DSPM: Dynamic Security Policy Management for Optimizing Performance in Wireless Networks

Avesh K. Agarwal Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695

Abstract - Military wireless networks suffer from privacy and performance concerns due to their shared radio medium and off-the-shelf products. Therefore, robust and efficient security management is essential in these networks, especially for the transmission of sensitive data. However, security solutions based on static-configuration paradigm do not adapt to changing network conditions, such as variations in wireless link characteristics, leading to degradation in system performance. The rationale for advocating dynamic security paradigm is to achieve optimized network performance and security based on network conditions. Therefore, we propose a dynamic security policy management (DSPM) in which security policies can be changed on the fly based on the network feedback about wireless link conditions. DSPM is analyzed by using semi-Markov decision process to determine the optimal instances for switching security policies. The results show that DSPM provides enhanced security and improved performance than static security.

Keywords- Wireless networks, security, performance, semi-Markov decision process.

I. Introduction

Wireless networks provide many salient features such as Internet everywhere and mobility support which enable users to interact with others regardless of location. However, air broadcast medium used in wireless networks poses many challenging issues, relating to the security for mobile users. Since interception and modification of data in broadcast medium is very easy, it requires strong security solutions for wireless networks [12]. Therefore, many security solutions, some native to wireless networks and some adopted from wired networks, are used such as Wired Equivalent Privacy (WEP) protocol, 802.1x framework with EAP support, SSL, IPSec and 802.11i. However, configuration of the security policies in wireless networks has been static in a way that once a security solution is configured, it does not change on the fly until modified by the system administrator as the need arises.

In addition, low delay-bandwidth product in wireless networks causes poor quality of service (QoS) experience for mobile users [8]. Moreover, lossy wireless links, high contention in the network with the increase in number of users, and roaming scenarios create a challenging environment for providing required QoS. Besides, enabling security in wireless networks leads to further degradation in performance due to additional overhead of security services. As conditions in wireless networks change rapidly, static configuration of security does not take them into account and leads to poor coordination between security services and performance. Consequently, there is a need for a dynamic security management which can adapt to the changing environment in wireless networks based on network performance. There are some existing studies which have considered the dynamic configuration aspects of security in various other contexts.

For instance, a user level dynamic authentication protocol, named Authenticast, is proposed in [15]. Authenticast provides dynamic security by determining which parts of communication are carrying critical information. If non-critical communication can be transmitted unsecured or with lower security level, then the performance of the system can be improved. A similar protocol like Authenticast but providing different security levels for encrypted MPEG, named SECMPEG, is discussed in [10]. SECMPEG includes the capability to encrypt only the most important and significant data, in order to improve performance. As with [10], [7] also employs information, such as frame type, to select the particular frames to be encrypted. Besides these, new system architectures to support dynamic security for wired networks have been proposed in the past too. For example, a flexible security architecture with wide variety of security policies and mechanisms is proposed in [4]. It provides applications and users the ability to create and enforce highly customized and situational policies dynamically. In [5], authors propose using distributed firewalls with dynamic security policies to protect Intranets from external and internal attacks. They implement micro-firewall at each network node and all network nodes together decide the security policies updates.

We notice that existing studies for dynamic security configuration focus on analyzing information content for improving performance and security in wired networks. However, these studies do not consider network conditions and their interaction with the overhead associated with security services. It is due to the fact that these studies are based on wired networks where network conditions, such as link error rate, are highly stable. In addition, although these studies discuss dynamic configuration of security, but security policies can not be altered while session is going on. Therefore, basic fundamentals of our work are very different from these studies. As our work aims wireless networks, where link conditions change very fast, we propose dynamic security management by considering link conditions for improving performance and security in these networks. Our work is based on network feedback which helps wireless clients in making decisions regarding the dynamic configuration of security. Moreover, our work focuses on providing dynamic security configuration while user sessions are in progress. The main contributions in this work are as follows.

A. Contributions

We propose a dynamic security policy management (DSPM) system which can provide adaptive performance and security as required by a network. Adaptive behavior of the system is provided by switching security policies based on the network conditions such as bit error rates (BER) over links. We have provided a generalized semi-Markov decision process model to analyze DSPM, which helps in achieving optimized network performance and security. The advantage of dynamic security management can be described as follows.

- Dynamic security management provides a better control to system designers to achieve improved coordination between security and QoS.
- As the dynamic security management changes policies during runtime, it enables adaptive and enhanced performance and security in wireless networks.
- As the demand for QoS is increasing rapidly by real-time mobile applications, the dynamic management empowers better QoS experience for mobile users by using feedback from the network conditions.
- Since improved coordination between security services and QoS will lead to better resource management which, in turn, will improve network scalability.

The advantages of DSPM are supported by the observations based on our results. The observations show that as wireless link conditions degrade, DSPM adapts to a security policy with lower overhead. Whereas it is observed that as wireless link conditions improve, DSPM shifts to a stronger security policy. In addition, we notice that the length of time durations between two switching instances and packet size impact the performance of DSPM.

