Artificial Intelligence Driven Optimization of Channel and Location in Wireless Networks

Samurdhi Karunaratne†, Ramy Atawia, Erma Perenda, and Haris Gacanin
Nokia Bell Labs, Copernicuslaan 50, 2018 Antwerp, Belgium
Email: haris.gacanin@nokia-bell-labs.com

***Abstract*—In this work, we develop a framework that jointly decides on the optimal location of wireless extenders[1] and the channel configuration of wireless extenders and access points (APs) in a Wireless Mesh Network (WMN). Artificial Intelligence (AI) is adopted to support network autonomy and to capture insights on system and environment evolution. We propose a Self-X (self-optimizing and self-learning) framework that encapsulates both environment and intelligent agent to reach optimal operation through sensing, perception, reasoning and learning, in a truly autonomous fashion. The agent derives adequate knowledge from previous actions, improving the quality of future decisions. Extensive simulations are run to validate its fast convergence, improved throughput and resilience to dynamic interference conditions.**

***Index Terms*—Artificial intelligence, Machine Learning, Wireless networks, Optimization, Self-X design**

I. Introduction

Wireless Self-organizing Networks (Wi-SONs) have been proposed to proactively address different optimization challenges in dense wireless networks such as channel assignment, coverage, user control etc. [1] – [3]. In essence, Wi-SONs monitor network performance and calculate an optimal configuration to determine a new recommendation policy for single or clustered APs. This method, however, is deemed sub-optimal as it overlooks both internal and external network dependencies[2]. While most efforts in SON literature [1], [3], [4] have been directed to define cost functions with deterministic (rule-based) optimization schemes, the above dependencies have to be explicitly addressed. In contrast, Artificial Intelligence (AI) with Machine Learning (ML) techniques should be considered to enable wireless systems with learning and sophisticated decision-making [5].

The reinforcement learning scheme in [6]—designed for sensor networks—adopted random exploration and simple reward exploitation. This can be sufficient for the considered radio and power selection problem under the foreseen slow dynamics. However, channel assignment and learning in multi-radio WMNs comprise more states and dynamics which slow down the convergence of purely random exploration, and impact the optimality of simple reward functions that do not exploit problem structure. In [7], an Adaptive Dynamic Channel Allocation (ADCA) algorithm was proposed to pick the configuration that maximizes throughput and minimizes delay. Each pair of neighboring nodes negotiates to select a common link channel that maximizes throughput. However, the algorithm might perform sub-optimally in the case of saturated traffic and also overlooks neighboring non-managed interference (external interference).

Finally, there are optimization techniques adopting graph coloring, integer linear programming (ILP) or meta-heuristic techniques [8], [9]. The primary drawback of graph-coloring is its sensitivity to centralized knowledge, which usually fails to capture the granularity of inter-AP interference in non-managed scenarios. Although ILP techniques can reach globally optimal channel assignments, they fail to obtain real-time solutions in dynamic environments, and hence is not resilient. On the contrary, meta-heuristic techniques like Genetic Algorithm [9] and Tabu search [8] can provide near-optimal channel assignments that cope with dynamic environments, but their performance has not been tested in non-managed environments. CLICA [15] provides a channel assignment that guarantees connectivity and low inter-channel interference, but is also not designed to handle external interference in non-managed environments.

In this work, an AI-framework design is presented to support the network with autonomy to capture insights on system and environment evolution. We introduce heuristics to achieve near-optimal channel and location configurations, and its performance is validated through extensive packet-level simulations. Our main contributions are as follows:

1) We propose an AI-driven Self-X framework called Intelligent Channel Assignment and Location Optimization (ICALO) that comprises both environment and intelligent agent. The intelligent agent perceives the environment through network parameters and stores them in a knowledge base (KB) that guides learning and decision making. On the contrary, existing optimization strategies do not leverage the observed impact of previous actions while deriving future decisions.

2) A guided reinforcement learning (G-RL) approach is proposed with embedded domain knowledge to achieve user-aware self-optimization. The agent strikes a balance between exploration when learning has low cost, and exploitation when network performance is critical. Both perceived network states and the KB are used either to select or assess new optimal configurations and retain them in the KB. The agent is aware

†Now affiliated with University of Peradeniya, Sri Lanka.

[1]Wireless extenders are wireless devices used to extend the coverage area of a main access point; they are also called mesh points, since the main access point refers to a mesh gateway in a WMN.

