This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

Self-optimization of Wireless Systems with Knowledge Management: An Artificial Intelligence Approach

Haris Gacanin, *Senior Member, IEEE*, Erma Perenda, Samurdhi Karunaratne and Ramy Atawia
Nokia Bell Labs, Nokia Corp.
Copernicuslaan 50, 2018 Antwerp, Belgium
Email: name.lastname@nokia-bell-labs.com

***Abstract*—In this paper, we propose a new concept of a knowledge management framework to enable a self-optimizing and self-learning for wireless system operation in real time. The framework 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. Domain experience was provided to guide the agent while exploring and exploiting the set of possible actions in the environment. Thus, it guarantees low-cost learning and achieves a near-optimal network configuration addressing the non-deterministic polynomial-time hardness problem of joint channel and location optimization in a wireless system. Extensive simulations are run to validate its fast convergence, high throughput, and resilience to dynamic interference conditions. We deploy the framework on off-the-shelf wireless devices to propose autonomous self-optimization with knowledge management.**

***Index Terms*—Artificial intelligence, learning, wireless.**

I. Introduction

Wireless self-organizing network (Wi-SON) has been proposed to pro-actively address different optimization challenges such as channel assignment, coverage, user control, etc., in dense deployments [1]. In essence, Wi-SON is monitoring network performance and calculating an optimal configuration to determine a new recommendation policy on single or clustered access points (APs). This method, however, is deemed sub-optimal as it overlooks both internal and external network dependencies. The internal dependency refers to the relation between configurations of the AP, extender (EXT) and user terminal set (e.g. the optimally of channel assignment depends on the location of AP, extender and end user). The external dependency appears in multi-operator multi-access deployments [2] due to the stochastic changes of neighbor configurations adopting the same or overlapping channels. While most efforts in SON literature [3], [4], [5] have been directed to define cost functions with deterministic (rule-based) optimization schemes, the above dependencies have to be explicitly addressed.

Existing multi-hop optimization strategies does not explore neighbouring information of the non-managed wireless system, ignore real-time performance tracking, and does not leverage the observed impact of previous actions while deriving future decisions. In contrast, artificial intelligence (AI) methodology, *sensing, reasoning, active learning, and knowledge management*, should be considered to enable wireless systems with learning and sophisticated decision-making [6]. To that end, we envision a truly autonomous wireless network that is capable of sensing and perceiving its neighborhood to learn network dependencies, build the necessary knowledge and enable its constituent nodes to reason out the optimal configuration [4]. Such a design leads to the self-sensing, self-optimizing and self-learning (self-X) space that allows nodes to adapt its goals based on sensed user activities.

In this work[1], a knowledge management framework by means of AI methodology is presented to support the autonomous operation. The framework enables wireless systems to capture insights on its own and its environment evolution. We demonstrate efficient convergence times, and verify its superiority over the state-of-the-art, before portraying its adaptability to dynamic network conditions. Our main contributions are as follows:

- We propose an AI methodology inspired optimization framework with knowledge management called Intelligent Channel Assignment and Location Optimization (ICALO) that comprises both environment and intelligent agent. The environment includes managed APs, user devices, and multi-radio wireless extenders, all modeled by a directed acyclic graph. The model considers the correlation between location and channel configurations to optimize an end-to-end user performance capturing the states of all links constituting the path from AP to a user. The intelligent agent perceives the environment by network parameters and stores them in a knowledge base (KB) that guides the learning and decision making.
- 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 KB are used either to select or assess new optimal configurations and retain them in the KB. The agent is aware of the learning cost that

[1]The preliminary results appeared in our recent work [31].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

interrupts user connectivity, and thus exploits spectral correlation to transfer knowledge among matching configurations.
- We prove that our problem is NP-hard and introduce heuristics demonstrating the effectiveness of the so produced AI-driven self-optimization. The performance is validated through extensive packet-level ns-3 simulations and an experiment with commercial off-the-shelf (COTS) access points (APs) testbed.

This work is the first demonstration of fully AI-driven operation in simulation and testbed, by means of sensing, reasoning and active learning, of a wireless system with knowledge management supporting self-X in real time.

