QoS^2: A Framework for Integrating Quality of Security with Quality of Service

Tarik Taleb *Senior Member, IEEE* and Yassine Hadjadj-Aoul

***Abstract*—Different security measures have emerged to encounter various Internet security threats, ensuring a certain level of protection against them. However, this does not come without a price. Indeed, there is a general agreement that high security measures involve high amount of resources, ultimately impacting the perceived Quality of Service (QoS).**

The objective of this paper is to define a framework, dubbed QoS^2, that provides means to find a tradeoff between security requirements and their QoS counterparts. The QoS^2 framework is based on the multi-attribute decision making theory. The performance of the QoS^2 framework is evaluated through computer simulations. A use-case considering worm email detection is used in the performance evaluation.

I. Introduction

Providing services with high perceived QoS is known to be frequently antagonistic with the provision of highly secure services. Indeed, security mechanisms clearly involve extra resources [3], which may impact the perceived QoS or even degrade the overall system performance.

This observation has been also confirmed by some of our previous research work, pertaining to detection of Internet worms in large scale networks [4], and detection and trace back of sophisticated attacks using encrypted protocols [5]. The work in [4] indicates that the longer the generated signature is, the more accurate the detection is. However, this also increases the end-to-end delay and impacts the overall Quality of Experience (QoE) of users. In [5], an Intrusion Detection System (IDS) capable of detecting attacks using cryptographic protocols was devised. The devised IDS uses strategically distributed Monitoring Stubs (MSs) that sniff the encrypted traffic, extract features for detecting these attacks and construct normal usage behavior profiles. Upon detecting suspicious activities, the MSs notify the victim servers, which may then take necessary actions. Depending on the detected attack, such actions may introduce additional delays to the end-to-end delay to disable attackers from making accurate estimates of the processing time required for the decryption of a particular key (e.g., remote time attack [6]). Such actions may also involve a random discarding of packets (e.g., password attack). All in all, existing network security measures may have a side effect on the overall QoS of the system. It is thus imperative to deploy security requirements along with their QoS counterparts.

T. Taleb is with NEC Europe LTD, Heidelberg, Germany. Email: talebtarik@ieee.org.

Yassine Hadjadj-Aoul is with the University of Rennes, France. Email: yassine.hadjadj-aoul@irisa.fr

Many researchers have, recently, started using the terms "Quality of Protection", when addressing the issue of integrating QoS with security, whereby sensitive information is protected using adequate authentication and cryptographic algorithms to ultimately ensure QoS [7]. For example, the work in [8] introduces a middleware adaptation scheme that dynamically tunes the encryption key length of the underlying encryption algorithm to the actual end-to-end delay. The major drawback of this work consists in its vulnerability to attacks such as man-in-the-middle [9]; in other words the work ensures a level of QoS but this comes at the price of some security flaws.

In this paper, our main concern is to design a novel framework, entitled QoS^2 (i.e., Quality of Service and Security), that enables network protection from malicious usage and attacks.However, in the absence of a potential threat, the QoS^2 framework, in an autonomic way, relaxes the system's overall security requirements in case the required QoS are not met under the current security settings. The framework provides an adjustable level of security to ensure acceptable QoS employing a Multi Attribute Decision Model (MADM) approach. An abridged version of this work has appeared in [1].

The remainder of this paper is organized as follows. Section II provides an overview of the state of the art. Section III describe inn details the envisioned QoS^2 framework. Section IV portrays the simulation setup and discusses the obtained results. The paper concludes in Section V with a summary recapping the main advantages and achievements of the proposed frameworks.

II. Related work

In [2], Shen *et. al.* claimed that little work has been done on the interaction between QoS and Security in networks. They noticed that while QoS and Security used to be treated as separated entities, they strongly impact each other and thus should be considered together when designing protocols for Mobile Ad-Hoc Networks (MANETs). For this reason, they proposed a distributed dynamic management system that should keep QoS and Security as good as possible in MANETs even if the available network resources change. In [3], Irvine *et. al.* suggest a Quality of Security Service (QoSS) theory that handles security as a dimension of QoS. A brief description of the proposed framework is given illustrating the use of variant security mechanisms and policies that should allow distributed Resource Management Systems (RMSs) to support the user requirements for both performance and security.

To enable service providers to advertise Security of Service (SoS) to their clients, researchers investigated the incorporation of security parameters into the Service Level Specifications (SLS) [10], [11]. The selected security parameters are integrated to enhance SLA-based management of QoS with the generation of network policies that guarantee the reservation of adequate resources for meeting both security and QoS needs. In [11], it is found that the integrated security protocol impacts the resources during the initialization phase in which the security context is established (i.e., during key generation, negotiation of the used algorithms, and so forth). This consumes the processing power of the end hosts, memory, and also adds to the end-to-end delay. The specific protocol data (IPSEC/TLS header) also consumes higher bandwidth. In order to determine the precise impact of the security protocol on the network bandwidth, different security levels applied on IPSec revealed the bandwidth costs for a MPEG video and a DVD sequence. It is found that while the multimedia sequence quality, the confidentiality level, and the authentication and integrity level do not impact much the bandwidth cost, the choice of the security services and the protocol do.