[2]The internal dependency refers to the relation between configurations of the AP-extender-user set. The external dependency appears in multi-operator deployments due to the stochastic changes of neighbor APs adopting the same or overlapping channels.

978-1-5386-4920-6/18/$31.00 ©2018 IEEE

of the learning cost that interrupts user connectivity, and thus exploits spectral correlation to transfer knowledge among matching configurations.

II. Network Model

We model the network as a directed acyclic graph $G = (V, E)$, where *V* is the set of nodes and *E* is the set of bidirectional links (edges) between them. $v_i \in V$ represents either the gateway master Access Point (mAP), extender (EXT) or user device, where v_0 refers to the mAP, $v_1, \ldots, v_M$ represent the extenders, and $v_{M+1}, \ldots, v_{M+U}$ are user devices. We assume *N* available channels and a set of possible locations *L* for deploying extenders. We denote the set of radio interfaces for each node v_i by D_i, and the set of channels associated to radio-interfaces by C_i. Each link $e_{ij} \in E$ comprises of two nodes v_i and v_j, where v_j is connected to v_i and the latter provides the next hop communication to the mAP. Both nodes are in the transmission range of each other and they have at least one common channel assigned to their interfaces (i.e. $C_i \cap C_j \neq \varnothing$). We define h_{ij} as the channel associated with e_{ij} and thus, the link can be represented with the triple $e_{ij} = \{v_i, v_j, h_{ij}\}$.

A. System Variables

The system variables of our implementation model are described in the following.

1) Location-specific RSSI: The Received Signal Strength Indicator (RSSI) at receiver node v_j at location l_j from sink node v_i, $RSSI_{ij}^{(l_j)}$ represents a measured received signal strength in dBm of beacon frames received on the channel (defined as dot11BeaconRssi [10]).

2) Channel Utilization: The utilization $u_{h_{ij}}^{(l_j)}$ for channel h_{ij} at location $l_j \in L$ is calculated as $u_{h_{ij}}^{(l_j)} = \frac{\text{CBtime}(t+\tau) - \text{CBtime}(t)}{\tau} \times 100$, where CBtime$(\cdot)$ is the channel busy time in milliseconds provided by the Clear Channel Assessment stats counters.

3) End-to-end Throughput: Practically, R_k—the end-to-end throughput at user v_k—can be estimated as a user goodput based on transmitted and received bytes by the user within a measurement period Δt as $R_k = \frac{(\text{TXBytes} + \text{RXBytes}) \times 8}{\Delta t}$, where TXBytes and RXBytes, respectively, denote the total number of bytes transmitted and received.

B. Problem Formulation

We define the objective function of our approach as the total end-to-end throughput of all user devices, written as $\max_{C,L} \sum_{k=M+1}^{M+U} R_k$, where the search is done across a set of channels $C = \bigcup_{\forall i \in [0,M]} C_i$ and a set of locations $L = \{l_i | i = [1, M]\}$. Optimization of the objective function is done under the following constraints:

(a) Finite set of available channels – only N channels are available from which one could be assigned to a radio in a given node.

(b) Radio constraints – the number of channels assigned to a node cannot exceed the number of radios on that node. That is $\forall v_i \in V, |C_i| \leq |D_i|$, which means that the same channel can be assigned to different radios of v_i.

(c) Connectivity – two adjacent nodes v_i and v_j must have at least one channel in common $C_i \cap C_j \neq \varnothing$.

Below we present a heuristic algorithm with guided learning to solve the problem defined by the above objective function.

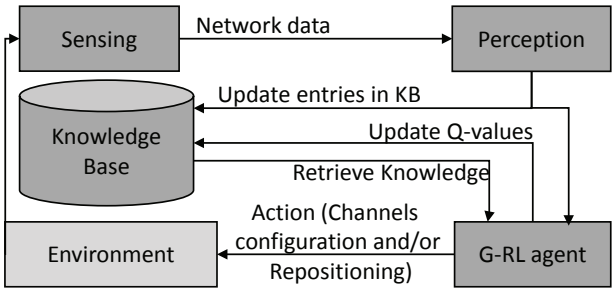


Fig. 1: AI-driven self-optimization framework.

III. Self-optimization Framework Design

The overall architecture of the proposed AI framework is summarized in Fig. 1, and comprises the Environment, the KB and their interaction with the Intelligent agent [12] through sensing, perception and reinforcement-learning (RL) [13]. The agent interacts with the environment by sensing the current state and then provides actions through reinforcement learning.