II. Related Works

One comprehensive survey on channel assignment in multi-radio wireless mesh networks (WMNs) classifies different techniques based on the type of decision making, network dynamics, granularity, communication layers and optimization methods [5]. The decision making can be either (i) centralized—maintaining awareness of the network topology and state—or (ii) distributed, failing to maintain connectivity. A dynamic channel assignment, compared to a static one, provides a robust solution that is aware of configuration changes due to users' mobility and reconfiguration of neighboring APs. The granularity of the channel assignment is defined either at the link-level or flow-level. The former assigns the same channel to two nodes to maximize the throughput of their inter-connecting link. The latter assigns the same channel to all nodes on the flow from the source to destination. In this way, end-to-end performance is optimized without exploiting multi-radios, in which the flow can involve multiple channels while maintaining connectivity through a common channel between every two neighboring nodes. In addition, the inter-dependence such as those between links of the same flow, and between the radios of the same node, were ignored. Cross-layer channel assignment (e.g. network, data link and physical) provides globally optimal performance, but updating routing tables and channels [10] is practically infeasible with off-the-shelf devices. In addition, the neighboring interference and real-time measurements that assess network connectivity are overlooked.

The reinforcement learning scheme in [8]—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 optimal reward functions that do not exploit problem structure. In [9], an Adaptive Dynamic Channel Allocation (ADCA) algorithm was proposed to pick the configuration that maximizes throughput and minimizes the delay. Every two neighboring nodes negotiate to select their common link channel that maximizes the 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 [12], [11]. 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 can provide near-optimal channel assignments that cope with dynamic environments, but their performance was not tested in non-managed environments. Genetic Algorithm [11] and Tabu search [12] are considered as quasi-static searching algorithms, but they do not provide good performance in dynamic non-managed environments. CLICA [13] provides a channel assignment that guarantees connectivity and low inter-channel interference, but it also is not designed to handle external interference in non-managed environments. The methodology for self-deployment was presented to increase the chance of reaching an optimal position of extenders at low searching and learning costs in [14].

A. Challenges in Practice

The channel assignment schemes above neglect the following practical aspects:

1) Neighbouring network interference: As a CSMA-based system, a target wireless station suffers from both exposed-node and hidden-node problems. The former refers to the contention due to neighboring nodes with high received power, operating on overlapping channels—causing busy channels and delaying transmissions. On the contrary, hidden nodes will cause packet collision at the receiver due to the mutual transmission of stations outside the sensing range of each other. However, calculating the exact amount of interference and/or contention is very challenging, as the traffic profiles of non-managed neighbors are not readily available and cannot be directly predicted.

2) Dynamics of re-positioning: While users have the flexibility to re-position extenders, the sources of dynamics should be extended beyond user devices to include extender locations as well. Thus, a natural need arises for dynamic optimization approaches to cope with the evolution in network topology, user association, and radio conditions. Such approaches should jointly solve for both channel and location of extenders to avoid positions where channel assignment is very challenging (e.g. due to excessive contention), possibly where no channel assignment is likely to offer satisfactory end-user experience. Additionally, it will mitigate the burden of moving the extender from an optimal location because of a poor channel configuration.

3) Learning Cost: Both neighboring interference and network dynamics are captured through measurements performed by APs and extenders, and thus typically require channel switching and extender re-positioning. Both, however, will increase the learning cost due to the service downtime due to the re-association process, and the physical movement to re-position the extender. Ignoring this cost will result in poor customer experience and increased user complaints.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

