User Facilitated Congestion and Attack Mitigation*

Mürsel Yildiz, Ahmet Cihat Toker, Fikret Sivrikaya,
Seyit Ahmet Camtepe, and Sahin Albayrak

DAI-Labor / Technische Universität Berlin, Germany
{muersel.yildiz,ahmet-cihat.toker,fikret.sivrikaya,ahmet.camtepe,
sahin.albayrak}@dai-labor.de

Abstract. The IEEE Wireless LAN standard has been a true success story by enabling convenient, efficient and low-cost access to broadband networks for both private and professional use. However, the increasing density and uncoordinated operation of wireless access points, combined with constantly growing traffic demands have started hurting the users' quality of experience. On the other hand, the emerging ubiquity of wireless access has placed it at the center of attention for network attacks, which not only raises users' concerns on security but also indirectly affects connection quality due to proactive measures against security attacks.

In this work, we introduce an integrated solution to congestion avoidance and attack mitigation problems through cooperation among wireless access points. The proposed solution implements a Partially Observable Markov Decision Process (POMDP) as an intelligent distributed control system. By successfully differentiating resource hampering attacks from overload cases, the control system takes an appropriate action in each detected anomaly case without disturbing the quality of service for end users. The proposed solution is fully implemented on a small-scale testbed, on which we present our observations and demonstrate the effectiveness of the system to detect and alleviate both attack and congestion situations.

Keywords: POMDP, User Centric Networks, Quality of Experience, Load Balancing in Wireless LAN.

1 Introduction

Today's end user connectivity is increasingly provided by a number of short-range wireless access technologies, among which IEEE 802.11 based Wireless LAN (WLAN) standard is the de facto solution. Both residential and enterprise environments have witnessed rapid expansion of WLAN coverage due to

* This work was supported in part by the European Commission's Seventh Framework Programme (FP7) through project ULOOP (User-centric Wireless Local Loop), grant agreement no. 257418.

K. Pentikousis et al. (Eds.): MONAMI 2011, LNICST 97, pp. 358–371, 2012.

its ease of deployment, low cost, and ever-increasing data rates provided by the evolution in 802.11 family of protocols. On the other hand, the Internet, to which WLAN provides the most common form of last hop connectivity, has been transformed into a critical infrastructure due to the ongoing convergence in telecommunications and emerging services on IP-based data networks. This trend raises the importance of maintaining high and reliable Quality of Experience (QoE) provided by wireless access networks, which is usually considered to be the bottleneck in an end-to-end broadband connection. However, widespread use of resource hungry applications over wireless networks combined with the high density of WLAN networks has started hurting users' experience, which is intensified by uncoordinated deployment and operation of wireless access points.

The ubiquity of wireless access points in today's networks also place them at the center of attention for malicious security exploits. A common form of security attacks in the networking domain is *Denial of Service (DoS)* and its distributed form (DDoS), which also applies well to wireless networks. DoS attacks either crash the system by exploiting a software vulnerability in the target system causing the server to be crashed or freeze, or sending massive volumes of dummy traffic to occupy all resources that could service legitimate requests [1]. Another attack type that emerged lately and has a similar behavior to DoS attacks is *Reduction of Quality (RoQ)*. In RoQ attacks, an adversary does not try to block access to the system but may significantly reduce service quality by injecting carefully timed and adjusted requests to the system (e.g. for impeding the slow start and congestion avoidance mechanisms of TCP), thereby causing the system to become very inefficient or unstable [2]. The main characteristic of RoQ attacks is that they are much more difficult to be detected compared to simple DoS attacks.

There exist many state of the art solutions in the literature for anomaly detection and attack deflecting problems in wireless networks; however, existing work mostly rely on known attack profiles as well as on complete observation or knowledge about the environment. Based on the observations that i) wireless networks are growing rapidly with larger deployments in campus areas and enterprises, and ii) wireless attack techniques are evolving day by day giving rise for WAPs to be easily intruded, we observe a strong need for advanced semantic precautions in wireless networks that can dynamically adapt to changing conditions and work well with partial information. Moreover, existing security solutions' major focus is on identifying and mitigating attacks while maintaining the users' quality of experience level is considered only at a secondary level, even if not neglected. As a result, security countermeasures may have the side effect of exposing users to poor QoE due to i) congestion situations which may be misperceived as an attack, or ii) precautions taken by the network against attacks with quality-degrading side effects.