A. Sensing

In the sensing stage, the values of physical parameters used in the Perception stage may be collected from the mAP and EXTs through the TR-98/181 protocol for remote management [11]. This information is periodically collected from each node with a certain period τ in milliseconds.

B. Perception

The perception phase translates the sensed information from each node v_i into performance indicators (i.e. system variables) that identify the network state. The performance indicators are calculated for each radio $d \in D_i$ of node $v_i \in V, i \in [0, M]$ based on two successive sensing samples. One such indicator is the channel utilization (u_d) of the channel assigned to radio d.

By means of real-time network monitoring, the perception component is capable of detecting when the current configuration becomes sub-optimal, and sending a signal to the G-RL agent (defined below) to evaluate the current state of the network.

C. Reinforcement Learning

The guided RL (G-RL) agent utilizes Q-learning to select the optimal action at each state based on stored reward values (referred to as Q-values). In essence, the G-RL agent considers that each node $\{v_i | \forall i \in [0, M]\}$ has its own states and corresponding actions in that state, while the rewards are derived on the system level. The states, actions and rewards for each node $v_i \in V, i \in [0, M]$ are defined as follows:

States (S): Besides channels optimization, the G-RL agent aims to place each node v_i at an optimal location. Thus, the states $s \in S$ of each node will refer to its location l_i. Each node v_i has $|L|$ possible locations for deployment and hence $|L|$ possible states.

Actions (A): The G-RL agent takes two types of actions: channel configuration $A^{(c)}$, and EXT repositioning $A^{(l)}$, with action set $A(s) = A^{(c)} \cup A^{(l)}$. Since each node $\{v_i | \forall i \in [0, M]\}$ is equipped with D_i radios, we define channel configuration actions for that node as the set of all possible combinations of the radios, where $|A^{(c)}| = N^{|D_i|}$. On the other hand, each repositioning action $a \in A^{(l)}$ changes the location of node v_i and results in a state transition.

Reward (R): The instantaneous reward at time instant t in the state s for a selected action a at node v_i is given by $r_t(s, a, v_i) = \sum_{k=M+1}^{M+U} R_k$. We define the reward at the network level because applying an action a at node v_i impacts the performance of the whole network. In Q-learning, the cumulative reward $Q_t(s, a, v_i)$ is calculated using the previous Q-value and the instantaneous reward as given by

$$\begin{cases} Q_t(s, a, v_i) := Q_t(s, a, v_i) + \eta \Delta(s, a) \\ \Delta(s, a) = r_t(s, a, v_i) + \gamma \max_a Q_{t+1}(s', a) - Q_t(s, a) \end{cases} \quad (1)$$

where $Q_t(s, a)$ is the cumulative reward at state s when action a is applied at time t, carrying the system to state s' [13]. Parameters η and γ, respectively, are the learning factor and discount rate, both with values in $[0, 1]$. η controls the convergence speed of the learning and its value is gradually decreased in time to achieve convergence. The discount rate, γ, is used to weight the near-term rewards. Specifically, as γ approaches 1, the weight of future rewards is increased.

Policy (π): The selection of action a during a certain state s is governed by a policy $\pi(a|s)$. A policy that maximizes the cumulative reward $Q_t(\cdot)$ is denoted as π^*. Finding the optimal trade-off between exploration and exploitation is very challenging while deriving the policy, as it impacts both the learning cost and convergence rate [13].

D. Knowledge Base

The knowledge base stores three types of tables for each node v_i, $i \in [0, M]$.

Perception table: For each radio-interface $d \in D_i$, this table stores all the next hop nodes, the used channel h_d of each radio, and the channel utilization denoted by u_d.

Q-table: This table saves the Q-values for each possible action a in state s calculated by Eq. 1.

Channel-Location table: The channel utilization of all available channels N at all candidate locations L is kept, updated at each time slot.

With such a design of the KB, the G-RL agent is aware of the network topology and the current state of the system.

IV. Guided RL Agent Design

The RL agent is considered as both a learner and a decision maker. Thus, the agent has to balance between exploring the environment to gain more information, and exploiting the KB by picking decisions with a high likelihood to reach the optimal state. While the user experience during such learning and decision-making processes remains a priority, the RL agent has to be guided by domain experience to minimize the learning cost. To that end, problem-specific knowledge is used instead of random exploitation and exploration, to provide a user-aware decision at the right time. The agent is aware of the following domain knowledge:

Spectral Correlation: Overlapping channels in a Wi-Fi system[3] will typically have similar utilization factors since a given channel can be sensed busy due to transmission on the same or an overlapping channel. Thus, the exploration stage should pick non-overlapping channels, while overlapping channels are visited through exploitation.