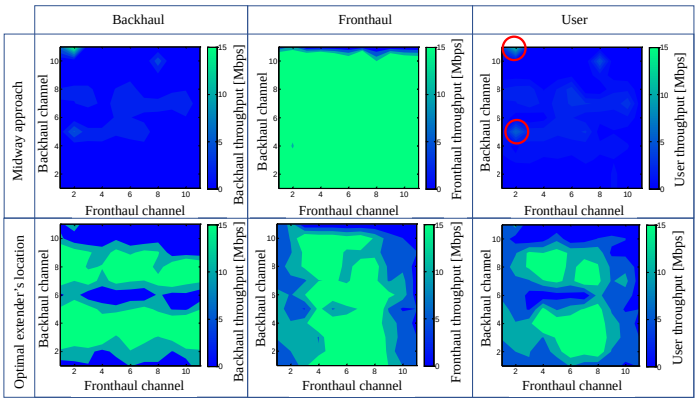
B. Motivation for AI

Figure 1 illustrates a laboratory experiment of a 2-hop WMN with a single user connected to an AP through one extender placed within a very busy office environment (i.e. surrounded by a lot of non-managed neighboring APs). Both, AP and the extender are equipped with two 2.4 GHz radios. We denote the link between AP and extender by backhaul, while the link between a user device and extender by fronthaul. Here, the end-user throughput is determined by the minimum of the backhaul and fronthaul throughput as $R_u^{(l)} = \min(R_b^{(l)}, R_f^{(l)})$. Thus, to maximize the end-user throughput the one needs to take into consideration the combinations of both backhaul and fronthaul achievable throughput values. The throughput at the backhaul, fronthaul and user device at two different extender locations $l \in \{l_1, l_4\}$, is measured and denoted by $R_b^{(l)}$, $R_f^{(l)}$ and $R_u^{(l)} = \min(R_b^{(l)}, R_f^{(l)})$, respectively. At the midway location l_4 between mAP's location l_0 and end-user device location l_2 in Fig. 1), the user throughput $R_u^{(l)}$ is maximized by selecting the channel combinations 2 and 11 or 2 and 5 for fronthaul and backhaul, respectively. This is indicated by circles in Fig. 1(a). Such a location is said to be sub-optimal for the backhaul as it suffers from high interference and/or low coverage. On the contrary, at location l_1, R_b is optimized over a wide range of channel combinations while R_f is maximized over a more tighter range of optimal channel combinations and do not include channels 2 and 11 that were deemed optimal at the first location. The channel combination of 6 and 4 for fronthaul and backhaul, respectively, is optimal when the extender is located at location l_1. As such, changing the location of extender typically alter the possible optimal channel combinations.

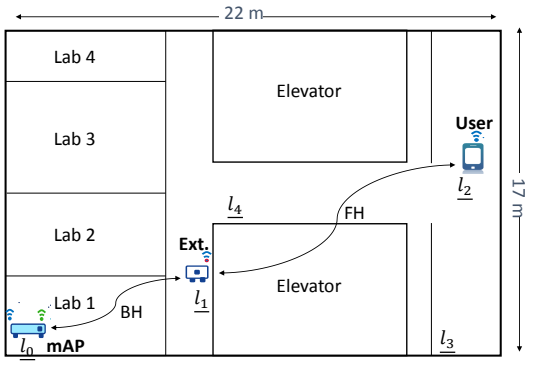
The network should be aware that the deployment is sub-optimal [14] in the first location (i.e. midway location) and performance improvement through channel assignment is not attainable, while the likelihood of reaching an optimal channel combination is very high in the second location. An unprecedented challenge is reducing the cost of learning which was very high at the first location due to the poor backhaul. As such, a delay of up to tens of seconds was experienced to collect backhaul measurements, followed by an extra delay of a few seconds to re-associate the user with the extender and the extender with the AP. Addressing these challenges is the goal of this work.

III. Problem Formulation

We consider a multi-radio multi-channel WMN with a single master AP (mAP), multiple neighboring APs and wireless extenders (EXTs) whose locations change over time. The mAP is a gateway with wired backhaul which provides access to the Internet, while EXT act as relays to expand the wireless service region of the *mAP*. The extender and mAP are equipped with a number of radio interfaces, where each radio is operating on a pre-defined channel. The network is serving user devices that are connected to the mAP either directly or through the extender.



(a) Measurements of channel and location coupling.



(b) A layout of tested environment: BH and FH denote backhaul and fronthaul, respectively.

Fig. 1: Motivation of optimization and placement problem wireless system with extender.

A. Graph Network Model

We model the network as a directed acyclic graph $G = (V, E)$, where V is the set of nodes v_i and E is the set of bi-directional links (edges). For the sake of theoretical generalization, the network includes a single *mAP*, M extenders and U user devices. $v_i \in V$ represents either *mAP*, 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 L possible locations 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. Further, we denote D as a union of the radio interface sets, i.e. $D = \bigcup_{i=0}^{M} D_i$.