In this paper, we propose an intelligent distributed control system that is capable of detecting anomalies in wireless local area networks and differentiating attack situations from overload cases, so that appropriate actions can be taken in each case without adversely affecting the quality of service. Based on the

mathematical model of Partially Observable Markov Decision Processes (POMDP), our system provides an integrated solution to congestion avoidance, anomaly detection, and DDoS / RoQ attack detection and prevention.

1.1 State of the Art

Jatinder et al. propose a detector system for reduction of quality attacks with response stage by checking RTS/CTS packets from the MAC layer [3]. The authors propose using three main patterns inside the MAC layer: frequency of receiving RTS/CTS packets, frequency of sensing a busy channel and number of RTS/DATA retransmissions for detecting the RoQ attacks.

A system against DDoS attacks is proposed in [4] that monitors and constructs a database of IP source addresses in the network. The authors claim that it is possible to detect DDoS attacks by using a carefully pre-built IP address database and sequentially monitoring the proportion of new source addresses. However, a new user can still being interpreted as a malicious. A DoS attack generally uses a large number of similar packets deviating from the normal traffic patterns. Based on this assumption, Kulkarni proposes a Kolmogorov complexity based detection algorithm to identify the attack traffic [5]. A large number of users requesting a service from the same destination inside the network are likely to be suffering from an attack prevention action.

Load balancing is the logical action taken after a decision that the network is congested, based on measurements that signal to the controller that the network is experiencing excessive load. Frame drop rate of real-time sessions in an access point's transmission queues can reflect its load [6], which can be utilized for load measurements. This method seems to be a theoretically well-defined measurement; however, those basic low-level estimations assume certain implementation conventions and can not be applied to all products [7]. Similarly, delay time between scheduled and actual transmission time of periodic beacon frames can be a good measure for the load of an AP as proposed by [8].

Having a measure of the load on the AP, it is possible to balance the load among APs in the network through both wireless station (WS)-based solutions or network-based solutions. AP selection for WS-based approaches can be realized in a static or dynamic fashion; however, letting stations to dynamically choose an AP can lead to unstable WS-AP associations. As a result, similar measurements among nearby wireless stations would create a collective handoff process, causing a so-called ping-pong effect [7]. A possible remedy to this problem is to assign random waiting times and number of measurement instances for each WS before executing the handover [9]. On the contrary, [10] proposes an AP-based load balancing system for which overloaded APs reduce their transmission power of beacon signal so that it is less likely to be discovered by new stations. However, this approach may have an adverse effect on the quality of experience for WSs that are already associated. Moreover, this method does not guarantee load balancing among APs in the case that most APs decrease their transmission power in a similar fashion.

In this work, we have used a rather simple observation for measuring the load on an AP for load balancing similar to the one proposed in [11], which proposes a straightforward solution for measuring the load of an AP as the percentage of the time that the AP transmits or receives data during some time intervals. Similarly, observations for RoQ and DDoS attacks are also chosen to be simple for ease of implementation. However, more sophisticated observations can easily be incorporated in our solution without any modification to the decision engine.

1.2 Beyond State of the Art

In this study, we propose an intelligent system that is equipped with artificial intelligent techniques in corporation with trusted users inside the network. Our WLAN architecture interprets the users as one of the key components of the network when evaluating network performance in order to detect anomalies, specifically RoQ or DDoS attacks or congestion. We propose a new WLAN architecture for which state of art solutions for load balancing and attack prevention can be seen as functional blocks of the network that can be developed and trained with better solutions in the future. Observations and decisions for actions are done remotely together with the help of network users in an intelligent way, which is a more secured process for detecting and preventing from new attack techniques.

In this paper, we propose a system that is capable of sensing air traffic remotely and detecting all APs around the network, which gives the opportunity for researchers to add a simple additional functionality to detect any unknown APs in the vicinity and to prevent users from associating to misconfigured APs with the proposed client network manager program.

