATE '09), pp. 870-873, April 20-24, 2009, Nice, France

[144] F. S. Filho, E. H. Watanabe, E. de Souza e Silva, "Adaptive forward error correction for interactive streaming over the Internet", Proceedings of the IEEE GLOBECOM, pp. 1-6, November 2006

[145] J. Fernandez-González, Gordon B. Agnew, Arturo Ribagorda, "Encryption and error correction codes for reliable file storage", Computers and Security 12(5): 501-510, 1993

[146] C. N. Mathur, K. Narayan, and K. P. Subbalakshmi, "High diffusion cipher: encryption and error correction in a single cryptographic primitive," in Proceedings of the 4th International Conference on Applied Cryptography and Network Security (American Conference on Neutron Scattering), Vol. 3989, pp. 309–324, Singapore, June 2006

[147] Chetan Nanjunda Mathur, "A Mathematical Framework for Combining Error Correction and Encryption", Winner of the Best Dissertation Award, Stevens Institute of Technology, May 2007

[158] R. Housley, "Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)", RFC 4309, Dec. 2005, http://www.rfceditor.org/rfc/rfc4309.txt

[159] S. Spinsante, F. Chiaraluce, E. Gambi, "Evaluation of AES-based authentication and encryption schemes for Telecommand and Telemetry in satellite applications", Proceedings SpaceOps 2006 Conference, Paper AIAA 2006-5558, Rome, Italy, 19-23 June 2006

[150] “Technical Document on Overview – Wireless, Mobile and Sensor Networks”, GDD-06-14 - Ver. 2.0, GENI: Global Environment for Network Innovations, Sep. 2006, http://www.geni.net/GDD/GDD-06-14.pdf

[151] Wesley M. Eddy, "At What Layer Does Mobility Belong?", IEEE Communications Magazine, Volume: 42 Issue: 10, pp. 155 – 159, Oct. 08, 2004

[152] Jun Zhao, Zihua Guo, Wenwu Zhu, “ Power Efficiency in IEEE 802.11a WLAN with Cross-Layer Adaptation” , IEEE ICC 2003, pages 20-30, 2003

[153] Ki-Ho Lee “A Multiple Access Collision Avoidance Protocol for Multicast Services in Mobile Ad Hoc Networks”, IEEE Communications letters, Vol. 7, No. 10, pp. 508-510, October 2003

[154] Zhi Li, Yong Lian, Qibin Sun, Chang Chen, “Authenticating Multimedia Transmitted Over Wireless Networks - A Content-Aware Stream-Level Approach”, ICME 2006: 545-548, 2006

[155] Shushan Zhao, Akshai Aggarwal, Shuping Liu, "Building Secure User-to-user Messaging in Mobile Telecommunication Networks", Wireless Telecommunications Symposium (WTS'08), 2008, pp. 151 - 157, Pomona, California, USA, April 2008

[156] Joachim Wilke, Erik-Oliver Blass, Martina Zitterbart, “ESAWN-NR: Authentic Aggregation and Non-Repudiation in Wireless Sensor Networks”, Proceedings of Fifth International Conference on Networked Sensing Systems (INSS 2008), Page 254, Kanazawa, Japan, Jun 2008

[157] Huang Ming-Way, Chen Hsing-Bai, Lee Wei-Bin, "An Efficient Non-repudiation Mechanism for SIP-Based Services", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2008 (IIHMSP 08), Page(s):1541 – 1544, August 15-17, 2008

[158] K.-W. Park, H. Seok, and K.-H. Park, “PKASSO: Towards Seamless Authentication Providing Non-Repudiation on Resource-Constrained Devices,” Proc. 21st IEEE Int'l Conf. Advanced Information Networking and Applications Workshops, Vol. 2, pp. 105-112, 2007

[159] Riaz Ahmed Shaikh, Sungyoung Lee, Young Jae Song, and Yonil Zhung, “Securing Distributed Wireless Sensor Networks: Issues and Guidelines”, Proc. of

IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2006), Vol. 2, pp. 226-231, Taiwan, June 2006

[160] Ulf Lamping, Richard Sharpe, “Wireshark User’s Guide – 31574 for Wireshark 1.2”, 2008, http://www.wireshark.org/download/docs/user-guide-a4.pdf

[161] Omnipeek™ Personal, Free Portable Network Analyzer, 2006 http://www.wildpackets.com/elements/omnipeek/OmniPeek_Personal.pdf

[162] Scatter Plot Tool, Mathcracker, http://www.mathcracker.com/scatter_plot.php

[163] David von Oheimb, "Formal Security Analysis", Information and Communications Security, Siemens Corporate Technology, Munich, Germany, 2006, http://david.von-oheimb.de/cs/talks/Lecture_FormalSecAna.pdf

[164] Peter Robinson, "Developing TLS Applications with Suite-B Cipher Suites", RSA, The Security Division of EMC, Session Code: DEV-203, October 28, 2008

[165] “NSA Suite B Cryptography”, Retrieve on July 5th, 2009, http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml

[166] “NSA Suite B Cryptography”, Wikipedia, Retrieved on July 5th, 2009, http://en.wikipedia.org/wiki/NSA_Suite_B_Cryptography

[167] Ian F. Blake, Gadiel Seroussi, Nigel Paul Smart, "Advances in elliptic curve cryptography", Cambridge University Press; 2nd edition, ISBN: 978-0521604154, May 31, 2005

[168] Arnab Roy, Anupam Datta, John C. Mitchell, “Formal Proofs of Cryptographic Security of Diffie-Hellman based Protocols”, Proceedings of Symposium On Trustworthy Global Computing (TGC ’07), LNCS 4912, pp. 312-329, November 2007

[169] R. Housley, “Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS)”, RFC 5084, November 2007, http://www.ietf.org/rfc/rfc5084.txt

[170] “Carry-Less Multiplication and Its Usage for Computing the GCM Mode”, Intel Software Network, Last Modified on May 12, 2009, http://software.intel.com/enus/articles/carry-less-multiplication-and-its-usage-for-computing-the-gcm-mode

[171] T. Iwata, “Comparison of CBC MAC variants and comments on NIST's Consultation Paper”, Department of Computer and Information Sciences, Ibaraki University, a comment to NIST, May 5, 2003, http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/800-38_Series-Drafts/RMAC/Iwata_comments.pdf

[172] David A. McGrew, John Viega, “The Security and Performance of the Galois/Counter Mode (GCM) of Operation”, 5th International Conference on Cryptology (INDOCRYPT ‘04), Vol. 3348, pp. 343-355, Chennai, India, December 20-22, 2004

[173] Vipul Gupta, Douglas Stebila, Stephen Fung, Sheueling Chang Shantz, Nils Gura, Hans Eberle, “Speeding up Secure Web Transactions Using Elliptic Curve Cryptography”, NDSS 2004, http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Gupta.pdf

[174] Brown, Daniel R. L., “The exact security of ECDSA,” University of Waterloo, Technical Report CORR 2000-54, Certicom Research, 2000

[175] "Secure Hash Standard", Federal Information Processing Standards Publication 180-2, National Institute of Standards and Technology (NIST), August 2002, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[176] Chu-Hsing Lin, Yi-Shiung Yeh, and Chen-Yu Lee: "Keyed/Unkeyed SHA-2," Journal of Discrete Mathematical Sciences and Cryptography, Vol. 6, No. 1, pp. 45-58, April 2003

[177] Philip Hawkes , Michael Paddon , Gregory G. Rose, “On Corrective Patterns for the SHA-2 Family”, Cryptology eprint Archive, pp. 1-26, Qualcomm Australia, August 2004

[178] T. Dierks, E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, RFC 5246, Aug 2008, http://tools.ietf.org/html/rfc5246