We define the kth user path $p_k = \{e_{ij} \mid i, j \in [0, M+U];\ i \neq j\}$ as a set of distinct links $e_{ij} \in E$ connecting *mAP* v_0 and the kth user node v_k ($k \in [M+1, M+U]$ denotes user index). We constrain two successive links e_{ij} and e_{nm} in path p_k by setting $j = n$. The set of nodes forming the links of path p_k must contain only one node each with index 0 and index $k \geq M+1$. Thus, 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. $C_i \cap C_j \neq \varnothing$). The connected nodes v_i and v_j with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

their respective radios $d \in D_i$ and $d^{'} \in D_j$, share the same operating channels h_d and $h_{d'}$, respectively. For example, the link is represented with a triple $e_{ij} = \{v_i, v_j, h_d\}$. Each node $v_i \in V$ $(i = [1, M])$ is capable to sense a set of managed and non-managed neighbors $\mathcal{N}_i$. For any pair of links e_{ij} and e_{km} we say that e_{ij} and e_{km} are mutual neighbors (and interfere) if there exists a configuration (v_i, v_j, v_k, v_m) such that two of the four nodes composing e_{ij} and e_{km} belong to different links and can sense each other. Thus $\mathcal{N}_{ij}$ is the set of neighbors of link e_{ij}. Although the neighborhood relationship is symmetric, i.e. $e_{ij} \in \mathcal{N}_{km} \Leftrightarrow e_{km} \in \mathcal{N}_{ij}$ it does not imply symmetric interference levels.

B. Objective Formulation

We define the objective function as the total end-to-end user sum throughput, written as

$$\max_{C,L} \sum_{k=M+1}^{M+U} R_k, \tag{1}$$

where R_k is defined as the throughput between the end device and *mAP* as described in Section IV-A. The optimization search is done across a set of channels $C = \{C_i \mid i = [0, M]\}$ and a set of locations $L = \{l_i \mid i = [1, M]\}$ that lead to optimal network configuration for each path $p_k, k = [M + 1, M + U]$ as defined in Section IV-A.

The optimization of the objective function is done under the following constraints:

(a) Finite set of available channels – the set of channels that can be assigned to any node is N.
(b) Channel-radio relationship – to each radio can be assigned only one channel. That is $\forall v_i \in V, card(C_i) = card(D_i)$, where $card(\cdot)$ denotes the cardinality of a set.
(c) Radio constraints – the number of channels assigned to one node cannot exceed the number of radios on the node. That is $\forall v_i \in V, \bar{C}_i \leq \bar{D}_i$, where $\bar{\cdot}$ denotes the number of distinct elements in a set—which means that the same channel can be assigned to different radios of v_i.
(d) Connectivity – two adjacent nodes v_i and v_j must have at least one channel in common $C_i \cap C_j \neq \varnothing$.

C. NP-hardness of Joint Channel and Location Search Problem

Next, we present the computational hardness property of the above defined objective function by the following Lemma.

Lemma 1. Joint channel assignment and location optimization in WMNs possess the non-deterministic polynomial-time hardness (NP-hard) property.

Proof of Lemma 1 is given as follows.

Proof: Under the assumption that the location of each node $v_i \in V$ is already determined, and given the neighboring environment, then one sample of our problem can be described as $V = \{v_0, v_1, \ldots, v_{M+U}\}$. We assume $E = \{e_{ij} \mid t_c(v_i, v_j) \neq \varnothing\}$, where $t_c(v_i, v_j)$ denotes the channel constraints matrix for the nodes v_i and v_j. For example, the channel constraints matrix contains the connectivity constraints mentioned above. Defined in such a way, $G = (V, E)$ presents an instance of an NP-hard coloring problem [27]. An optimal coloring of G given by $C \times V \rightarrow \{1, \cdots, X(G)\}$ is also an optimal channel assignment for the set V under the channel constraints matrix, already given a set of extenders' locations and a static environment. Other set of extenders' locations and other instances of the environment might result in different $X(G)$. $X(G)$ denotes the minimal number of colors necessary to color the nodes of G such that no two adjacent nodes receive the same color. In the coloring problem, the coloring is equivalent to channel assignment, thus a color means a channel index. On the other hand, if $G = (V, E)$ is an instance of the coloring problem and we let $V' = \{v'_0, v'_1, \ldots, v'_{M+U}\}$ and $t_c(v'_i, v'_j)$, where $t_c(v'_i, v'_j) = \{0\}$ if $\{e'_{ij}\} \in E$ or $t_c(v'_i, v'_j) = \{\}$ if $\{e'_{ij}\} \notin E$ ($\{0\}$ denotes non-empty set). Now, if an optimal channel assignment for V' is given by $C' \times V' \rightarrow \{1, \cdots, \min(V', t_c)\}$, then C' is also an optimal coloring for G, i.e. $X(G) = \min(V', t_c)$ [28]. Here, t_c is the new channel constraint matrix and $\min(V', t_c)$ is a minimum-order channel assignment for V'.