The rest of the paper is organized as follows. The relation between the overhead associated with security policies and network performance is discussed in Section II. In Section III, we explain main components of DSPM, analyze DSPM as a semi-Markov decision process, define cost matrices and optimality equations, and describe value iteration algorithm. The DSPM control algorithm and its implementation aspects are presented in Section IV. We demonstrate numerical results in Section V. Finally, Section VI provides conclusions along with future work.

II. Impact of Security Policies on Network Performance

In our work, we characterize network performance in wireless networks by considering throughput. Throughput, denoted as η, is considered for quantifying the system performance as perceived by a wireless client, and is defined as the number of packets successfully delivered per unit time. Throughput is affected depending upon how many packets are lost during transmission. For example, if packet losses are high in a wireless network, it will lead to low throughput. In addition, we know that the probabilities, whether a packet is lost or successfully transmitted, are dependent on packet size. When security policies are applied, extra bits are added to each packet leading to increased packet size. Increased packet size affects packet losses in two ways. First, a packet will have higher chance of collision due to its bigger size. Second, as link conditions in wireless networks changes rapidly causing varying bit errors on a wireless, bigger packet size has higher chance of having bit errors. Therefore, extra bits added by security policies directly impact packet loss probability.

Assume that a client is configured with security policy ρ_i, original packet size is d, and per packet additional bits added by security policy ρ_i is $O(\rho_i)$. Therefore, size of a transmitted packet will be $d+O(\rho_i)$. Assume that BER in wireless network is denoted as ε. Then, the probability that packet is lost, denoted as p_{loss}, can be obtained as follows.

$$p_{loss} = 1 - (1-\varepsilon)^{d+O(\rho_i)} . \tag{1}$$

Equation (1) shows the relationship between packet loss probability and security policy overhead. Specifically, (1) implies that if one or more bits are in error, packet is considered lost. It is due to the fact if packet is received with errors, the packet is discarded at the destination.

III. Dynamic Security Policy Management (DSPM)

In this section, first we discuss different components associated with DSPM system. Then we present semi-Markov decision process (SMDP) model to analyze DSPM. The cost in the model has been evaluated by computing the packet losses occurred during the configuration of different security policies. The analysis helps us in finding an optimal SMDP policy regarding the switching of security policies in changing wireless environment. To avoid confusion, we clarify that SMDP policy is different from a security policy. A security policy specifies a security protocol or a combination of security protocols at different layers configured in a system. On the other side, a SMDP policy specifies actions which guide switching among different security policies. The notations followed in this paper are as given in [14].

A. DSPM Architecture

Fig. 1 shows the main components of DSPM system which consists of a monitor, a decision-maker and a switching, modules. The monitor module collects statistics such as signal strengths and BERs over wireless links, and provides the feedback to the decision-maker at regular intervals. Whenever the decision-maker module obtains feedback from the monitoring system, it runs an algorithm to determine the decisions regarding the switching of security policies. The monitor and decision-maker are executed as background processes so that they do not interfere with the ongoing data transmission in a system. The decision-maker sends its decision to the switching module. The switching module finally changes the current security policy if the decision sent by the decision-maker module is positive. Since the switching module runs in foreground, it adds extra overhead to the ongoing data transmission. It is because the data can not be transmitted until the configuration of new security policy is completed. However in real scenario, the time involved between consecutive decisions made by the decision-maker module is in the order of hours. Whereas, the time involved in switching a security policy is in the order of seconds. By assuming that systems have enough buffer so packets are not lost and as the cost in our model are concerned with only packet losses but not delay, we ignore the switching time during the analysis.

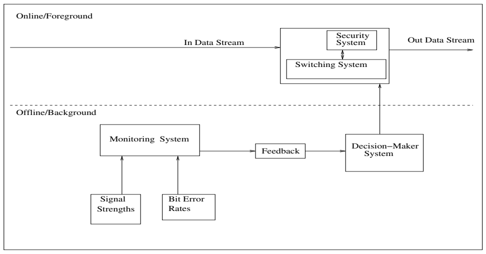

Fig. 1. Dynamic Security Policy Management.

B. Semi-Markov Decision Process Model

Here we formulate the problem of dynamic switching of security policies as a semi-Markov decision process (SMDP). SMDP consists of state space, action set for each state, decision rules, policies, decision epochs and cost functions [14]. In context of DSPM, the decision-maker module makes decisions regarding the switching of policies at different time instances called decision epochs based on the feedback obtained from the monitor module. Time instances from t_0 to t_k shown in Fig. 2 are the decision epochs. We assume that time duration between two decision epochs, denoted as τ, follow general distribution so that it covers variety of situations in real scenarios. For example, system designers can choose to monitor system either at constant intervals or whenever some event occurs such as change in BERs or a specific number of packets are lost. However, by choosing the general distribution, we do not restrict our model to some specific situation. Further, time instance t_0 is when the client enters the network and makes first decision to choose initial security policy. Moreover, time instance t_k represents when the client makes the last decision, and the client leaves the network at time T. Therefore, the total time T represents the client residence time in a network. We assume that T is exponentially distributed with rate λ.