To ensure both QoS and security requirements, some researchers addressed the problem using an adequate adaptive theory. For instance, the work in [12] exploits the co-operative game theory-based strategies to model the interaction between intruders and IDSs in MANETs and wired networks. On the other hand, Bayesian Nash algorithm is employed in [13] to analyze the interaction between an intruder and a defender in both static and dynamic network settings with the aid of monitoring systems. Nash equilibrium-based game theoretic studies have also been conducted towards solving QoS problems (i.e., without any security incorporation) involving power and rate control problem where network users compete with each other to obtain maximum throughput with minimum energy consumption [14]. While these work, based on game theory and probabilistic models, may be able to formulate the problem of intrusion detection in a general case, they are not realistic for addressing the intricate interaction of multi-level QoS and security requirements. In order to formulate the QoS-security problem, we resort to the use of a distributed and elegant algorithm obtained from the Multi Attribute Decision Making (MADM) theory. As defined in [18], MADM means "making preference decisions (e.g., evaluation, prioritization, selection) over the available alternatives that are characterized by multiple, usually conflicting attributes". MADM approach can be applied using different algorithms. Some of the commonly used ones are the Simple Additive Weighting (SAW) and Minimal Distance to Utopia Point (MDUP) algorithms. Authors in [17] conclude their performance evaluation that the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) algorithm outperforms both the SAW and MDUP algorithms. The advantages offered by MADM approach are more evident with the implementation of TOPSIS which we use in this paper.

III. Envisioned Solution

Before delving into details about possible security-QoS integration, it is important to investigate the security challenges and their counter-measures. To this end, in the European research group ResumeNet [15][1], a taxonomy is developed to systematically document and assess the impact of various challenges, which pose a threat to the system. This taxonomy first identifies the challenge categories that reflect the nature of the challenge. A second-level classification is then formulated based on the specific scenario to which this challenge applies. The work then addresses the need to formulate defensive mechanisms for each of the challenges in different scenarios, and to also find the appropriate defensive measures. By performing rigorous system analysis and understanding the challenges that lead to the high likelihood of systems failures, in addition to learning from past events, in the ResumeNet project a library of the best possible defensive mechanisms specific to the given challenge and the prevailing scenario or set-up is built. Whilst the purpose of our current work is not to explicitly deal with these defensive mechanisms, we intend employing feedback from them in our QoS^2 framework for integrating security with QoS attributes.

In this paper, we envision a "network security advisory system" with a number of threat levels ranging from low to severe. The security advisory system defines the threat level of the network based on events reported by entities such as firewalls and IDSs. It analyzes the events in specific timeslots and constantly updates the threat level. For each threat level and each associated security level, a particular defensive measure can be applied, following the taxonomy developed in [15].

Based on the alert level indicated by the security advisory system, we are interested in devising a security/QoS policy control that indicates the security level that should correspond to a desired QoS level. When under the indicated threat level, the security advisory system recommends a range of security levels (e.g., range of encryption/decryption key lengths, anomaly detection score, worm signature lengths, etc), we are interested in finding out the highest security level that should be selected in a way that the QoS requirements of users are not compromised. If for a particular security level, from within the recommended range, the QoS requirements of users cannot be satisfied (i.e., this can be inferred from a learning phase), the security/QoS policy control unit asks for security relaxation. In contrast, if the network is under potential attack and the security advisory system recommends the highest security level, the system has to stick to the recommended level although this decision may compromise the required QoS. QoS relaxation (e.g., transmission rate adaptation), thus, becomes mandatory in such a case.

Note that the proposed framework can be integrated into Service-Oriented Architectures (SOA), as either: (i) a module, which helps to decide the best suited quality, into the SOA layer, or (ii) a web service into the implementation layer. Particularly, it can clearly enriches the framework, provided in [23], which allows to integrate dependability and security concepts in service-oriented architectures. Indeed, this can permits to find-out the tradeoff existing between security requirements and their QoS counterparts provided by the users,

[1]The present research work is part of the ResumeNet project.

as it is not provided by the framework proposed in [23].