Spatial Correlation: A Wi-Fi system that is suffering from a coverage problem typically cannot be optimized by re-configuring the channels, and thus prompts a change in the location of nodes (i.e. re-positioning EXTs). As such, distinguishing the coverage problem from contention and interference will help the agent to exclude channel re-configuration from the set of possible actions, and thus accelerate the learning process.

The main stages of ICALO are detailed as follows:

A. Selecting the Type of Action

Using perception data, the agent monitors the system performance by checking the changes in contention, interference and coverage levels at the current extender location. In particular, the $RSSI$ value on an EXT's backhaul—connecting EXT to mAP—is assessed versus a minimal threshold $RSSI'$ that achieves the target signal quality at the extender if the channel is optimized. In the case of poor coverage, channel exploration at such a location is unnecessary and thus a repositioning action must be selected. The new location is calculated as the midway between the current position of the extender and the next hop towards the mAP. If this location was visited before, then a random distance is added to the calculated midway location to provide exploration. The new location is stored in the channel-location table with the corresponding channel utilization of the last channel configuration.

In case of a high signal level, i.e. no coverage problem, the agent should explore and exploit using the channel configuration actions until no improvement is observed, and then a new location is selected.

B. Zero-Cost Knowledge-Driven Exploration

The second policy performs greedy exploration, yet with zero learning cost, since it is followed when 1) no users are associated or 2) the connected users are not requesting any traffic. In particular, the agent will pick a channel configuration action, compute its reward and store the cumulative reward in the Q-table to maximize the gained knowledge. As such, for every possible action a that is not applied before (i.e. with zero reward value in Q-table), the total Euclidean distance to all previously visited actions $i \in I$, is calculated by β_a as a

[3]Here, we consider 2.4 GHz band, but overlapping in 5 GHz band is observed by usage of dynamic channel bandwidth (20, 40, 80 and 160 MHz).

sum of Euclidean distances between action a and those actions in I. i.e. $\beta_a = \sum_{i=1}^{I} \sqrt{\sum_{d=1}^{D}(c_{a,d} - c_{i,d})^2}$, where $c_{a,d}$ is the channel assigned to radio d in action a. The optimal action, from an exploration perspective, is the one with maximum β_a.

In the case that all actions are visited (i.e. no zero entries in the Q-table), a random action is picked from the Q-table using a uniform distribution. This exploration process is repeated for every node $\{v_i | \forall i \in [0, M]\}$ until a connection or traffic request is received from a user device. By doing so, the G-RL agent accelerates the learning process of its environment without the degradation of user experience. After the agent applies this exploration action, the corresponding Q-value is updated in the KB, and the channel configuration is switched back to the former value.

C. Modified Basic Soft Max: Exploiting Spectral Correlation

In the case of perceiving interference or contention problems, the third policy is triggered. In essence, the third policy is defined based on Basic Softmax (BSmax) combined with Value-Difference Based Exploration (VDBE Softmax) [14], spectral correlation and the KB.

$$\pi(s) = \begin{cases} \text{Modified BSmax policy} & \xi < \varepsilon(s) \\ arg\max_{a \in A(s)} Q(s,a) & otherwise, \end{cases} \quad (2)$$

where ξ is a uniform random number over [0, 1], and $\varepsilon(s)$ is a state-dependent exploration probability. In essence, a high value of $\varepsilon(s)$ enables the agent to perform guided exploration, while a low value triggers exploitation by picking the action with maximum cumulative reward (i.e. Q-value).