Since the formulation of our self-optimization problem is equivalent to the coloring problem (with constraints of the static environment and given EXT locations), we deduce that the defined problem is NP-hard. ■

We note here that, unlike the coloring problem formulation given in the proof, our problem considers a fully dynamic neighboring environment and search for an optimal configuration set of channel and location. Hence, below we present a heuristic algorithm with guided learning to achieve a near-optimal configuration.

IV. Self-optimization Framework Design

A key aspect of self-optimization is the autonomy, in which the network can configure both the mAP and extenders without manual troubleshooting or instructions by the operator's help desk [4]. The network is typically modeled as two main elements: Environment and Intelligent agent. The former consists of managed wireless system (master and extender nodes) and non-managed neighboring APs. Unlike supervise learning [19] and deep RL [20], we design an agent by the principle of reinforcement learning (RL) to interact with the environment by sensing the current state and then, decide upon an action [21]. The *intelligent agent* perceives the environment through a sequence of *sensing*, *reasoning* and *acting* in order to build its own knowledge and use it in future actions [18]. The agent evaluates the actions based on a reward, which is a function of the resultant network state. Thus, *good* actions, e.g. achieves the QoS levels, can be reused directly in future when similar network conditions are sensed, while *bad* actions, e.g. creates coverage holes, will be used to refine the searching strategy of the agent. It stores the perceived states and rewards of each action in a knowledge base that can be utilized to improve the quality of future decisions.

The overall architecture of the proposed AI framework is summarized in Fig. 2, and comprises the environment, the KB and their interaction with the agent: sensing, perception and reinforcement-learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

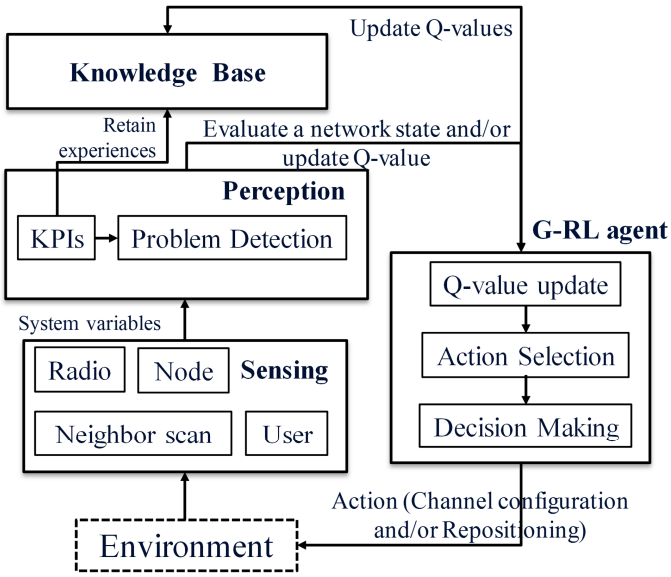


Fig. 2: AI inspired self-optimization framework with knowledge management.

A. Sensing

In the sensing stage, the values of physical parameters that can be used to describe the wireless system may be collected from the *mAP* and EXTs (i.e. from each $v_i \in V, i \in [0, M]$) through the TR-98/181 protocol for remote management [17] or through another interface defined by the device software development kit. The collected data contains radio-interface level statistics (e.g., the indices of used channels, Clear Channel Assessment statistics counters such as channel busy time etc.) and user-device level statistics (e.g. RSSI, counter values for total number of re-transmissions, failed packets, sent packets, sent and received bytes etc.). This information will allow the agent to perceive the environment, detect its current state and assess the performed actions. The sensing stage collects the data from each node with a certain period τ in milliseconds.

System Variables: The system variables are described as follows:

1) Location-specific RSSI: The Received Signal Strength Indicator (RSSI) at receiver node v_j at location l_j from sink node v_i, $RSSI_{ij}^{(l_j)}$, is a signal strength in dBm of beacon frames received on the channel (i.e. defined as dot11BeaconRssi [15]). RSSI is usually measured during the reception of the physical (PHY) preamble and its value is forwarded to the Medium Access Control (MAC) layer in the RXVECTOR [15]. Beacon's RSSI may be averaged over time using a vendor specific smoothing function. In case that the beacon frame is received by means of multiple receive chains, the RSSI is averaged in linear domain over all chains. The RSSI value range is -100 dBm to 40 dBm [15].