A. Envisioned architecture

Fig. 1 depicts an example deployment scenario, portraying a carrier transport network administrated by a particular Network Operator (NO) and connecting a number of content/service providers (i.e., clients of the network operator) to their customers, located in different local networks. The corresponding network architecture along with the envisioned interfaces between the different network components is illustrated in Fig. 2. The architecture is divided into two levels, namely the network and the service levels, which are in turn split into the control and data planes. The overall network topology comprises a number of Monitoring Stubs (MSs), which are intelligently deployed over the core network next to strategic routers. These MSs form a hierarchical threat detection system, consisting of two layers, namely Local Security Monitors (LSMs) and Metropolitan Security Monitors (MSM). The LSMs gather information about behavioral anomaly of a particular local network and deliver them to the respective MSMs (i.e., using interface $F1$ in Fig. 2). Upon receiving information about a suspicious event from one of its LSMs, a MSM filters through its database of past attacks and evaluates if the threat is real. If the suspicious event is matched with one of the previous threats, the MSM notifies the security advisory (i.e., via interface $A2$). The security advisory contains a library of existing attacks and their counter-measures (i.e., following the taxonomy developed in [15]). The security advisory verifies if a similar threat originated from any other local networks administrated by another MSM (e.g., in case of rapid worm spread or DDoS traffic). The security advisory also consults its library of counter-measures (i.e., built following the taxonomy developed in [15]) to find the most appropriate mechanism to combat against the arising challenge. Then, the security advisory defines the threat level and relays the information pertaining to the threat detection and counter mechanism to all MSMs (i.e., via $A2$ interface), which in turn, forward the information to all collaborating LSMs (i.e., via $F1$ interface). When security measures are to be partially or fully enforced at the network elements (e.g. routers), the security advisory instructions are communicated by LSMs and MSMs to the respective network elements via the $I1$ and $I2$ interfaces. The security advisory also notifies the service managers of the different service providers (i.e., via $A1$ interface) of details on the on-going threat and triggers them to take adequate measures to adapt their QoS demands to the new network dynamics. Instructions on QoS adaptation/relaxation is communicated either to servers using $SM2$ interface, to end users using $SM1$ interface, or to both when required.

B. MADM-based QoS-security level decision model

The QoS^2 system aims to apply a security level under a certain attack while maintaining a good level of QoS. QoS and security requirements are known to be conflicting in general, hence the merits behind employing a multi-attribute decision making (MADM) algorithm for the determination of security level vs. QoS level. QoS^2 relies on the advisory system that

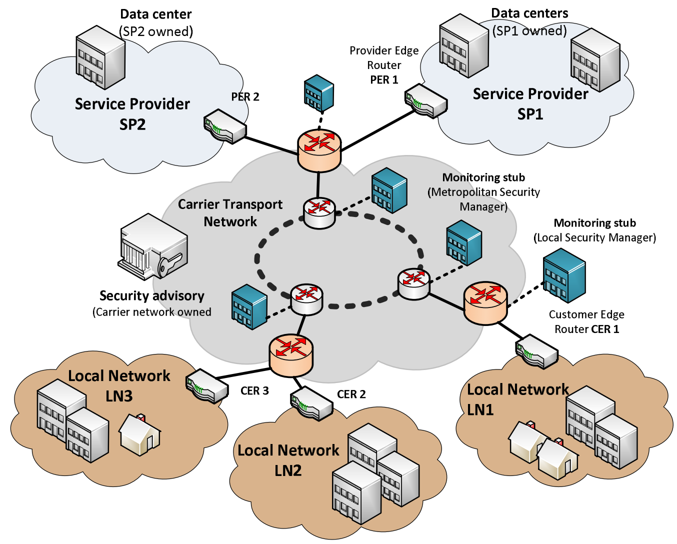

Fig. 1. An example deployment scenario.

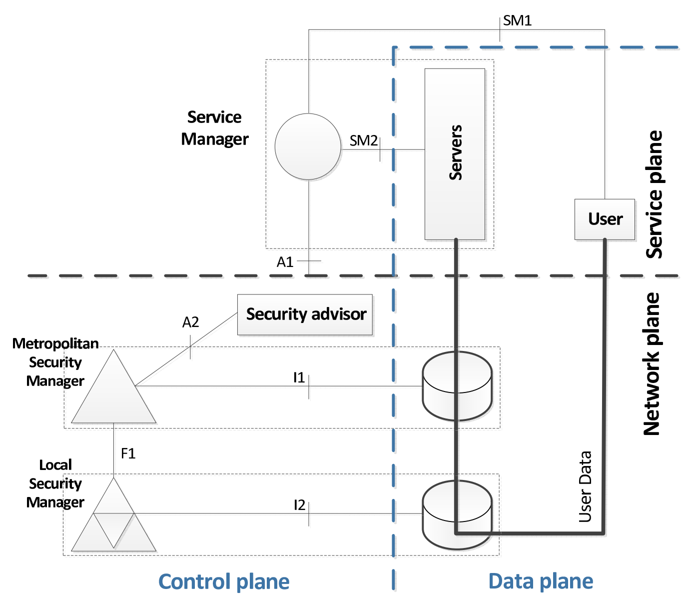

Fig. 2. Network architecture of the envisioned QoS^2 framework.

recommends for a given threat level, a range of security levels that could counter the attack. Then the system tries to choose the highest security level that keeps QoS requirements as good as possible.