The decision-maker module takes into account the states of the system while determining decisions. We represent the state space in the system as S where each $s \in S$ contains the current security policy configured in the system and the current BER. For example, if current state of the system is $s = (\rho, \varepsilon)$ at decision epoch k, then ρ is the security policy and ε is the BER. Besides, every state has an action set associated with it. The action set for a state s, denoted as A_s specifies the actions which are taken into consideration by the decision-maker when the system is in state s. We assume that action set associated with each state does not change with time. In this work, we assume that there are $at\ most$ two actions available for each state. Those two actions are defined as "SW" and "NSW". "NSW" refers that security policy should not be switched, whereas "SW" means that current security policy should be switched to some other security policy. It is important to note that it is possible that some states may have both actions associated with them, whereas other states may have just any one of "NSW" and "SW" actions associated with them. However, each state will have $at\ least$ one action associated with them. X_i and Y_i, where $0 \leq i \leq k$, in Fig. 2 are random variables showing state and action at each decision epoch, respectively.

Now we discuss state transition probability matrices P which describes how the transitions among different states take place given a particular action. We introduce two probability distribution functions to define state transition probability matrices. Let $p(\rho_{i'}/s, a)$ or $p(\rho_{i'}/\rho_i, \varepsilon_j, a)$ represent the probability that security policy in next state is $\rho_{i'}$ given the current state is (ρ_i, ε_j) and action a is chosen. In addition, let $p(\varepsilon_j)$ denote the probability that BER is ε_j. Then, state transition probability matrices can be represented as follows.

$$\mathbf{P}(\rho_{\mathbf{i'}}, \varepsilon_{\mathbf{j'}}/\rho_{\mathbf{i}}, \varepsilon_{\mathbf{j}}, \mathbf{a}) = \begin{cases} p(\rho_{i'}/\rho_i, \varepsilon_j, a) \cdot p(\varepsilon_{j'}) & \text{a=SW} \\ p(\rho_i/\rho_i, \varepsilon_j, a) \cdot p(\varepsilon_{j'}) & \text{a=NSW} \end{cases} . \tag{2}$$

In (2), when $a = NSW$, then $p(\rho_i/\rho_i, \varepsilon_j, a)$ will be equal to 1, and effectively $P(\rho_i, \varepsilon_{j'}/\rho_i, \varepsilon_{j'}, NSW)$ will be equal to $p(\varepsilon_{j'})$. (2) implies that security policy in next state depends upon current state, whereas BER in next state is independent of current state.

A SMDP policy, denoted as π, specifies the decision rules to be used at each decision epoch. A decision rule, denoted as δ_k, specifies the action chosen for each state $s \in S$ at the decision epoch k. For example, $\delta_k(s)$ denotes the action chosen for state s at decision epoch k, where $\delta_k(s) \in A_s$. Therefore a SMDP policy $\pi = \{\delta_1, \delta_2, \ldots\}$ is a set and consists of decision rules to be used at all decision epochs. Here, we consider stationary SMDP policies with deterministic Markovian decision rules. Stationary SMDP policies are policies where $\delta_k = \delta \forall k$. It means that action chosen at a particular state is same at all decision epoch. Since the action set associated with a state does not vary with time, therefore considering stationary SMDP policies is valid in our scenario. We denote the set of all stationary SMDP policies by Π. In addition, Markovian decision rules are rules which depend upon previous states and action only through current state. Deterministic nature of decision rules ensures that an action is chosen with probability 1. For details, readers can refer [14].

In addition, we define a cost rate function $r(s)$ associated with each state s, which captures the packet losses occurred when system is in state s. Assuming source rate is γ and packet loss probability associated with state s is $p_{loss}(s)$, then $r(s)$

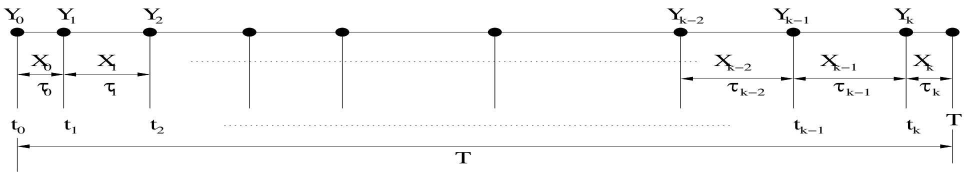

Fig. 2. DSPM as a Semi-Markov Decision Process.