[179] M. Salter, E. Rescorla, R. Housley, “Suite B Profile for Transport Layer Security (TLS)”, RFC 5430, March 2009, http://tools.ietf.org/html/rfc5430

[180] L. Law, “Suite B Cryptographic Suites for IPSec”, RFC 4869, Apr 2007, http://tools.ietf.org/html/rfc4869

[181] "Digital Signature Standard (DSS)", Federal Information Processing Standards Publication, FIPS PUB 186-3, June 2009, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[182] “Security in Vehicle Safety Communications - Applications (VSC-A)”, IEEE P1609 Meeting, University of Berkeley, June 16-18, 2009

[183] Varun Jannepally, Sohum Sohoni, “Fast Encryption and Authentication for Cache-to-Cache Transfers using GCM-AES”, International Conference on Sensors, Security, Software and Intelligent Systems, pp. 1-7, Coimbatore, India, Jan 2009

[184] D. McGrew, “Authenticated Encryption with AES-CBC and HMAC-SHA1 (and other generic combinations of ciphers and MACs), draft-mcgrew-aead-aes-cbchmac-sha1-01.txt”, Internet-Draft, March 9, 2009, http://tools.ietf.org/html/draft-mcgrew-aead-aes-cbc-hmac-sha1-01

[185] K.M. Igoe, J.A. Solinas, “AES Galois Counter Mode for the Secure Shell Transport Layer Protocol, draft-igoe-secsh-aes-gcm-02”, Internet-Draft, May 20, 2009, http://tools.ietf.org/html/draft-igoe-secsh-aes-gcm-02

[186] Zhou Gang, H. Michalik, L. Hinsenkamp, "Efficient and High-Throughput Implementations of AES-GCM on FPGAs", International Conference on Technology, 2007, (ICFPT 2007), page(s):185 – 192, December12-14, 2007

[187] T. Egemen, M. Aukar, "Design and System Implementation of a Crypto Processor for AES and DES Algorithms", Information Security and Cryptology Conference, Ankara, December 2007, http://www.iscturkey.org/2007/pdf/sozlu/20.pdf

[188] Reed Solomon Decoder IP Core, HITech Global Design and Distribution, LLC, Retrieved on July 14th, 2009, http://www.hitechglobal.com/ipcores/rsd.htm

[189] Neil Sholer, “Abacus: A candidate for sha-3”, WaveStrong, Inc., Version 1.0 Submission to NIST, October 2008

[190] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, D. Singer, "RTP Payload Format for H.264 Video", RFC 3984, http://www.rfc-editor.org/rfc/rfc3984.txt

[191] Zhihai He, Yong Kwan Kim, Mitra, S.K. "Object-level bit allocation and scalable rate control for MPEG-4 video coding", Proceedings of Workshop and Exhibition on MPEG-4, pp. 63 – 66, 2001

[192] Xingjun Zhang, Xiao-Hong Peng, Dajun Wu, "A Hierarchical Unequal Packet Loss Protection Scheme for Robust H.264/AVC Transmission", Proceedings of the 6th Annual IEEE Consumer Communications and Networking Conference (CCNC 2009) UASS workshop, pp. 1-5, Las Vegas, Nevada, USA, January 10-13, 2009

[193] Olivia Nemethova, Wolfgang Karner, Markus Rupp, "Error Prediction Based Redundancy Control for Robust Transmission of Video over Wireless Links", IEEE International Communications Conference (ICC '07), pp. 1803 – 1808, 2007

[194] Y. S. Kavian, A. Falahati, A. Khayatzadeh, M. Naderi, "High Speed Reed-Solomon Decoder with Pipeline Architecture", Second IFIP International Conference on Wireless and Optical Communications Networks, pp. 415-419, 2005

[195] Cristian Coarfa, Peter Druschel, Dan S. Wallach, "Performance Analysis of TLS Web Servers" ACM Transaction Computer Systems 24(1): 39-69, 2006

[196] C. Kaufman, Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005, http://www.ietf.org/rfc/rfc4306.txt