Let SM and PM denote the group of QoS metrics (e.g., bandwidth, delay, packet loss rate, etc) and the group of security metrics (e.g., encryption/decryption key length, timeliness, etc)[2], respectively. Let N_s and N_p denote the number of the QoS and security metrics, respectively. We assume that a user's Service Level Agreement (SLA) has S levels of QoS and P

[2]The work in [16] provides an extensive list of important security and QoS metrics.

levels of security. When a user[3] connects to the network, the system advisory of QoS[2] must serve the user with a security level in such a manner to maintain good QoS requirements (i.e., QoS level). The suggested procedure is as follows.

1) Step 1: Defining all possible QoS level and security level combinations: For each QoS level, there are some requirements related to the values of its metrics that should be respected. These values should not be beyond (resp., below) a predefined threshold for cost metrics (resp., for benefit metrics). For example, for a given QoS level, the bandwidth (BW) should exceed a threshold (e.g., $BW_{th} \leq BW$) and the delay should be less than a threshold (i.e., $D \leq D_{th}$).

On the other hand, when we apply a security level, the values of the QoS metrics get negatively influenced in general. This means that these values cannot exceed some thresholds that we can measure by experiments. Consequently, some combinations will be impossible. Indeed, let QoS_{sup} denotes the highest QoS level and Sec_{sup} denotes the highest security level. Let's suppose that for QoS_{sup}, the bandwidth (BW) is ($BW_{high} \leq BW$) and that for Sec_{sup}, the bandwidth will be such as $BW < BW_{high}$. Thus, the combination (Sec_{sup}, QoS_{sup}) is impossible. Hence, the possible combinations (alternatives) of a QoS level and a security level will be limited. Let J denote the number of these alternatives ($J < N_s \cdot N_p$). For each QoS level $s \in \{1, ..., S\}$, the value of the QoS metric sm ($sm \in SM$) would be such as $sm_s \in [(sm_s)_{min}, (sm_s)_{max}]$ (i.e., sm_s: the value of the metric sm as defined in the QoS level s). On the other hand, for a security level $p \in \{1, ..., P\}$, the value of the QoS metric sm (sm_p) will be such as $sm_p \in [(sm_p)_{min}, (sm_p)_{max}]$.

As for the values of security metrics, they may have exact values (e.g., encryption key length) or may belong to an interval (e.g., timeliness) like all QoS metrics. The values of security metrics do not intervene in the definition of the alternatives. Now the selection of a combination of a QoS level ($s \in \{1, ..., S\}$) and a QoP level ($p \in \{1, ..., P\}$) will not be possible unless the following condition is satisfied:

$$\{[(sm_s)_{min}, (sm_s)_{max}] \cap [(sm_p)_{min}, (sm_p)_{max}] \neq \Phi\} \quad (1)$$

In the remainder of this paper, an alternative (a combination) is denoted as j (i.e., $j \in \{1, ..., J\}$).

2) Step 2: Defining the Decision Matrix (DM) for a user's connection c: Let $X^c_{jk}(t)$ be the value of metric k measured at time instant t for a connection c when the QoS-security combination j is used (i.e., $k \in \{1, ..., K\}$ being the index of the QoS-security metric). As we deal with different kinds of metrics (e.g., some expressed in kbps, others in seconds, etc), we normalize their values to be able to compare among them. This is done as follows:

$$\hat{X}^c_{jk}(t) = \frac{X^c_{jk}(t)}{max(X^c_{jk}(t))} \quad (2)$$

[3]In this paper, the term "user" has a wider scope as it refers to a client of the network operator (e.g., content/service provider). Additionally, a connection does not necessarily refer to an end-to-end connection between a server and a client, but it does rather refer to the communication path between a content/service provider and a local network where some of its subscribers reside (Fig. 1).

Using the following notation ($K = N_s + N_p$), the normalized vector for a given QoS-QoP level j at a time instant t is:

$$\left(\hat{X}^c_{j1}, ...\hat{X}^c_{jk}, ...\hat{X}^c_{jK}\right) \quad (3)$$

Depending on the provided service and the user requirements, the QoS-Security metrics may not have the same importance. Thus, we assign each attribute a weight such as the sum of all the weights is equal to one. Let $Y^c_{jk}(t)$ be the normalized value of the metric k multiplied by its relative weight $w_{c.k}$. We obtain the Decision Matrix for a given user's connection c at a time instant t as follows:

$$DM^c = \begin{bmatrix} Y^c_{11} & ... & Y^c_{1k} & ... & Y^c_{1K} \\ ... & ... & ... & ... & ... \\ Y^1_{j1} & ... & Y^c_{jk} & ... & Y^c_{jK} \\ ... & ... & ... & ... & ... \\ Y^1_{J1} & ... & Y^c_{Jk} & ... & Y^c_{JK} \end{bmatrix} \quad (4)$$

It should be noted that a decision D taken at a time instant t remains valid for a period ΔT^c_D during which the system gathers information from monitoring stubs and the collaborating networks.