will be equal to $\gamma \cdot p_{loss}(s)$. Therefore, if the time duration for decision epoch k is τ_k and state is s, Then, total cost incurred during the decision epoch period will be $\tau_k \cdot r(s)$. Now we compute the *expected total cost*, denoted as $v^{\pi}(s)$, assuming a SMDP policy π is chosen and initial state is s. By using Fig. 2, $v^{\pi}(s)$ can be deduced as follows.

$$v^{\pi}(s) = E_s^{\pi}\left\{\sum_{i=0}^{k-1} \tau_i r(X_i) + (T - t_k) r(X_{t_k})\right\}. \qquad (3)$$

The first term in (3) computes the cost upto decision epoch k, and the second term computes the cost between decision epoch k and the exit time instance T. As T is distributed with rate λ, $v^{\pi}(s)$ can be expressed as follows as derived in [16].

$$v^{\pi}(s) = E_s^{\pi}\left\{\sum_{i=0}^{\infty} e^{\lambda t_i} c(X_i, Y_i)\right\}. \qquad (4)$$

Where

$$c(s', a) = E_{s'}^{a}\left\{\frac{1}{\lambda}(1 - e^{-\lambda\tau}) r(s'')\right\}. \qquad (5)$$

Here $c(s', a)$ is the *expected total cost between two decision epochs*, when system is in state s' and action a is chosen. It is important to note in (5) that the cost rate is $r(s'')$, where (s'') is the system state after the decision epoch. It is due to the fact that the state change in our system takes place immediately after the decision instance as shown in Fig. 2. Now assuming that $G(t)$ represents the distribution for time duration between decision epochs, and is independent of the states and actions. Then (4) can be written as follows [14].

$$v^{\pi}(s) = c(s, a) + \sum_{s' \in S} \int_0^{\infty} e^{-\lambda t} v^{\pi}(s') P[s'|s, a] G(dt). \qquad (6)$$

Now, our goal is to find a SMDP policy π among all stationary policies, which minimizes $v^{\pi}(s)$. Assume that $v(s)$ represents the *minimum expected total cost* with initial state s. Then optimality equation can be written as follows.

$$\begin{aligned} v(s) \quad = \quad & \min_{a \in A_s}\{c(s, a) + \\ & \sum_{s' \in S} \int_0^{\infty} e^{-\lambda t} v^{\pi}(s') P[s'|s, a] G(dt)\}. \qquad (7) \end{aligned}$$

Optimality equation (7) implies that we need to find an action a for each state s which minimizes expected total cost. Now, we describe cost function, $c(s, a)$, which demonstrates *expected total cost between two decision epochs*. The cumulative cost over a decision epoch period will depend upon security policy and BER in that decision epoch period. Therefore, cost function for state s, if action a is chosen, can be computed as follows.

$$\begin{aligned} c(s, a) \quad & = \textstyle\sum_{s' \in S} \int_0^{\infty} \frac{1}{\lambda}(1 - e^{-\lambda t}) r(s') P[s'|s, a] G(dt) \\ & = (\textstyle\sum_{s' \in S} \frac{1}{\lambda} r(s') P[s'|s, a]) \int_0^{\infty} (1 - e^{-\lambda t}) G(dt). (8) \end{aligned}$$

Now we discuss the algorithm to find a stationary deterministic optimal SMDP policy. We use value iteration algorithm to find an optimal policy which is used widely to solve Markov decision processes [16]. We describe algorithm in Fig. 3 as given in [14], [16]. In this paper, the function $||v||$ is defined as $max_{s \in S} v(s)$. Since step 2 in the algorithm correspondence to contraction mapping, $v^n(s)$ converges in norm to $v(s)$. Step 2 in Fig. 3 finds the optimal value of cost, whereas step 4 chooses an optimal action for each state.

```
value_iteration_algorithm()
1) Set $v^0(s)$ = 0 for each state $s \in S$. Choose $\epsilon > 0$ and $n = 0$.
2) For each $s \in S$, compute $v^{n+1}(s)$ as follows.
      $v^{n+1} = min_{a \in A_s}\{c(s, a) + \sum_{s' \in S} \int_0^{\infty} e^{-\lambda t} v^n(s') P[s'|s, a] G(dt)\}$
3) If $||v^{n+1} - v^n|| < \epsilon$, go to step $4$. Otherwise $n = n + 1$ and go to step $2$.
4) For each $s \in S$, compute the stationary optimal policy as follows.
      $\delta(s) = arg\ min_{a \in A_s}\{c(s, a) + \sum_{s' \in S} \int_0^{\infty} e^{-\lambda t} v^n(s') P[s'|s, a] G(dt)\}$
5)exit.
```

Fig. 3. Value Iteration Algorithm for Finding An Optimal SMDP Policy.

IV. DSPM Control Algorithm

In this section, we describe DSPM control algorithm and its implementation aspects in detail. In real scenarios, DSPM control algorithm will be executed on every client system in the network. For better performance, the DSPM control algorithm is divided into two parts: offline and online. The offline part can be executed in background so that it does not interfere with the normal processing at client systems. The online part is to be executed as a foreground process so that security policy for next state can be determined while user session is going on. A sketch of the DSPM control algorithms is provided in Figs. 4 and 5.