2 Solution Model

2.1 Partially Observable Markov Decision Process

Markov Decision Process (MDP) is a formal mathematical framework used to develop decision makers that control Markovian processes. In MDP formulation the next state depends on the current state and the action taken, thus the conditional transition probability between from s_i to s_j becomes a function of the action a, i.e. $p_{ij}(a)$. In addition to the actions, MDP framework associates rewards r_i with each state. The decision maker observes the current state, takes an action according to the control policy. The stochastic process underlying the system lands in the next state according to the conditional transition probabilities, and the reward associated with the next state is gathered by the controller. The decision maker starts the same procedure beginning with the new current state. The design problem in MDPs is to develop a policy π, which associates an action for each state, so that the long term total reward is maximized. The optimal policy π^* can be found using the well known Dynamic Programming (DP) algorithm and the Bellman's Condition. An excellent introduction to MDPs and DP can be found in [12].

Partially Observable MDPs (POMDPs) extend the MDP framework to systems in which the system states are not completely vivid, but only partially observable, through imperfect observations. A priori observation probability distributions describe how likely an observation is for each state. In POMDP formalization, decision maker periodically makes observations and keeps a Bayesian estimate of the likelihood of each state. This estimate is also called the belief b_i associated with the state s_i. POMDP policies associate mutually exclusive partitions in the belief space with individual actions. After each observation, the POMDP controller calculates the partition that the current belief belongs to and executes the action for the partition.

Some POMDP models have a special property called the *finite transience*. In such models the observations transform belief values belonging to an individual belief partition jointly to another belief partition. Since each belief partition has a single action associated with it, this property makes the implementation of optimal policy π^* as a Finite State Controller (FSC). FSCs can be described by state transition graphs, whose nodes represent actions and the directed edges represent observations. A more detailed introduction to POMDPs is given in [13].

The model we propose in the rest of this section is an outcome of our experimental studies and fine tuning of the involved parameters (states and transition probabilities, etc.) for intuitive behavior of the system. For clarity of presentation we directly provide the resulting Markov state diagram here, depicted in Figure 1, without presenting the steps in the evolution of this model. As will be discussed in the last section, our ultimate goal of a more dynamic and self-learning system to generate and optimize the parameters of the model is within our planned research agenda.

2.2 States

In our proposed model, there are nine states, four of which are defined to be the main states; namely, *OK*, *Congested*, *Attacked* and *Critical System Failure*. OK state represents the world state of a full performance working AP condition. In Congested state observed AP is congested giving rise to bad network experience for users. Attack state is another main world state representing an attack case to the observed AP. Finally, Critical System Failure state represents a failure case for any backbone element in the network serving the users.

In addition to these four main states, five intermediate states are defined for doing additional observations. First intermediate state is the *OK to Congestion* (O_C) state, in order to make SNMP check observation in addition to observed bad data traffic and bad user network delay experience. Secondly, in *Ok to Attack* (O_A) state, data traffic check observation is done in addition to observed bad user network delay experience and bad SNMP check observation. *Ok to Critical System Failure* (O_S) state is the third state for which SNMP check observation is done in addition to observed good data traffic and bad user network delay experience. In *Attack to Congestion* (A_C) state, SNMP check observation is done in addition to observed bad data traffic and bad user network delay experience. Finally, *Attack to Critical System Failure* (A_S) state is for data traffic check

observation that is done in addition to observed bad user network delay experience and bad SNMP check observation. These additional states are introduced in order to differentiate the similarly characterized *Congestion* and *Attack* states, after the initial observation of an anomaly leading to those states.

2.3 Observations

There are three observations in our proposed POMDP model. *Data traffic observation* is done with patched *airodump-ng* tool[1] in order to sense high and low data traffic rates for the observed AP. Secondly, during *SNMP check observation*, CPU load and MAC addresses together with the IP addresses of users attached to the observed AP is fetched with SNMP protocol in order to sense malicious programs keeping CPU busy or malicious users associated with the observed AP who are not authorized for user experience database. Final observation is the *User network delay experience observation*, which is a core feature of the proposed system. Our system is a combination of one of well known tools in artificial intelligence, namely POMDP and trusted user corporation in order to interpret feedback about network services from users in an intelligent way. In this observation, delay times are read from provided database and compared with a threshold in order to differentiate between an acceptable delay time and an unacceptable one.