3) Step 3: Applying a MADM algorithm to find the best alternatives among the available ones: In order to find the ideal alternatives or utopia points, we need to account for two types of QoS[2] parameters, namely "cost" metrics (e.g., bandwidth, delay, security level) and "benefit" metrics (e.g., throughput and fairness). The objective of formulating an utopian vector is to maximize the benefit while minimizing the cost as much as possible. In other words, the utopia vector should permit selection of the best value for each single QoS[2] attribute amongst all the alternatives. This is characterized by the utopia vector of attributes at time t, defined in Eq. 5.

$$\left({}^{utopia}Y^c_1, ...{}^{utopia}Y^c_k, ...{}^{utopia}Y^c_K\right) \quad (5)$$

where each component of the utopia vector may be obtained using the following expression.

$${}^{utopia}Y^c_{jk} = \begin{cases} Y^c_{jk} : j = \underset{j \in \{1,J\}}{\text{argmin}} Y^c_{jk}, \ for\ cost\ metrics \\ Y^c_{jk} : j = \underset{j \in \{1,J\}}{\text{argmax}} Y^c_{jk}, \ for\ benefit\ metrics \end{cases} \quad (6)$$

In a reverse way, it is also possible to obtain the knowledge pertaining to the worst alternatives or nadir points in DM^c. The nadir point may be computed as follows.

$$\left({}^{nadir}Y^c_1, ...{}^{nadir}Y^c_k, ...{}^{nadir}Y^c_K\right) \quad (7)$$

In this case, each component of the nadir vector holds similar considerations as those for Eq. 6 and is given by:

$${}^{nadir}Y^c_{jk} = \begin{cases} Y^c_{jk} : j = \underset{j \in \{1,J\}}{\text{argmax}} Y^c_{jk}, \ for\ cost\ metrics \\ Y^c_{jk} : j = \underset{j \in \{1,J\}}{\text{argmin}} Y^c_{jk}, \ for\ benefit\ metrics \end{cases} \quad (8)$$

The TOPSIS selection algorithm [17] is derived from the MADM theory to extend these two contrasting utopia and nadir points to exploit the knowledge of both. We employ the TOPSIS selection algorithm for choosing the appropriate security level for ensuring the appropriate QoS requirements. To this end, the TOPSIS algorithm selects the optimum TOPSIS vector from J alternatives, by minimizing the similarity to positive-ideal solution as follows.

$$j^{c,optimum}_{TOPSIS}(t) = \underset{j\in\{1,J\}}{\operatorname{argmax}} \left(\frac{S_j^{ng}}{S_j^{ps} + S_j^{ng}} \right) \tag{9}$$

where S_j^{ps} and S_j^{ng} denote the Positive and Negative separations, respectively. The former implies the Euclidean distance between the alternatives and the utopia point while the latter denotes that between the alternatives and the nadir point. S_j^{ps} and S_j^{ng} are expressed by Eqs. 10 and 11, respectively.

$$S_j^{ps} = \sqrt{\sum_{k=1}^{K} (Y_{jk}^c - {}^{utopia}Y_k^c)^2} \tag{10}$$

$$S_j^{ng} = \sqrt{\sum_{k=1}^{K} (Y_{jk}^c - {}^{nadir}Y_k^c)^2} \tag{11}$$

IV. Performance Evaluation

Having described details on our MADM-based QoS^2 approach, we now direct our focus to its performance evaluation using the Network Simulator (NS3) [19]. Given their strict QoS requirements, we consider IPTV streaming services. As a network threat, we envision the spread of Internet worms in a number of local networks whereby a number of subscribers to the IPTV service are located. As a counter measure, we adopt a signature-based worm detection approach as in our previous research work [4]. Along with a vast spread of the Internet worm, the security advisory recommends to MSMs and LSMs the filtering of inbound and outbound traffic using a generated signature with a particular length. The longer the signature length is, the longer the filtering-due delay becomes [4]. This intuitively impacts the end-to-end delay between the content servers and the end-clients (Fig. 1). At the security advisory, six threat levels are defined to the above mentioned security counter measure; each threat level is characterized by $i)$ a range of worm signature substring length L_{worm} (i.e., principally responsible for additional delays at routers collocated with LSMs and MSMs) and $ii)$ a minimum number of signature substrings S_{min} that should exist in a traffic flow to generate an alarm (Table I).

TABLE I
Worm spread threat levels and corresponding worm signature generation/detection parameters.

Threat level	L_{worm} range (Bytes)	S_{min}
1	0 - 40	11
2	40 - 73	9
3	73 - 115	7
4	115 - 166	5
5	166 - 226	3
6	226 - 295	1

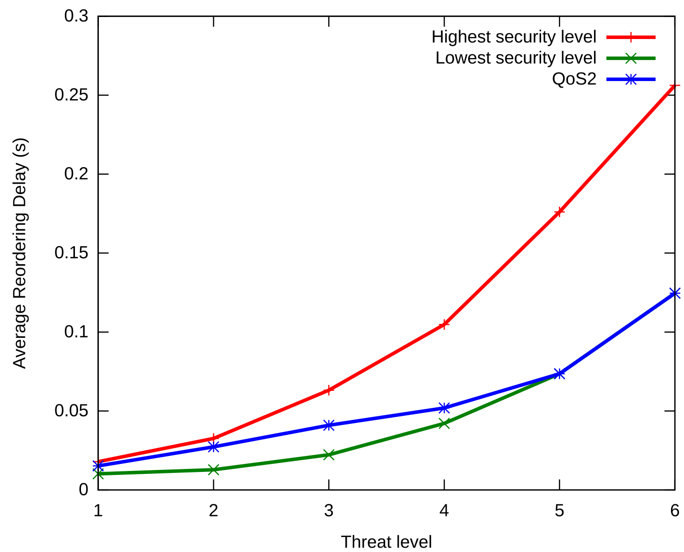

Fig. 3. Reordering delay for different threat levels.