A. Implementation Aspects of the Offline Part

During the execution of the offline part, various inputs such as set of security policies, set of BERs, probability distribution functions, overhead and utility functions are provided as shown in Fig. 4. The control algorithm considers only those security policies which are similar between a wireless client and access-points in a network. Information about the set of security policies associated with access-points can be obtained in advance from the network administrator.

```
Part 1: Offline
Global Input Parameters:
$\rho$ := set of security policies
$\varepsilon$ := set of BERs
S : $\rho$ x $\varepsilon$ := set of states
$O(\rho_i)$ := overhead associated with security policy $\rho_i$
$p(\rho_{i'}/\rho_i, \varepsilon_j, a)$ := probability that security policy in next state is $\rho_{i'}$
given current state is $(\rho_i, \varepsilon_j)$ and action $a$ is chosen.
$p(\varepsilon)$ := probability distribution function for BERs
$P(\rho_{i'}, \varepsilon_{j'}/\rho_i, \varepsilon_j, a)$ := state transition probability matrices
$A = (SW, NSW)$ := action set associated with each state
$d$ :=packet size
$G(t)$ := general distribution for decision epoch periods
$\lambda$:= mean residence time
$\gamma$ := source rate in packets/sec
$T$ := total residence time
Let initial state $(\rho_i, \varepsilon_j)$
$\delta$ = value_iteration_algorithm()
exit
```

Fig. 4. DSPM Control Algorithm: Offline.

Besides, we consider a finite set of BERs during the implementation of the control algorithm to have a finite state space. We assume that if BER is between ε_i and ε_j, then similar decisions will be taken for this range, and this range will be represented by the mean value of ε_i and ε_j. For instance, if BER is between $3.5e{-}4$ and $4.5e{-}4$, then this range will be represented by the value $4e{-}4$. The values of BERs which we have considered will be described when we discuss numerical results in a later section.

The value iteration algorithm function is called during offline part, and provides the optimal action associated with each state. The information about optimal actions acts as a look-up table during the online part of the DSPM control algorithm. The advantage of calling value iteration algorithm offline is that there is lesser time spent in finding optimal action from the look-up table than in determining optimal actions each time during the online part.

B. Implementation Aspects of the Online Part

In the online part as shown in Fig. 5, each iteration in while loop corresponds to each decision epoch period. Duration of each epoch is computed using the distribution function which can be exponential, uniform or any other distribution. Then, associated cost in terms of packet losses is computed by obtaining the product of packet loss probability, source rate and the time duration of decision epoch. Then, security policy in next state is obtained by using the look-up table δ and state transition probability matrices P as shown in steps $14-15$. In addition, prediction of BER in next state is achieved by using BER distribution function.

```
Part 2: Online
1) c_sp = $\rho_i$ \\ current security policy
2) c_ber = $\varepsilon_j$ \\ current BER
3) n_sp = NULL \\ security policy in next state
4) n_ber = NULL \\ BER in next state
5) r_t = $T$ \\ time remaining
6) c_p_l = 0 \\ cumulative number of packets lost
7)  while TRUE do
8)      $t = G(t)$ \\ returns random decision epoch period
9)      if r_t $\leq$ t then
10)         c_p_l = c_p_l + $\gamma \cdot p_e(\text{c\_sp}, \text{c\_ber})\cdot$ r_t
11)         break \\ out of while loop
12)     else \\ r_t $> t$
13)         c_p_l = c_p_l + $\gamma \cdot p_e(\text{c\_sp}, \text{c\_ber}) \cdot t$
14)         a=$\delta(\text{c\_sp})$ \\ From Offline Part
15)         Choose some $\rho_{i'}$ such that $p(\rho_{i'}/\text{c\_sp}, \text{c\_ber}, a) > 0$
16)         n_sp = $\rho_{i'}$
17)         n_ber = $\varepsilon_{j'}$, choose by using $p(\varepsilon)$
18)     end if
\\ n_sp and n_ber become c_sp and c_ber
\\ in next while loop iteration
19)     c_sp = n_sp
20)     c_ber = n_ber
21)  end while
22)     $\eta$ = $\frac{\gamma \cdot T - \text{c\_p\_l}}{T}$
23)  exit
```

Fig. 5. DSPM Control Algorithm: Online.

We notice that each step in the online part is $O(1)$, except the step 16 where security policy in next state is deduced. In the step 15, security policy for next state is determined by state transition probability matrix which is $O(k)$, where k is number of security policies. As k will be constant in general, the time complexity for each decision is of $O(1)$. As the while loop runs for the number of the decision epochs, and assuming there are n decision epochs, then the time complexity for the online part is of $O(nk)$ or $O(n)$, which is linear in n.