2.4 Actions

No action is required for intermediate states during further analysis and OK state, which is also introduced as one of the four actions in our model. Secondly, *Attack Response* is defined to be command from controller AP on the observed AP to drop packets of unauthorized user detected. *Load Balancing* is the third action where controller AP starts a network initiated handover process and commands on the user with worst network delay time experience to hand off another free AP. Finally, by *Critical System Failure Report* action, controller AP reports a critical system fail report to the admin.

2.5 Rewards

No action is rewarded highly for the OK state and intermediate states. On the contrary, it is a low reward action for attack, congested and critical system failure states. It is unnecessary to take an action for a desired OK state and not feasible to take an action without increasing the belief of the states during intermediate states. Attack response is rewarded highly only for a possible attack state; however, it has a very low reward for a congested state because of the probability for a user to be interpreted as a malicious user, which gives rise to suffering users from the false alarms for attack precautions of the network. On the contrary, load balancing is rewarded highly for congested state and very low

[1] http://www.aircrack-ng.org

for an attack case. This is because of the fact that a RoQ attack would spread easily among APs in case of a false alarm for a congestion situation. Critical system failure report is rewarded highly for critical system failure state but very low for an OK belief state, especially for lazy admins who do not want to check the network frequently due to false alarms.

2.6 State Transition Functions

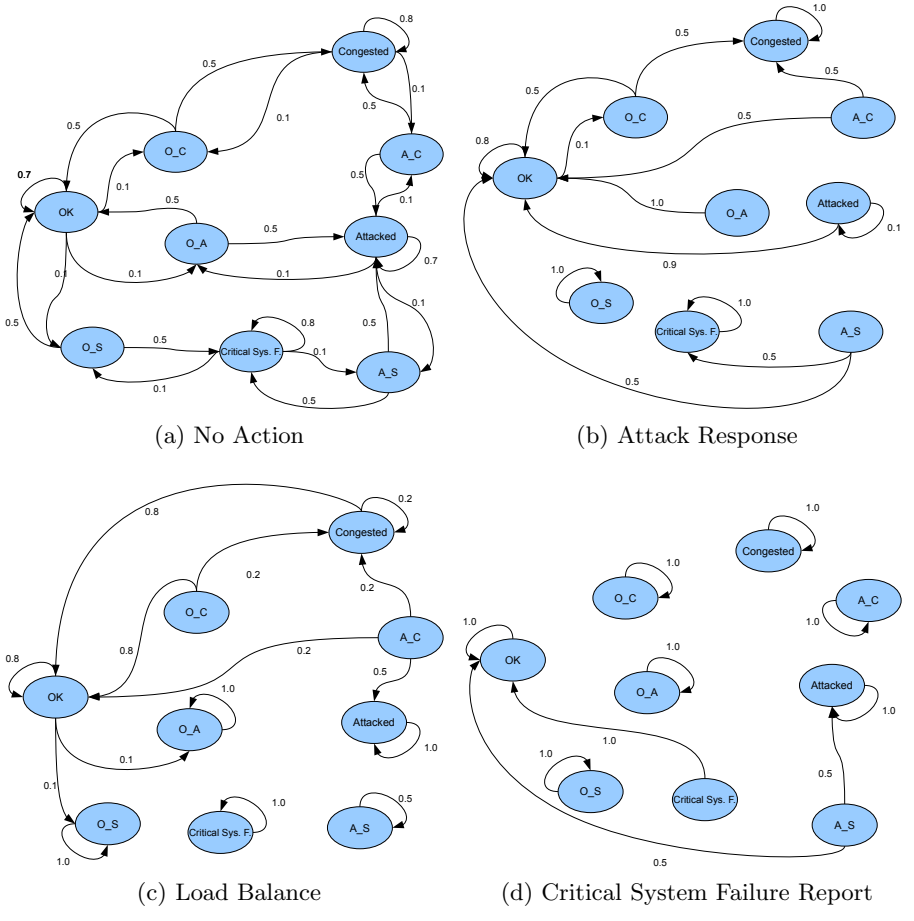
As observed from Figure 1a, there are mainly no direct transition from OK state to any other state during analysis with the taken observations. No action maintains most probably the belief state except for the intermediate states for which there is an uncertainty that should be eliminated by additional observations during controller operation. Attack response results most probably in a transition from O_A and *attacked* states to the OK state and it has almost no effect for the other states. Similarly load balancing results in a transition from intermediate O_C state and *congested* state to OK state. This action is dangerous for the intermediate states between *attacked* and *congested* and results most probably in a transition to the *attacked* state. This is because of the characteristics of RoQ attacks, which do not completely deny network services but throttle them as if there exists a congestion case inside the network. Unfortunately, if the controller tries to share the load with the attacked AP, RoQ attack would spread inside the network. Finally, critical system failure report has nothing to do with *attacked* and *congested* states. It only results in a transition from critical system failure state to the OK state and with 50% probability from the intermediate state A_S to the OK state. This is because of the uncertainty for this intermediate state for which there exits a 50% probability of having a system failure inside the network.