In the simulations, we consider a network topology similar to that of Fig. 1 with video data streamed from a single content provider (i.e., Service Provider 1) to N_u subscribers located in local network $LN3$. To avoid multicast scenarios, each subscriber is receiving a different video content over a dedicated session. At the content provider side, different video traces, encoded in MPEG-4, are used [20] and servers use User Datagram Protocol (UDP) to provision IPTV service following the framework described in [21]. To simulate network dynamics, we input some worm-affected background traffic along the path between Customer Edge Routers (CERs) 1 and 2 simulating different Variable Bit Rate (VBR) UDP flows. The sending rate of each UDP flow is varying during the course of the simulation and is randomly chosen every $1s$ in a way that Core Network Edge routers (i.e., the brown routers in Fig. 1) are operating at loads exceeding 70% their full capacity. In the simulations, a noticeable increase or decrease in the background traffic rate triggers the security advisory to increase or decrease the threat level. With no specific purpose in mind, the aggregate propagation delays of links between Provider Edge Router (PER1) and CER3 is set to $15ms$. Without any loss of generality, all links are given a capacity equal to $50Mbps$ (i.e., customer/provider edge links as well as core network links). In order to remove limitations due to small buffer sizes on the network congestion, buffers equal to the bandwidth-delay product of the end-to-end link are used [22]. Due mostly to its simplicity and its wide usage in today's switches and routers, all simulated routers use Drop-Tail as their packet-discarding policy. The data packet size is fixed to $1336B$. The client side has a limited playback buffer length B, set to $100pkts$. The IPTV streams are characterized by an average streaming rate denoted as R_p and equal to $100pkts/s$ (i.e., 0.8Mbps). Simulations were all run for 900s; a duration long enough to ensure that the system has reached a consistent behavior. The presented results are averaged over the simulated N_u $(= 15)$ subscribers, averaged again on the total simulation runs (i.e., 36 times).

As shown in Table I, for each threat level the security advisory recommends a range of parameters for signature generation, subsequent traffic filtering and worm detection. In the envisioned QoS^2 approach, the TOPSIS algorithm is run to select the best set of parameters (i.e., from within the recommended range) to meet both the QoS and security requirements. As comparison terms, we compare the performance of the QoS^2 approach against that of two conventional methods whereby the highest (i.e., longest) and the lowest (i.e., shortest) security levels (i.e., signature substring length) are selected. As parameters to quantify the users' perceived QoS, we consider the following metrics:

- Packet reordering delay: Difference between the arrival time of a packet and the arrival time of its preceding one. This metric is important for real time multimedia streaming services (e.g., IPTV) as if it exceeds the end-user's playback delay (i.e., $\frac{B}{R_p}$), the corresponding packet will be simply discarded before being transmitted to the application layer. The user shall then notice ruptures in the streaming service, a fact that impacts the perceived video quality.
- Playback ratio: Defined as ratio of the playback rate to the average streaming rate R_p. The playback rate is computed as the number of packets that were transmitted to the application layer every monitoring period of time (e.g., playback buffer delay $\frac{B}{R_p}$) and were indeed displayed over the monitoring period of time.
- Queue occupancy: The number of packets residing in the client's buffer and awaiting reordering before their transmission to the application layer. This metric is measured every monitoring period of time (i.e., $\frac{B}{R_p}$). To achieve acceptable perceived QoS, this metric should not exceed the client's buffer size (i.e., overflow) and should not be in the vicinity of zero (i.e., underflow). Indeed, keeping a moderate value of this metric is highly important as it ensures for the application layer that there are always enough packets to display without having to discard them at the queue due to overflow.

Fig. 3 plots the average reordering delay experienced by the end-terminals for different threat levels and that is when the three security approaches (i.e., QoS^2 approach, and the security approaches applying the highest and the lowest security levels) are in use. Whilst the average reordering delay remains largely lower than the playback delay (i.e., $\frac{B}{R_p}$) for the three simulated approaches, in the simulations there were some instants when the value of this reordering delay exceeded the playback delay, and that is particularly when the highest security level is adopted in the event of high threat levels. The corresponding packet got intuitively discarded and this obviously would have been noticeable at the terminal's display. This will be manifested in poor playback ratio as indicated in Fig. 5. It should be noted that whilst the envisioned QoS^2 approach and the lowest security level exhibit relatively similar performance in Fig. 3, their main differentiator consists intuitively in their adopted security levels.