[197] D. McGrew, J. Viega, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543, May 2006, http://www.faqs.org/rfcs/rfc4543.html

[198] Gabriela Limon Garcia, "IPSec performance analysis for large-scale Radio Access Networks", Master of Science Thesis, Faculty of Information and Natural Sciences, Department of Computer Science and Engineering, Helsinki University of Technology, July 30, 2008

[199] P. V. Oorschot, D. Hankerson, A. Menezes, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, ISBN: 978-0849385230, Dec 16, 1996

[200] Francisco Rodriguez-Henriquez, Arturo Diaz Perez, Nazar Abbas Saqib and Cetin Kaya Koc, "A Brief Introduction to Modern Cryptography", Cryptographic

Algorithms on Reconfigurable Hardware, Signals and Communication Technology, Springer, ISBN: 978-0-387-36682-1, April 03, 2007

[201] Serge Vaudenay, "The Security of DSA and ECDSA - Bypassing the Standard Elliptic Curve Certification Scheme", EC Validation in ECDSA, Ecole Polytechnic, 2003, http://grouper.ieee.org/groups/1363/WorkingGroup/presentations/ecdsa_pkc03.pdf

[202] Daniel R. L. Brown, "The Exact Security of ECDSA", Certicom Research, 2000, http://eprints.kfupm.edu.sa/69865/1/69865.pdf

[203] Mihir Bellare, “Attacks on SHA-1”, University of California at San Diego, March 2005, http://www.openauthentication.org/pdfs/Attacks%20on%20SHA-1.pdf

[204] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin, "Efficient Collision Search Attacks on SHA-0", CRYPTO 2005, http://www.cs.cmu.edu/~dbrumley/srg/spring06/sha-0.pdf

[205] Rui TU, Jinshu SU, Ruoshan KONG, "An Identifier-Based Network Access Control Mechanism Based on Locator/Identifier Split", International Journal of Communications Network and System Sciences (IJCNS), pp. 641-644, 2009

[206] Y. Rebahi, J. J. Pallares, T. M. Nguyen, S. Ehlert, G. Kovacs, D. Sisalem, "Performance analysis of identity management in the Session Initiation Protocol (SIP)", IEEE ACS International Conference on Computer Systems and Applications (AICCSA '08), pp. 711-717, 2008

[207] Xiaoyun Wang and Hongbo Yu, “How to Break MD5 and Other Hash Functions”, Advances in Cryptology – Eurocrypt, pp. 19-35, 2005

[208] Michael Neve, Kris Tiri, "On the complexity of side-channel attacks on AES-256 - methodology and quantitative results on cache attacks", Cryptology ePrint Archive: Report 2007/318, IACR 2007

[209] Henrik Ahlstrom, Karl-Johan Skoglund, “Encryption in Delocalized Access Systems”, LITH-ISY-EX--07/4046-SE, Thesis, Linkoping 2007

[210] Darrel Hankerson, Alfred Menezes, Scott Vanstone, "Guide to Elliptic Curve Cryptography", ISBN: 978-0387952734, Springer, 2004

[211] Men Long, “Roaming Authentication and End-to-End Authentication in Wireless Security", Auburn University, Ph.D. Sissertation, 2005

[212] Anthony Nicholson, Mark D. Corner, Brian D. Noble, “IEEE Transactions on Mobile Computing”, Vol. 5, No. 11, pp. 1489-1502, November 2006

[213] Helena Rifa-Pous, Jordi Herrea-Joancomrti, "Cryptographic Energy Costs Are Assumable in Ad Hoc Networks", IEICE Transactions on Information and Systems, 92-D, No. 5, pp. 1194-1196, Classification 69, 2009

[214] I.Y. Jung, I.S. Cho, H.Y. Yeom, "A Cost-Effective Guarantee of Security and Scalability on HVEM DataGrid with Active Disk", the Annual 32nd IEEE International Computer Software and Applications (COMPSAC '08), pp. 409-416, 2008