3 Software Components

In our proposed solution, a *network manager component* is developed for the clients in order to corporate with the *controller component* on the access point. Software block diagram is given in Figure 2

3.1 Client Network Manager Program

The main functionalities of this software component are i) recognizing and configuring wireless LAN interface of the client, doing initial network association with the pre-determined access point, ii) performing network tests by periodically manipulating DITG (Distributed Internet Traffic Generator) [14] and generating randomized TCP and UDP packets, iii) processing received round trip time for each generated traffic and writing them periodically to the QoE database, iv) communicating with the controller AP. Moreover, this component stores the client's experience locally for further analysis and tests.



(a) No Action (b) Attack Response

(c) Load Balance (d) Critical System Failure Report

Fig. 1. State Transition Probabilities

In fact, It is possible to extract user QoE of delay time even on the access point at the edge of the network with a protocol analyzer software tracking request time and answer time for each flow specific to the users. However due to performance issues and scalability, it is a better solution to distribute observations throughout the network rather than overloading the APs on the network. Similarly, registering the MAC address of some users in the network to the blackmail lists of APs dynamically and forcing users to perform handover might be another suggestion for avoiding the need of an additional network controller software at user clients. However, the handover would not be seamless and would result in delay times and distributions at the user clients during handover processes. Although it is a logical suggestion to avoid extra network manager software on the clients, due to these reasons, we propose to include this software.

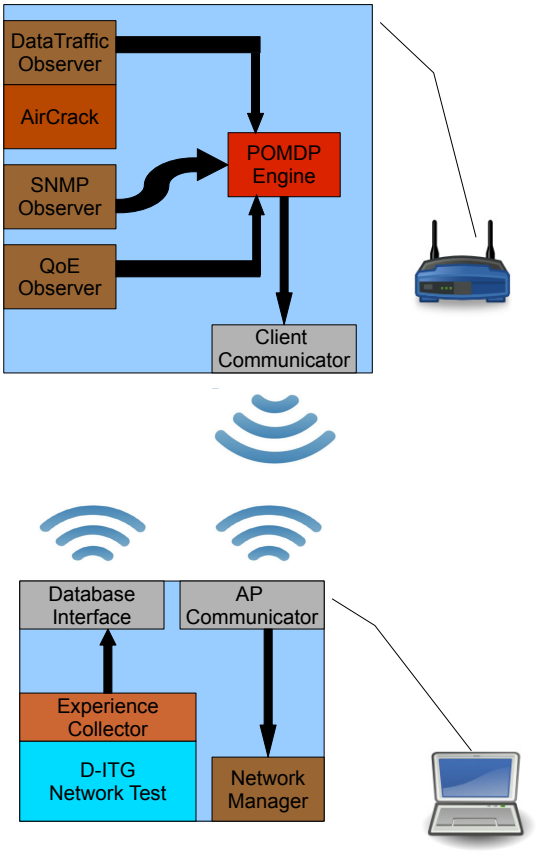


Fig. 2. Software Block Diagram

3.2 Controller Access Point Program

This is a multithread program that is responsible for performing observations, by opening a domain socket and communicating with the patched *airodump-ng* to count data packets through the observed AP with a specific BSSID and channel. Data count values are compared with a threshold periodically in order to set *Bad Data Traffic* or *Good Data Traffic* observation flags. The controller program also checks user QoE entries periodically and sets *Bad / Good Experience flag* when necessary. It periodically checks CPU load for malicious programs and associated user's IP and MAC addresses. Moreover, it compares this addresses with the ones in QoE database in order to detect malicious clients for a possible RoQ attack detection. This program sets *Bad SNMP* observation flag in case of: i) no response from observed AP, ii) bad CPU load observation, or iii) after comparing authenticated users with the ones who are really associated with the AP.