1) Exploration Probability $\varepsilon(s)$: The state-dependent exploration probability $\varepsilon(s)$ is calculated iteratively as follows:

$$\begin{cases} \varepsilon_{t+1}(s) = \psi(s) f(s,a,\sigma) + [1 - \psi(s)]\varepsilon_t(s), \\ f(s,a,\sigma) = \frac{1 - e^{\frac{-|\eta\Delta(s,a)|}{\sigma}}}{1 + e^{\frac{-|\eta\Delta(s,a)|}{\sigma}}} \end{cases} \quad (3)$$

where σ and $\psi \in [0, 1]$, respectively, denote a positive constant called inverse sensitivity and the influence of the selected action on $\varepsilon(s)$. A reasonable setting for $\psi(s)$ is the inverse of the number of actions in the current state, $\psi(s) = \frac{1}{|A(s)|}$, since all actions should contribute equally to $\varepsilon(s)$. The parameter σ influences $\varepsilon(s)$ in such a way that low values allow high exploration even at small Q-value changes; high values of σ cause high levels of exploration only when the Q-value changes are large [14].

2) Action Selection: In the case of $\xi < \varepsilon(s)$, the G-RL agent is in exploration phase. The exploration phase takes five steps to pick a new action. First, the G-RL agent calculates for each $a \in A(s)$, the action selection probability $\rho(s_t = s, a_t = a, v_i) = min\{\rho_o(s,a,v_i), \rho_u(s,a,v_i)\}$ by using the BSmax probability $\rho_o(s,a,v_i)$ and the environment probability $\rho_u(s,a,v_i)$. The latter probability takes into account channel diversity, hidden node impact and contention impact caused by channel utilization and overlapping channels. $\rho_o(s,a,v_i)$ is determined using a Boltzmann distribution $\rho_o(s,a,v_i) = e^{\frac{Q(s,a)}{T}} / \sum_{b \in A(s)} e^{\frac{Q(s,b,v_i)}{T}}$, where T is a positive parameter called temperature. High temperatures cause all actions to be nearly equiprobable (more exploration), whereas low temperatures cause greedy action selections (more exploitation). $\rho_u(s,a,v_i)$ is determined as $\rho_u(s,a,v_i) = \frac{CD}{UI+HI+CI}$, where CD denotes the impact of channel diversity given as $CD = 1 + \sum_{d \in D_i} \sum_{d' \in D_i, d \neq d'} |h_d - h_{d'}|$ so that the action with the same channel tuned on all radio interfaces has the lowest probability. UI denotes the impact of channel utilization on the action given as $UI = \sum_{d \in D_i} u_{h_d}^{l_i}$. HI denotes the impact of hidden nodes and is defined as the difference between channel utilization observed on both sides of links that contains node v_i, multiplied by a factor 100: $HI = \sum_{j, e_{ij} \in E} |u_{h_{ij}}^{l_i} - u_{h_{ij}}^{l_j}| \times 100$. Finally, CI denotes the impact of contention from overlapping channels given as $CI = \frac{\sum_{d \in D_i} \sum_{h \in N, |h - h_d| \leqslant 5, h \neq h_d} (5 - |h - h_d|) u_h^{(l_i)}}{50}$. The second step is finding the maximal probability $\rho_{max} = \max(\rho)$ and on basis of it calculating the minimal allowed probability as $\rho_{min} = 0.9 \times \rho_{max}$.

In the third step, the G-RL agent finds all actions $A(s)'$ for which $\rho(s,a,v_i) > \rho_{min}$. Afterwards, the G-RL agent calculates $\kappa_{a\prime}$ for each $a\prime \in A'_s$ as $\kappa_{a'} = \left(1 + \sum_{d \in D_i} \sum_{d' \in D_i, d \neq d'} |h_{d,new} - h_{d',new}|\right) \times \sqrt{\sum_{d \in D_i} (h_{d,current} - h_{d,new})^2}$. The first factor denotes the channel diversity of the new action, and the second denotes Euclidean distance from the current applied action at node v_i. If the applied action does not satisfy perception thresholds, then it is highly likely that the actions with low Euclidean distance behave in the same way. However, the G-RL agent gives a higher probability to actions that have a higher Euclidean distance from the currently applied action. In the last step, the G-RL agent picks the action a^* that has the highest κ value.

D. Decision Making - Control Stage

After a new action is found, ICALO checks whether it knows anything about this action, i.e. whether the Q-value for this action is different from zero. In case that the Q-value is equal to zero, ICALO will apply the new action. Otherwise, it checks whether the Q-value of the new action is 15% higher than the Q-value of the currently applied action. This is because it is not worth applying a new action if it brings only a small improvement. By controlling the execution of actions in such a way, ICALO alleviates the issue of unnecessary network instabilities.