[215] Daniel Moreno Rossello, “Cryptographic System for Supply Chains over RFID Implementation”, Master of Science Thesis, Department of Telecommunication Engineering and Management, Polytechnic University of Catalunia, 2008

[216] Creighton T. R. Hager, Scott F. Midkiff, Jung-Min Park, Thomas L. Martin, "Performance and Energy Efficiency of Block Ciphers in Personal Digital Assistants", PerCom 2005: 127-136

[217] “ARM7TDMI: ARM 32-bit RISC core at 16-bit system costs”, Retrieved on November 18th, 2009, http://www.arm.com/products/CPUs/ARM7TDMI.html

[218] “ARM7 Family: ARM7TDMI, ARM7TDMI-S, ARM7EJ-S, and ARM720T”, http://www.arm.com/pdfs/ARM7_thumb_flyer_35_4.pdf

[219] Jaroslav Ban, "Cryptographic library for ARM7TDMI processors", Master of science thesis, Technical University Kosice, 2007

[220] Serge Vaudenay, "The Security of DSA and ECDSA - Bypassing the Standard Elliptic Curve Certification Scheme", EC Validation in ECDSA, Ecole Polytechnic, 2003, http://grouper.ieee.org/groups/1363/WorkingGroup/presentations/ecdsa_pkc03.pdf

[221] Zhi Guan, Zhen Cao, Xuan Zhao, Zhong Chen, Xianghao Nan, "WebIBC: Identity Based Cryptography for Client Side Security in Web Applications", the proceedings of 28th International Conference on Distributed Computing Systems, Washington DC, USA, pp: 689-696, June 17-20, 2008

[222] Y. Zhu, J. E. Rice, "A Lightweight Architecture for Secure Two-Party Mobile Payment", proceedings of the International Conference on Computational Science and Engineering 2009, pp: 326-333, 2009

[223] Comparison of Mobile Processors (CPU Benchmarks), Retrieved on July 8th, 2010, http://www.notebookcheck.net/Mobile-Processors-Benchmarklist.2436.0.html

[224] Abdul Kalam Kunnel Aboobaker, "Performance Analysis of Authentication Protocols in Vehicular Ad Hoc Networks (VANET)", Master of Science Thesis, Department of Mathematics, University of London, September 2nd, 2009

[225] Murat Kantarcioglu, "Digital Signatures", The University of Texas at Dallas, April 17th 2008, http://www.utdallas.edu/~muratk/courses/crypto07_files/ds.pdf

[226] Jörg Wallerich, Holger Dreger, Anja Feldmann, Balachander Krishnamurthy, Walter Willinger. A methodology for studying persistency aspects of internet flows. SIGCOMM Computer Communications Review (CCR), 35(2):23-36, 2005

[227] Jörg Wallerich, "Capturing the Variability of Internet Flows in a Workload Generator for Network Simulators", Ph.D. Dissertation, Faculty of Computer Science, University of Munich, 2007

[228] Andre Broido, Young Hyun, Ruomei Gao, Kc Claffy, "Their share: diversity and disparity in IP traffic", the 5th annual Passive and Active Measurement Workshop (PAM '04), LNCS3015, pp. 113-125, 2004

[229] Wentian Li, "References on zipf's law", Retrieved on May 5, 2009, http://linkage.rockefeller.edu/wli/zipf/

[230] Konstantina Papagiannaki, Nina Taft, Christophe Diot, "Impact of Flow Dynamics on Traffic Engineering Design Principles", INFOCOM 2004, 7-11 March 2004, Vol. 4, Page(s): 2295- 2306

[231] J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, F.D. Smith, and M.C. Weigle, "Stochastic Models for Generating Synthetic HTTP Source Traffic", pp. 1547-1558, IEEE INFOCOM, Hong Kong, March 2004

[232] Hongyue Zhu, "Optical WDM Networks: Traffic Grooming in Mesh Networks and Metro Networks using ROADMs", Ph.D. Dissertation, Department of Computer Science, University of California, Davis, September 2005