V. Numerical Results

We have implemented a generalized DSPM toolkit by using MATLAB toolbox [9] to analyze SMDP model and to obtain numerical results. We have made DSPM toolkit available for public use at [6]. To analyze DSPM, we consider three security policies based on their uses in wireless networks. First and second security policies consist of WEP and IPSec (AH) protocols, respectively. On the other side, third security policy is integration of 802.11i (AES) with IPSec. The additional overhead to each packet added by WEP, IPSec (AH) and 802.11i (AES) is around 7, 20 and 16 bytes, respectively, [1], [2], [3]. Therefore, the additional overhead added by 802.11i with IPSec is equal to 20+16 = 36 bytes. Besides, we consider 10 bit error rates, $(1/j)e-3$ where $1 \leq j \leq 10$, to model wireless link conditions [11]. The value of $(1/10)e-3$ models a good wireless channel, whereas $(1/1)e-3$ or $1e-3$ models a noisy channel. In addition we assume that BERs are equally distributed with probability $1/10$, so $p(\varepsilon_j) = \frac{1}{10}\ \forall\ j$. As we notice that our state space, $S = \rho \mathrm{x} \varepsilon$, now consists of 30 states. Due to the big state space, we provide value of probability matrix $p(\rho_{i'}/\rho_i, \varepsilon_j, a)$ in the appendix at the end. The design of the probability matrix follows that whenever BERs are higher, system should try to switch to security policies with low overhead, and when BERs are low, system tries to switch to policies with strong security. In this way, DSPM system provides complete control to system designers to choose probability transition matrix tuned to their network requirements, and then the optimal performance can be obtained by using the DSPM control algorithm. Further, we assume that, in general, decision epoch periods are exponentially distributed with parameter $\mu = 0.005$ and λ is $5e-4$, unless stated otherwise. In addition, packet size is varied from 128 to 1024 bytes, as most of the applications in Internet transmit packet of sizes in this range [13]. Source rate γ is 10 packets/sec, and ϵ is $1e-20$ for value iteration algorithm.

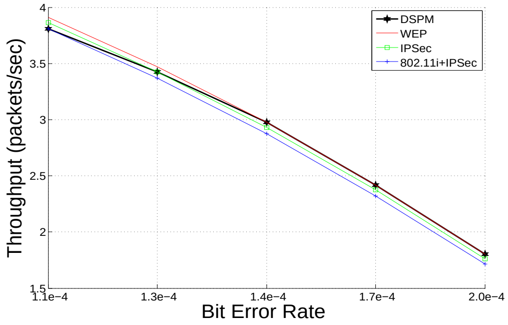

Fig. 6. DSPM Adaptability to Network Conditions.

Fig. 6 demonstrates how DSPM adapts to different wireless link conditions. We notice that as a wireless system is good with low bit error rates, DSPM uses 802.11i with IPSec, since 802.11i with IPSec is assumed to provide strong security in the system. On the other side, whenever bit error rates are in middle range such as $1.4e-4$, DSPM uses IPSec to improve performance. However, whenever system is noisy, DSPM uses WEP to improve performance at the cost of low security. It is to note that some wireless networks may not allow WEP at all due to their sensitive data contents, however probability matrices should be modified accordingly to find optimized performance and security tradeoffs in that scenario.

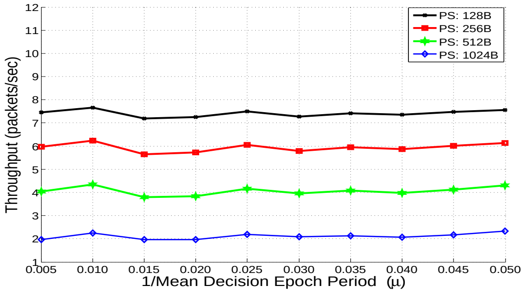

Fig. 7. Decision Epochs Duration vs. Packet Size.

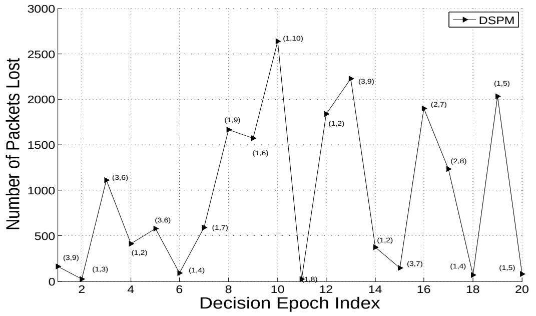

Fig. 8. DSPM Switching Pattern.