Decision engine is the core thread of the controller component, where POMDP policy graph is implemented for the optimal decision of actions on the network. This thread periodically collects all observations from other threads, combines them for possible OK, attack, congestion or system failure observations, and takes an action accordingly. This thread takes no action in case of an OK observation. It sends commands to the observed AP to drop packets of unauthorized user detected by SNMP check thread. In case of a congestion decision, it gives a handover command to the user whose IP address is detected by database check thread as the most suffering client. Finally, it reports to the network admin a system fail in case of a critical system failure observation.

4 Evaluation

After solving the POMDP model we presented above, we have obtained a FSC with 29 states. In order to evaluate the control policy we have developed three scenarios, chosen as representative operation conditions of the controller. We have also set up a testbed to run these scenarios realistically.

The reaction time of the system depends on:

- QoE of delay time observation period
- Data traffic observation period
- SNMP request observation period

It is possible to enhance the reaction time of the system by tuning the observation periods, however due to experimentation objectives and in order not toe overload the network traffic with observations, we tuned the reaction time of the system to a slower rate.

4.1 Testbed

Difficulties for conducting experiments with real wireless networks gives rise to the fact that majority of publications in this area are based on simulation results [15]. In order to test our proposed model in a physical system, we set up an IEEE802.11 WLAN test depicted in Figure 3.

In our testbed there are two WLAN access points (AP). On AP1 we run the POMDP controller. The second AP is observed by the POMDP controller on AP1. The POMDP controller is responsible for controlling both APs.

We have used six Linux-based notebooks running multiple threads and emulating a large user population. The users are divided into traffic generating users, malicious users and normal users. The POMDP controller is responsible for the QoE of the normal user pool. We use the distributed traffic generator D-ITG [14] to emulate the users. D-ITG server resides on the application server and serves as a TCP traffic source. An independent D-ITG receiver runs for each user on the user laptops. The receivers request TCP packets that are exponentially distributed with a mean of 750 Bytes. The inter-arrival time of these packets are exponentially distributed with an average of 1 ms. After completion of packet download, D-ITG clients on the user side report the experienced delay to the QoE database.

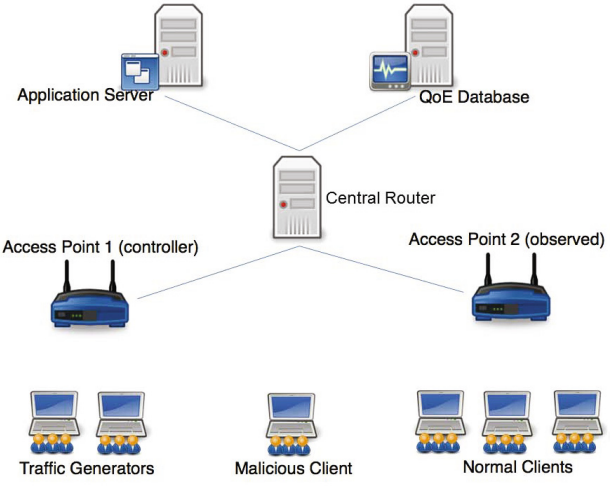


Fig. 3. Testbed Block Diagram

4.2 Scenarios

We consider the following three scenarios to evaluate the performance of the control policy.

Attack Case. We simulate a RoQ attack with the sudden appearance of 20 malicious flows. The malicious users aim to reduce quality of service in AP2, by initiating numerous service requests. Our system is able to discern between a RoQ attack and a congestion by the virtue of the comparing the QoE database entries and SNMP requests. Since the malicious users are not writing to the QoE database, our POMDP controller is able to infer that there is a RoQ attack. We initiate two separate RoQ attacks and split the the delay experienced by a normal user in Figure 4.