[233] J. P. Dubois, “Burstiness Reduction of a Doubly Stochastic AR-Modeled Uniform Activity VBR Video”, Proceeding of WASET, 23: 454-458, 2007

[234] Kun-Chan Lan, John Heidemann, “A measurement study of correlations of Internet flow characteristics”, USC Information Sciences Institute, ISI-TR-574, 2006

[235] Yin Zhang, Lee Breslau, Vern Paxson, Scott Shenker, “On the characteristics and origins of internet flow rates”, Proceedings of ACM SIGCOMM, pp. 309-322, Pittsburgh, PA, USA, August 2002

[236] Naimul Basher, Aniket Mahanti, Anirban Mahanti, Carey Williamson, and Martin Arlitt, “A Comparative Analysis of Web and P2P Traffic”, pp 287-296, Proceedings of the 17th International World Wide Web Conference (WWW2008), Beijing, China, April 21-25, 2008

[237] Abuagla Babiker Mohammed, Sulaiman Mohd Nor, "Near Real Time Online Flow-based Internet Traffic Classification Using Machine Learning", Computer Science Journals (CSC), pages: 370-379, 2009

[238] Yin Zhang, Lee Breslau, Vern Paxson, and Scott Shenker, “On the characteristics and origins of internet flow rates”, Proceedings of ACM SIGCOMM, pp. 309-322, Pittsburgh, PA, USA, August 2002

[239] Johannes Krohn, “VoIP Standards and Status”, 2005, http://www.ite.fh-wiesbaden.de/~hofmannk/Vortrag/Vortragsfolien/20051118-Krohn-voip.pdf

[240] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, M. K. Reiter, "On Web Browsing Privacy in Anonymized NetFlows Export", Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium (SS'07), pp. 1-14, 2007

[241] A Spenst, T Herfet, J. Miroll, "An Implementation of the User-Centric QoS Management Approach in Wireless Home Networks", IEEE International Symposium on Wireless Communication Systems (ICWCS ‘07), pp. 107-112, Trondheim, Norway, October 16th-19th, 2007

[242] Yonatan Levy, F. Huebner, D. Liu, “A Hierarchical Multi-Class Traffic Model for Data Networks”, Proceedings of ITC-16, pp. 1221-1229, Edinburgh, Scotland June, 1999

[243] Kenny Qing Shao, "Traffic measurement in hybrid satellite-terrestrial network", Communication Network Laboratory, School of Engineering Sciences, Simon Fraser University, Retrieved on May 6th, 2009, http://www.ensc.sfu.ca/~ljilja/cnl/presentations/kenny/traffic_satellite/index.htm

[244] E D Puschita, T P Palade, "QoS Perspective for Wireless Scenarios", Broadband Europe 2005, Paper #: W01A.02, Bordeaux, 12-14 December 2005

[245] Cristina Aurrecoechea, Andrew T. Campbell and Linda Hauw, "A Survey of QoS Architectures", Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6, No. 3, pp.138-151, May 1998

[246] F. Guo, "Traffic Analysis: from Stateful Firewall to Network Intrusion Detection System", RPE Report, January 2004

[247] Time Szigeti, Christina Hattingh, "End-to-End QoS Network Design: Quality of Service in LAN's WANs, and VPNs", Cisco System Press, November 2004

[248] Generic QoS API, Windows 2000 Resources Kit, Microsoft Windows, Retrieved on May 20, 2009, http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/cnet/cndc_qos_hrgn.mspx?mfr=true

[249] QoS API, Application-Driven QoS Components, Microsoft Windows MSDN, Retrieved on May 20, 2009, http://msdn.microsoft.com/en-us/library/aa374050(VS.85).aspx

[250] George Porter, "Improving Distributed Application Reliability with End-to-End Datapath Tracing", University of California, Berkeley, Technical Report No. UCB/EECS-2008-68, May 22, 2008