Now we compare the performance of DSPM with variations in mean decision epoch periods and packet size as shown in Fig. 7. We notice that DSPM shows better performance with small packets than large packets. It is due to the fact the large packet has higher probability of getting lost than small packets. Moreover, we notice as mean decision epoch period is reduced, DSPM provides better performance. This is because that as frequency of decisions is high, DSPM adapts better to network conditions, and throughput is improved consequently. To emphasize the fact further, we show the switching pattern followed by DSPM in Fig. 8. Although we found the switching pattern for a very large number of decision epochs, here we present the pattern only for 20 decisions epochs for clarity purpose. Every set (a, b) at each decision epoch in Fig. 8 shows the index of security policy and bit error rate as given in the appendix. We notice that as bit error rates are varied faster, DSPM reacts faster and switches to other security policies for optimization. For instance, BER is varying fast at decision epochs from 2 to 6, so the switching of security policies is taking place at each decision epoch. However, as packet losses are high in next decision epochs from 7 to 12, security policy is not switched and maintained at WEP (index 1) to improve the long term throughput. Therefore, Fig. 8 depicts the adaptability of DSPM at micro level. In our future work, we aim to find an optimum value of decision epoch period by considering cost of switching among security policies.

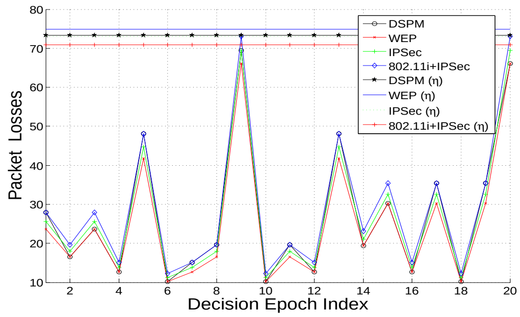

Fig. 9. Average Throughput vs. Packet Losses During Decision Epochs.

Another advantage of the DSPM is shown in Fig. 9. Here, source rate is set to 100 packets/sec. We notice that there are high variations during different decision epochs with regard to packet losses. It appears that the performance of DSPM is similar to that of static security management. However, the long term throughput shown above in Fig. 9 shows that DSPM throughput is higher than that of IPSec and 802.11i with IPSec policies. Although, DSPM throughput is lower than that of WEP, but as DSPM uses WEP, IPSec and 802.11i dynamically, it is obvious that DSPM provides stronger dynamic security than WEP over a long time period. Therefore, it can be concluded that DSPM optimizes performance and security in a better way than static security management.

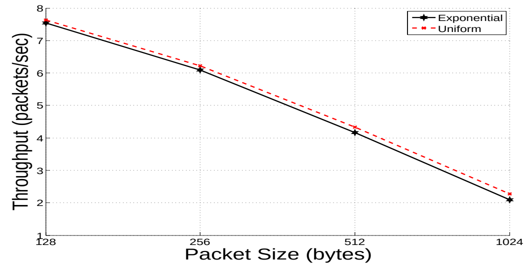

Fig. 10. Exponential vs. Uniform Distributions.

In the model of DSPM, we consider that decision epoch durations follow general distributions. Fig. 10 shows the performance of DSPM for exponential and uniform distributions (with interval $[0, \frac{2}{\mu}]$) with equal mean value μ. We notice that DSPM performs better with uniform distribution. The reason is that decision periods in the uniform distribution are always less than $\frac{2}{\mu}$ unlike unrestricted length of periods in exponential, which leads to more number of decisions in uniform distribution for equal total time. Consequently uniform distribution is able to adapt better to network conditions as explained above. In our future work, we want to analyze the DSPM with other distribution as well to determine the optimal decision periods according to network conditions.

VI. Conclusions

In this work, we proposed a dynamic security policy management (DSPM) system which adapts to network conditions and optimizes the performance accordingly. We analyzed the DSPM by using semi-Markov decision process, and found the optimal policies by using value iteration algorithm. In addition, we analyzed the performance of DSPM in terms of throughput by computing cost functions based on packet losses. We demonstrated by numerical results obtained in different scenarios that DSPM adapts to network conditions fast, and provides a tradeoff between system performance and security according to network requirements. As link conditions changes very fast in wireless network, we believe that DSPM can be highly beneficial in such environments. Moreover, it can provide better control of security management to system designers. To the best of our knowledge, we believe that this is first work which combines network feedback and overhead associated with security policies for providing dynamic security management. For our future work, we are implementing DSPM control algorithm in real time scenarios consisting of wireless LAN and ad-hoc networks.

References

[1] IPSEC. *http://www.freeswan.org*.

[2] IEEE 802.11i. Available at http://standards.ieee.org/.

[3] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications:The Insecurity of 802.11. In *Proc. of the ACM MobiCom'01*, pages 180–189, July 2001.

[4] R. H. Campbell, Zhaoyu Liu, M. D. Mickunas, P. Naldurg, and Yi Seung. Seraphim: Dynamic Interoperable Security Architecture for Active Networks. In *Proc. of the IEEE Third Conference on Open Architectures and Network Programming (OPENARCH'00)*, pages 55–64, March 2000.