Figs. 4 and 5 show the impact of the three security approaches on QoS in terms of two correlating metrics, namely

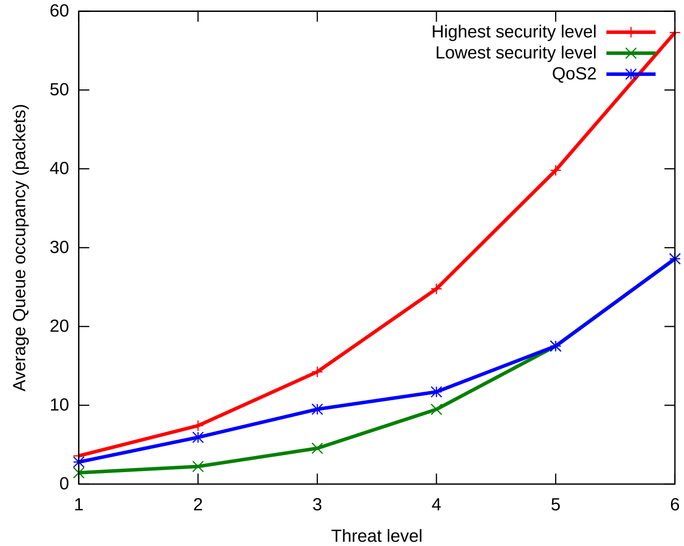

Fig. 4. Buffer occupancy for different threat levels.

average buffer occupancy and playback rate, respectively. The two figures indicate an obvious observation: the best QoS is achieved when the lowest security level is adopted. Indeed, the lowest security measures induce only few reordering, which directly impacts the queue occupation length (i.e. the latency before transmission to the application playout buffer). However, this may not be acceptable from the security point of view. When the security measures are tight using parameters corresponding to the highest security level, the performance degrades significantly as there is not sufficient data at the terminal's playout buffer to play and consequently the playback rate is remarkably poor. In contract to these two approaches, the proposed QoS^2 approach ensures an acceptable level of security and simultaneously achieves a QoS performance more or less similar to that obtained when the lowest security level is adopted.

The above simulation results show how the QoS^2 system jointly addresses the conflicting QoS and security requirements and demonstrate that adaptation of the security level according to QoS requirements yield satisfying results. However, it should be recalled that for a significantly high threat level, the advisory system may recommend significantly high security levels even if the QoS requirement are not met. In such events, QoS relaxation at servers and/or end-terminals becomes the only rescue.

V. Conclusion

In this study, we addressed the problems of providing high perceived QoS, while provisioning security requirements. We clearly demonstrated the need of addressing jointly these antagonistic problems. Thus, we devised a MADM-based network policy framework, named QoS^2, which provides a mean to find the tradeoff between security requirements and their QoS counterparts. The proposed framework is intended to be used at the global security advisory system, which select the most suitable security conditions based on real-time feedbacks from different monitoring systems deployed over the networks.

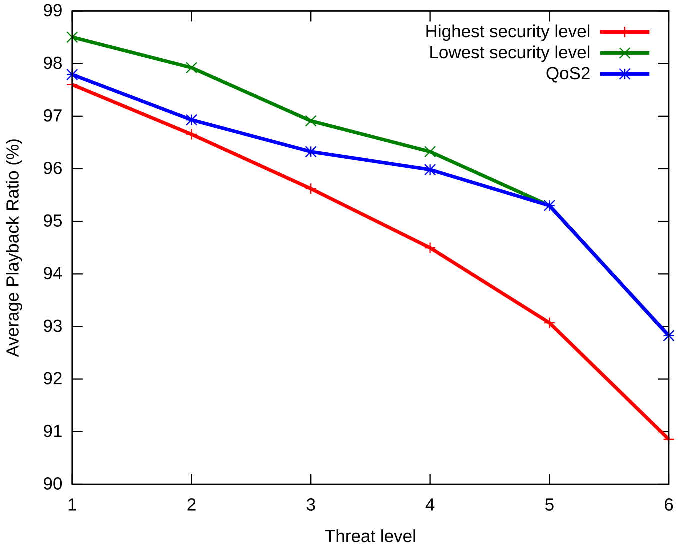

Fig. 5. Playback rate for different threat levels.

Extensive simulations considering the IPTV services showed that our envisioned QoS^2 framework achieves its design goals, as it provides simultaneously guarantees in terms of terms of perceived QoS and security. In the future, the proposed policy framework system is expected to evolve to cope with more complex network scenarios, different services, and more complex security counter measures that impact not only the E2E delay but also the bandwidth consumption and/or packet drops.

ACKNOWLEDGMENT

The work presented in this paper is supported by the European Commission, under Grant No. FP7-224619 (the ResumeNet project).