The first RoQ attack is initiated at T2 instant and stopped at T3. This is a short RoQ attack, which reduces the QoE of the user. However, it is not long enough for our system to register it as a RoQ attack. At T3 we re-initiate the attack, and do not stop it for the rest of the experiment. When the POMDP controller has enough time to make an additional SNMP observation, it is able to register a RoQ attack at instant T4. The controller issues an attack response action at this instant, and commands the second AP to drop the users involved with the attack. These users are identified with their MAC addresses, which are not existent in the QoE database.

Congestion Case. For emulating a congestion scenario, we associate five users to AP2 sequentially. In Figure 5 we plot the delay experienced by three users. Congestion starts after T4, when the fifth user enters the system. Immedeately

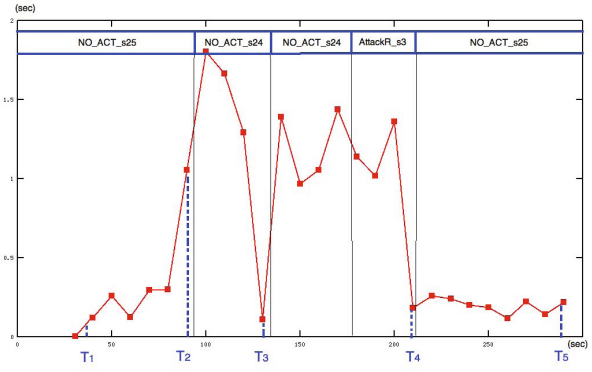


Fig. 4. Attack Scenario User Experience w.r.t Time

at T5 this user is handed over to AP1, as a result of load balancing action. Load balancing action is only taken if the SNMP list matched the QoE database list, meaning that with high probability there is no RoQ attack. In the scenario, the handover of a single user is not enough to increase the QoE levels in AP2. For this reason, the POMDP controller continues to make bad QoE observations. This leads the controller to handover yet another user at T6, after which the delay values stay in acceptable region.

Critical System Fail Case. We emulate a critical system failure in the observed AP2, by initiating a simple Linux shell based fork bomb, that drives the CPU load of the AP up to 100%, therefore making it unresponsive. For both RoQ attacks and system failures, the POMDP controller gets a bad SNMP observation. However in the case of system failure, the data traffic on the air interface is very low, since no client is able to reach the AP.

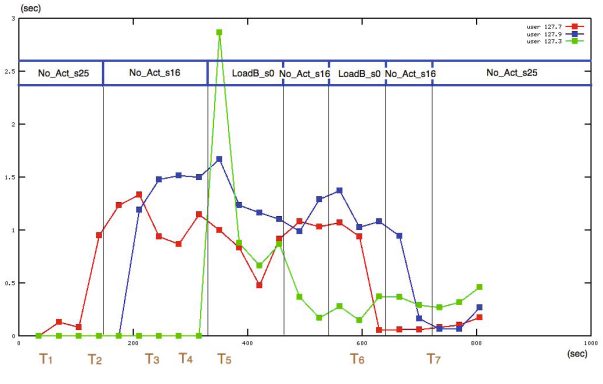


Fig. 5. Congestion Scenario User Experiences w.r.t Time

Similar to our attack scenario, we initiate two system failure emulations. At T1 a shorter duration system failure is started, which is stopped at T2 as depicted in Figure 6. The POMDP controller requires a second observation to make sure that there is a system failure. This second observation cannot be made in the first system failure. At T4, we re-initiate the fork bomb and let it run until the end of the experiment. At T5, POMDP controller detects a system failure and reports this to the system admin.