[5] Kai Hwang and M. Gangadharan. Micro-firewalls for Dynamic Network Security with Distributed Intrusion Detection. In *Proc. of the IEEE International Symposium on Network Computing and Applications (NCA'01)*, pages 68–79, October 2001.

[6] NetWIS LAB. http://www.ece.ncsu.edu/netwis/dspmtoolkit.

[7] Y. Li, Z. Chen, S. Fan, and R. Campbell. Security Enhanced Mpeg Player. In *Department of Computer Science, University of Illinois at Urbana-Champaign*, 1996.

[8] S. Maniatis, E. Nikolouzou, and I. Venieris. End-to-end QoS Specification Issues in the Converged All-IP Wired and Wireless Environment. *IEEE Communications Magazine*, 42(6):80– 86, June 2004.

[9] MATLAB. http://www.mathworks.com.

[10] J. Meyer and E Gadegast. Security Mechanisms for Multimedia-Data with the Example Mpeg-I-Project. In *Project description of SECMPEG*, 1995.

[11] E. Modiano. An Adaptive Algorithm for Optimizing the Packet Size Used in Wireless ARQ Protocols. *Wireless Networks*, 5(4):279–286, July 1999.

[12] N.Borisov, I.Goldberg, and D.Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proc. of the ACM MOBICOM'01*, pages 180–189, July 2001.

[13] S. Pollin, A. Motamedi, A. Bahai, F. Catthoor, and L. Van der Perre. Delay Improvement of IEEE 802.11 Distributed Coordination Function using Size-based Scheduling. In *IEEE ICC'05*, volume 5, pages 3484 – 3488, May 2005.

[14] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley, New York, 1994.

[15] P. A. Schneck and K. Schwan. Dynamic Authentication for High-Performance Networked Applications. In *Proc. of the Sixth International Workshop on Quality of Service (IWQoS'98)*, pages 127–136, May 1998.

[16] V.W.S. Wong, M. E. Lewis, and V.C.M. Leung. Stochastic Control of Path Optimization for Inter-Switch Handoffs in Wireless ATM Networks. *IEEE/ACM Transactions on Networking*, 9(3):336–350, June 2001.

APPENDIX

Security Policy	WEP	IPSec	802.11i(AES)+IPSec
Index	1	2	3

BER	$(\frac{1}{10})e-3$	$(\frac{1}{9})e-3$	$(\frac{1}{8})e-3$	$(\frac{1}{7})e-3$	$(\frac{1}{6})e-3$	$(\frac{1}{5})e-3$	$(\frac{1}{4})e-3$	$(\frac{1}{3})e-3$	$(\frac{1}{2})e-3$	$1e-3$
Index	1	2	3	4	5	6	7	8	9	10

Probability Transition Matrix ($p(\rho_{i'}/\rho_i, \varepsilon_j, a)$)

```
Actions-->                SW                 NSW
Security policies-->  1   2   3           1   2   3

States
(1,1)                [0.0 0.0 1.0]
(1,2)                [0.0 0.0 1.0]
(1,3)                [0.0 0.0 1.0]
(1,4)                [0.0 0.5 0.5]       [1.0 0.0 0.0]
(1,5)                [0.0 0.5 0.5]       [1.0 0.0 0.0]
(1,6)                [0.0 0.5 0.5]       [1.0 0.0 0.0]
(1,7)                [0.0 0.5 0.5]       [1.0 0.0 0.0]
(1,8)                [1.0 0.0 0.0]
(1,9)                [1.0 0.0 0.0]
(1,10)               [1.0 0.0 0.0]
(2,1)                [0.0 0.0 1.0]       [0.0 1.0 0.0]
(2,2)                [0.0 0.0 1.0]       [0.0 1.0 0.0]
(2,3)                [0.0 0.0 1.0]       [0.0 1.0 0.0]
(2,4)                [0.5 0.0 0.5]       [0.0 1.0 0.0]
(2,5)                [0.5 0.0 0.5]       [0.0 1.0 0.0]
(2,6)                [0.5 0.0 0.5]       [0.0 1.0 0.0]
(2,7)                [0.5 0.0 0.5]       [0.0 1.0 0.0]
(2,8)                [1.0 0.0 0.0]
(2,9)                [1.0 0.0 0.0]
(2,10)               [1.0 0.0 0.0]
(3,1)                [0.0 0.0 1.0]
(3,2)                [0.0 0.0 1.0]
(3,3)                [0.0 0.0 1.0]
(3,4)                [0.5 0.5 0.0]       [0.0 0.0 1.0]
(3,5)                [0.5 0.5 0.0]       [0.0 0.0 1.0]
(3,6)                [0.5 0.5 0.0]       [0.0 0.0 1.0]
(3,7)                [0.5 0.5 0.0]       [0.0 0.0 1.0]
(3,8)                [1.0 0.0 0.0]
(3,9)                [1.0 0.0 0.0]
(3,10)               [1.0 0.0 0.0]
```