REFERENCES

[1] T. Taleb, Y.H. Aoul, and A. Benslimane, "Integrating Security with QoS in Next Generation Networks," in *Proc. IEEE Globecom 2011*, Miami, USA, Dec. 2010.

[2] Z. Shen, J.P. Thomas, "Security and QoS Self-Optimization in Mobile Ad Hoc Networks," in *IEEE Transactions on Mobile Computing*, Vol. 7 No. 9, September 2008, pp. 1138-1151.

[3] C. Irvine, T. Levin, E. Spyropoulou, and B. Allen, "Security as a dimension of quality of service in active service environments," in *Proc. Active Middleware Services Workshop*, San Francisco, CA, USA, Aug. 2001.

[4] K. Simkhada, T. Taleb, Y. Waizumi, A. Jamalipour, and Y. Nemoto, "Combating against internet worms in large-scale networks: an autonomic signature-based solution," *J. Security and Communication Networks*, Vol. 2, No. 1, pp. 11–28, Aug. 2008.

[5] Z. M. Fadlullah, T. Taleb, A. Vasilakos, M. Guizani, and N. Kato, "DTRAB: Combating against attacks on encrypted protocols through traffic-feature analysis," *in IEEE/ACM Transactions of Networking (TON)*. (to appear)

[6] D. Brumley and D. Boneh, "Remote timing attacks are practical," in *Proc. 12th USENIX Security Symposium*, Washington DC, USA, Aug. 2003.

[7] S. N. Foley, S. Bistaerlli, B. O'Sullivan, J. Herbert, and G. Swart, "Multilevel security and quality of protection," in 1st *Workshop on Quality of Protection*, Como, Italy, Sep. 2005.

[8] W. He and K. Nahrstedt, "An integrated solution to delay and security support in wireless networks," in *Proc. IEEE WCNC*, Las Vegas, NV, USA, Apr. 2006.

[9] Z. Fadlullah, T. Taleb, N. Nasser, and N. Kato, "Exploring the security requirements for quality of service in combined wired and wireless networks," in *Proc. IWCMC'09*, Leipzig, Germany, Jun. 2009.

[10] J. Jason, L. Rafalow, and E. Vyncke, "IPSec configuration policy information model," Network Working Group, *RFC 3585*, Aug. 2003.

[11] S. Duflos, V. C. Gay, B. Kervella, and E. Horlait, "Improving the SLA-based management of QoS for secure multimedia services," in *Proc. 8th Int. Conf. on Management of Multimedia Networks and Services (MMNS'05)*, Barcelona, Spain, Oct. 2005.

[12] H. Otrok, M. Mehrandish, C. Assi, M. Debbabi, and P. Bhattacharya, "Game theoretic models for detecting network intrusions," *Computer Communications*, Vol. 31, No. 10, Jun. 2008. pp. 1934-1944.

[13] Y. Liu, C. Comaniciu, and H. Man, "A bayesian game approach for intrusion detection in wireless ad hoc networks," in *Proc. ACM GameNets'06*, Pisa, Italy, Oct. 2006.

[14] L. Chen and J. Leneutre, "A game theoretic framework of distributed power and rate control in ieee 802.11 wlans," *IEEE JSAC*, Vol. 26, No. 7, Sep. 2008. pp. 1128-1137.

[15] M. Scholler and P. Smith, "Resilience and survivability for future networking: Framework, mechanisms, and experimental evaluation," *ResumeNet Deliverable D1.1b*, URL: www.resumenet.eu.

[16] G.A. Fink, B.L. Chappell, T.G. Turner, and K.F. O'Donoghue , "A metrics-based approach to intrusion detection system evaluation for distributed real-time systems," Pentagon Reports, Apr. 2002.

[17] F. Bari and V. Leung, "Multi-Attribute Network Selection by Iterative TOPSIS for Heterogeneous Wireless Access," in Proc. IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, Jan. 2007.

[18] K. P. Yoon and C. L. Hwang, "Multiple attribute decision making: an introduction," in Sage Publications, Inc., Thousand Oaks, CA, USA, 1995.

[19] NS-3, "The network simulator ns-3", available at URL: http://www.nsnam.org/, 2011.

[20] P. Seeling, M. Reisslein, and B. Kulapala, "Network Performance Evaluation with Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial," IEEE Communications Surveys and Tutorials, Vol. 6, No. 3, Third Quarter 2004. pp. 58-78.

[21] C. H. Ke, C. K. Shieh, W. S. Hwang, A. Ziviani, "An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission, J. of Information Science and Engineering, Vol. 24, No. 2, pp.425-440, March 2008.

[22] R. Goyal and R. Jain, "Buffer management and rate guarantees for TCP over satellite-ATM networks," Int. J. Satellite Commun., vol. 19, no. 1, pp. 111129, 2001.

[23] J. Hu, I. Khalil, S. Han, A. Mahmood, "Seamless integration of dependability and security concepts in SOA: A feedback control system based framework and taxonomy", Journal of Network and Computer Applications, Volume 34, Issue 4, Pages 1150-1159, July 2011.