V. Performance Evaluation

To evaluate the proposed framework, we use the IEEE 802.11 compliant discrete-event network simulator ns-3. We consider scenarios where there is the mAP in conjunction with a single EXT and a variable number of client devices. The EXT is modeled as a node that has two radios—one, an adhoc mode interface that is used to establish backhaul communication with the mAP, and the second, an AP mode interface that is used to allow client devices to associate. The finite time required to relocate extenders in practice is not considered as it would have no qualitative impact on the

subsequent test results—in practice, this could run anywhere from a few seconds to a few minutes.

All subsequent tests were carried out with all the radios operating on the 2.4 GHz band and a channel width of 20 MHz. Packet size is set to 1000 bytes and transmission power of all radios is 12 dBm. The network area considered was 20 m $\times$ 10 m. We may consider it as an apartment of length 20 m and width 10 m, consisting of 8 rooms as given in Fig. 2(a).

In every test, we transmit a Constant Bit Rate (CBR) UDP data stream of 5 Mbps from the mAP to each of the client devices, and the ICALO parameters are set as: $\varepsilon_{EXT}(0) = 1$, $\varepsilon_{mAP}(0) = 1$, $T = 50$, $\sigma = 100$, $\psi_{EXT} = \frac{1}{121}$, $\psi_{mAP} = \frac{1}{11}$, $\eta = 0.7$, $\gamma = 0$, $\tau = 2$, $u_{thr} = 60(\%)$, $RSSI' = -60$ dBm.

A. Speed of convergence to steady-state throughput

To demonstrate ICALO's speed of convergence to the steady-state throughput, we consider a family of five living in the apartment, as visualized in Fig. 2(a). Then we introduce a single external (non-managed) node to act as external interference to our network and conduct 50 independent tests where the EXT is placed at a random location within the apartment in each test. The fronthaul (FH) and backhaul (BH) channels of the EXT is set to 3 and 7, respectively, while the external AP is in channel 3. A plot of the Cumulative Distribution Function (CDF) of the convergence times so obtained is given in Fig. 2(b). From this, we can observe that for 90% of tests, the convergence time is less than 36 s. All tests converged within 43 s. The mean convergence time was 23.6 s and the standard deviation was 7.9 s. Analysis of configuration changes (location changes plus channel changes) until steady-state revealed a mean number of configuration changes of 9.4 and a standard deviation of 2.5.

B. Comparison of steady-state throughput to state-of-the-art schemes

In this section, we compare the steady-state throughput of ICALO with that of three other channel assignment approaches—namely single channel assignment, Common Channel Assignment (CCA) [16] and Connected Low Interference Channel Assignment (CLICA) [15]. In single channel assignment, we consider the channel that produced the highest throughput. In CCA, we assign a random couple of orthogonal channels to the extender's FH and BH in each test. In CLICA, we consider the physical model presented in [15].

We conduct two similar experiments. In Experiment 1, there are two people in the dining room and one each in the living room, the storeroom and the bathroom (blue pentagons). Four external APs are introduced outside the apartment with operating channels of 8, 4, 1 and 2 as indicated by Fig. 3(a). The BH and FH channels of the EXT are set to 2 and 5, respectively, and the EXT is placed at 50 random locations within the apartment; for each such location, the system is allowed to reach a steady-state. The CDF of the steady-state throughput for each algorithm is plotted in Fig. 3(b).

As expected, single channel assignment has the worst overall performance. CCA performs much better as it eliminates (in our tests) inter-channel interference by choosing orthogonal channels. However, even in this case, it has zero sense of external interference and is inferior to CLICA. CLICA performs better than both the first two approaches, and in some cases, matches the performance of ICALO. But any single formula (as used in CLICA to estimate channel conflicts) is unlikely to fully capture both external and internal interference effects accurately. This is where the exploratory phase of ICALO comes into effect and results in increased performance.

In Experiment 2, we follow the same procedure except that the client devices are relocated to locations denoted by green triangles and the channels of the external APs are changed to 9, 1, 3 and 7 as shown in Fig. 3(a). Also, the FH and BH channels of the EXT, respectively, are changed to 3 and 7. The corresponding results are shown in Fig. 5. It can be seen that the same general trend as in Fig. 3(b) is present here as well.