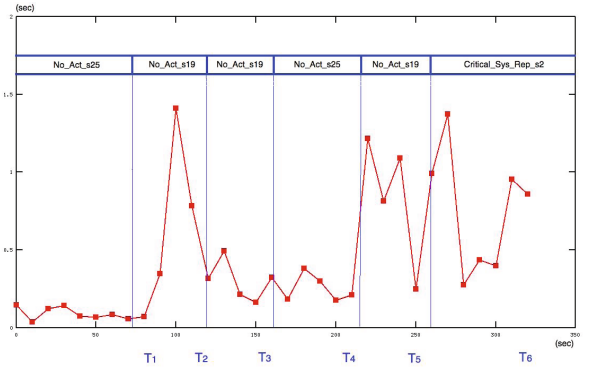


Fig. 6. Critical System Failure Scenario User Experiences w.r.t Time

5 Conclusion

In this paper, we have implemented an intelligent system for IEEE802.11 WLANs in order to differentiate and properly handle attack, congestion and critical system failure situations in the network. We have tested our proposed system with three scenarios representing those cases and observed the actions taken by the decision engine on the controller APs. We have observed that the decision maker reacted intelligently and acted in a protective manner keeping users' QoE in an acceptable region.

In this study, we focused on intelligent controllers with POMDP engines applicable for anomaly detection, specifically attack and congestion mitigation in network. We chose the technical use cases in IEEE wireless LAN however it is possible to use this approach in other controller-based networks and other technologies beyond IEE WLAN.

Basic observations are done for sensing attack or congestion conditions; however implementation framework can be easily equipped with more sophisticated methods. As part of our future work, we are planning to increase the complexity of the system by adding more observations with high processing capacity to the controller program. Moreover, we have assigned transition probability functions intuitively at this early stage of our research. We are in the process of introducing learning engines to the controller module for long term observations on the network and dynamically assigning the probability functions to optimize the decisions of the controller.

Moreover, we focused on a single controller agent, however, due to scalability issues, we plan to distribute our controller agents throughout the access points. As a future work, we are planning to implement intelligent negotiation protocols for user migration action in a multi agent environment in order to avoid ping pong effects, which may occur because of asynchronous user migration action for access points.

References

1. Peng, T., Leckie, C., Ramamohanarao, K.: Survey of network-based defense mechanisms countering the dos and ddos problems. ACM Comput. Surv. 39 (April 2007)
2. Guirguis, M., Bestavros, A., Matta, I.: Exploiting the transients of adaptation for roq attacks on internet resources. In: IEEE ICNP, pp. 184–195 (2004)
3. Singh, J., Gupta, S., Kaur, L.: A MAC Layer Based Defense Architecture for Reduction of Quality (RoQ) Attacks in Wireless LAN, Arxiv preprint arXiv:1002.2423 (2010)
4. Peng, T., Leckie, C., Ramamohanarao, K.: Proactively Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 771–782. Springer, Heidelberg (2004)
5. Kulkarni, A., Bush, S.: Detecting distributed denial-of-service attacks using kolmogorov complexity metrics. Journal of Network and Systems Management 14(1), 69–80 (2006)
6. Brickley, O., Rea, S., Pesch, D.: Load balancing for QoS enhance- ment in IEEE802. 11e WLANs using cell breathing techniques. In: IFIP MWCN (2005)
7. Yen, L., Yeh, T., Chi, K.: Load Balancing in IEEE802.11 Networks. IEEE Internet Computing, 56–64 (2009)
8. Vasudevan, S., Papagiannaki, K., Diot, C., Kurose, J., Towsley, D.: Facilitating access point selection in IEEE 802.11 wireless networks. In: ACM SIGCOMM, p. 26 (2005)
9. Yen, L., Yeh, T.: SNMP-based approach to load distribution in IEEE 802.11 networks. In: IEEE VTC, vol. 3, pp. 1196–1200 (2006)
10. Aleo, V.: Load distribution in IEEE 802.11 cells, MSc Thesis, KTH Royal Institute of Technology (2003)
11. Lee, M., Lai, D.: Enhanced algorithm for initial AP selection and roaming, uS Patent App. 10/228,668 (August 26, 2002)
12. Bertsekas, D.: Dynamic Programming and Optimal Control. In: Bertsekas, D. (ed.), vol. II. Athena Scientific, Belmont (1995)
13. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artif. Intell. 101, 99–134 (1998)
14. Botta, A., Dainotti, A., Pescape, A.: Multi-protocol and multi- platform traffic generation and measurement. In: IEEE INFOCOM, DEMO Session (2007)
15. Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., Singh, M.: Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In: IEEE WCNC, vol. 3, pp. 1664–1669 (2005)