C. Resilience of ICALO to dynamic network conditions

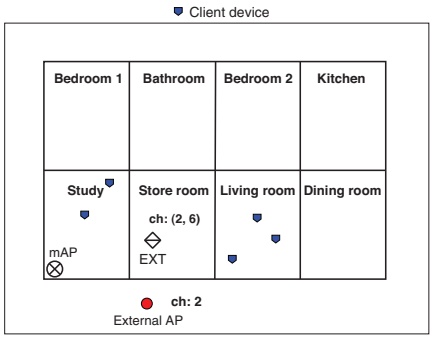
To verify the ability of CLICA to recover from the effects of dynamic network conditions, we simulate the scenario illustrated by Fig. 4(a): starting with AP1 activated, remove and activate each of the APs, {AP1, AP2, AP3, AP4} one by one, pausing for the system to reach a steady-state before removing the current AP and activating the next. Continuing in this order, finally only AP4 is left activated. AP1, AP2, AP3 and AP4 transmit at channels 3, 10, 4 and 8, respectively.

The per-user throughput variation versus time for this scenario is given in Fig. 4(b). The moments at which the system reaches the steady-state is marked by dot-dashed lines (green) and the moments at which the current external AP is removed and the next one is activated are marked by dashed lines (red).

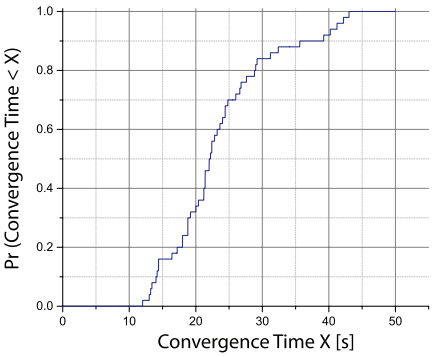
It can be seen clearly that the network reaches near-optimal throughput at each stage after successfully recovering from the decline in throughput due to a sudden change in external interference conditions. Note also that successive convergence times decrease as 28 s, 16 s, 12.5 s and 10 s. This reflects the effect of the growing knowledge base. Following this pattern, as the system evolves, we can ideally expect ICALO to make optimal decisions with little lag.

VI. Conclusion and Future Works

This paper presented ICALO, a self-optimization scheme for wireless extenders in a WMN which adopts an AI-driven learning framework. Our results show significant throughput improvements over several other channel assignment approaches while converging to peak-performance much faster and with a lower number of actions than an unguided RL approach. We also portrayed the resilience of ICALO by demonstrating its ability to quickly recover from throughput degradation caused by sudden changes in the network environment. We relate this performance to the guidance of the reinforcement learning agent by using domain knowledge, to curb unnecessary exploration while fostering smarter exploitation. We conclude that ICALO successfully addresses the problem of joint channel assignment and location optimization of WMNs by guaranteeing low-cost learning and achieving near-optimal network configurations.

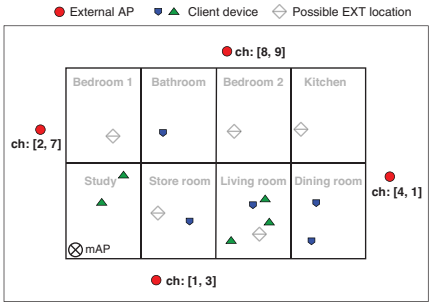


(a) Simulation environment for demonstrating convergence

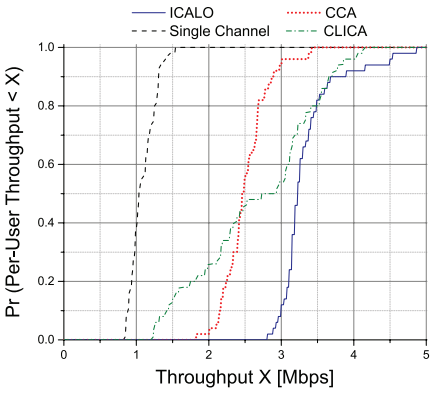


(b) Convergence times of ICALO for 50 tests

Fig. 2: Convergence of ICALO to steady-state throughput

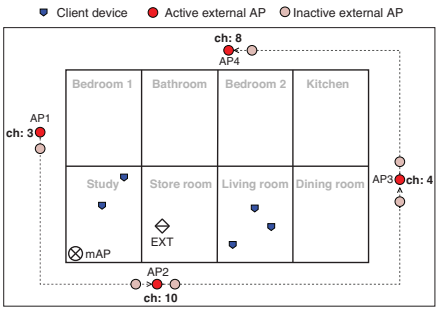


(a) Simulation environment for Experiment 1&2

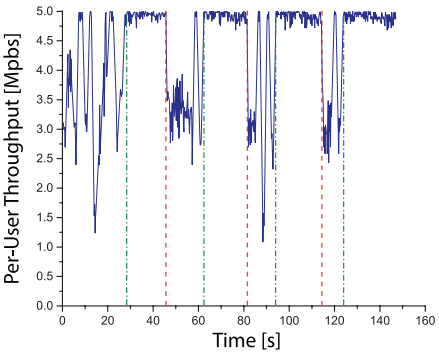


(b) Results of Experiment 1

Fig. 3: Experiment for comparing ICALO performance with single channel, CCA and CLICA



(a) Simulation environment for experiment



(b) Results of experiment

Fig. 4: Recovery of per-user throughput in dynamic network conditions by ICALO

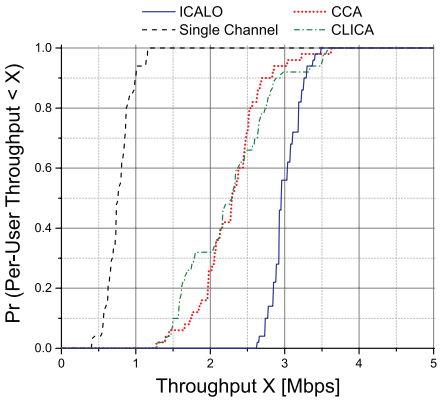


Fig. 5: Results of Experiment 2 for comparing ICALO performance with single channel, CCA and CLICA

The basic idea of the proposed ICALO framework is validated for a 2-hop network with multiple stations by considering only the 2.4 GHz band. However, as a future work, we aim to thoroughly evaluate the performance of ICALO in more complex scenarios such as in WMNs with multiple extenders with dual-band radios (2.4 GHz and 5 GHz); furthermore, the authors also intend to improve the scalability of ICALO.

REFERENCES

[1] H. Gacanin and A. Ligata, "Wi-Fi SON: Challenges and Use Cases," IEEE Communication Magazine, Network & Service Management Series, July 2016.

[2] Wi-Fi Alliance, *Multi-AP Technical Specification*, draft, Jan. 2018.

[3] S. Chieochan, E. Hossain, and J. Diamond, "Channel assignment schemes for infrastructure-based 802.11 WLANs: A survey", IEEE Communications Surveys & Tutorials, vol. 12, no. 1, pp. 124-136, 2010.

[4] A. B. M. Alim Al Islam, Md. Jahidul Islam, Novia Nurain, Vijay Raghunathan, "Channel Assignment Techniques for Multi-Radio Wireless Mesh Networks: A Survey," IEEE Communications Surveys and Tutorials 18(2), pp. 988-1017, 2016.

[5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K-C. Chen and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," IEEE Wireless Communications Magazine, Vol. 24, No. 2, April 2017.

[6] J. Gummeson, D. Ganesan, M. D. Corner and P. Shenoy *An adaptive link layer for range diversity in multi-radio mobile sensor networks* IEEE INFOCOM, pp. 154–162, 2009.

[7] Y. Ding, K. Pongaliur, and L. Xiao *Channel allocation and routing in hybrid multichannel multiradio wireless mesh networks* IEEE Trans. Mobile Comput., vol. 12, no. 2, pp. 206218, Feb. 2013.

[8] A. Hertz and D. de Werra. *Using Tabu Search Techniques for Graph Coloring* Computing, vol. 39, no. 4, 1987.

[9] A. Kenneth, D. Jong. *Evolutionary Computation: A Unified Approach* MIT Press, 2006.

[10] *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* IEEE.

[11] Broadband Forum, "TR-181 Device Data Model for TR-069 protocol", Issue 2, 2010.

[12] S. Russell and P. Norvig, *Artificial Intelligence A modern approach*, Prentice-Hall, 1995.

[13] R. Sutton and A. Barto *Reinforcement Learning: An Introduction.* Cambridge, MA, USA: MIT Press, 2012.

[14] M. Tokic, G. Palm. *Value-Difference Based Exploration: Adaptive Control between Epsilon-Greedy and Softmax.* KI, volume 7006 of Lecture Notes in Computer Science, page 335-346. Springer, (2011).

[15] M. Marina, S. Das, "A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks," Proceedings of Broadnets, 2005.

[16] A. Adya, P. Bahl, J. Padhye, A. Wolman, Lidong Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," Proceedings of BroadNets, 2004.