2) Channel Busy Time: Channel Busy Time, $CBtime$ denotes a period in milliseconds during the radio's lifeline when the operating channel is sensed as busy. It is measured during Clear Channel Assessment procedure [15].

3) Link Throughput: We define a throughput of the link e_{ij} at receiver node v_j placed at location $l_j \in L$, as R_{ij}. The maximal link throughput is obtained as follows [15]:

$$R_{ij}^{max} = \min\left[\log_2\left(1 + 10^{\frac{RSSI_{ij}^{(l_j)} + P_{adjust}}{10}}\right), maxBPS\right] \times \frac{maxNSS}{PPDU} \times N_{OFDM}, \quad (2)$$

where P_{adjust} is the implementation specific power adjustment parameter in dBm taking into account potential transmit power differences between Beacon/Probe response frames to data frames; $maxBPS$ denotes a maximum number of bits per second which is equal to: 40/6 if 256-QAM 5/6 modulation is allowed in the link, 6 if 256-QAM 3/4 modulation is allowed in the link or to 5 otherwise [15]; $maxNSS$ is the maximum number of spatial streams; N_{OFDM} denotes the number of OFDM sub-carriers and $PPDU$ is the duration of one physical protocol data unit payload symbol in seconds [15]. The link throughput value is calculated for transmit and receive modes whose values are stored as an L_DATARATE parameter within TXVECTOR and RXVECTOR primitives [15]. For example, these values can be obtained through Broadband Forum Technical Report (TR)-181 specification as *InternetGatewayDevice.LANDevice.*$\{i\}$*.WLANConfiguration.* $\{i\}$*.AssociatedDevice.*$\{i\}$*.X_BL_TxRate* and *InternetGatewayDevice.LANDevice.*$\{i\}$*. WLANConfiguration.*$\{i\}$*.AssociatedDevice.*$\{i\}$*.X_BL_RxRate* [17]. The maximum link throughput is multiplied by the percentage of time the medium is is sensed as idle at radio interfaces $d \in D_i$ to obtain the link throughput given by

$$R_{ij} = R_{ij}^{max} \times (100 - u_d^{(l_j)}). \quad (3)$$

where $u_d^{(l_j)}$ denotes the channel utilization defined next.

4) End-to-end Throughput: Computationally, the end-to-end throughput of the kth user device is defined as

$$R_k = \min\{R_{ij} \mid e_{ij} \in p_k\}. \quad (4)$$

On the other hand, R_k can be practically estimated as a user goodput in bits by using transmitted and received bytes by the user within a measurement period Δt as

$$R_k = \frac{(TXBytes + RXBytes) \times 8}{\Delta t}, \quad (5)$$

where $TXBytes$ and $RXBytes$, respectively, denote the total number of bytes transmitted and the total number of bytes received. These values are available through specific vendor extensions (e.g. statistics counters *InternetGatewayDevice.LANDevice.*$\{i\}$*.WLANConfiguration.*$\{i\}$*.AssociatedDevice.*$\{i\}$*.Stats.BytesSent* and *InternetGatewayDevice.LANDevice.*$\{i\}$*.WLANConfiguration.*$\{i\}$*.AssociatedDevice.*$\{i\}$*. Stats.Bytes Received*, respectively). Although the second way to obtain end-to-end user throughput is more accurate than the first, it has one drawback since it requires that the user-devices are always active with the transmitting and receiving data requests.

B. Perception

The perception phase translates the sensed measurements from each node v_i into system variables (i.e. key performance indicators (KPIs)) that identify the network state.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

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. By means of real-time network monitoring of the aforementioned metrics, the perception function detects 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. These indicators include:

1) Channel utilization: (in %) at receiver node v_j placed at location l_j is calculated at radio interface d within the time interval τ based on the Clear Channel Assessment statistics of channel busy time in milliseconds as

$$u_d(l_j) = \frac{CBtime(t+\tau) - CBtime(t)}{\tau} \times 100. \quad (6)$$

2) Activity factor: (in %) at receiver node v_j placed at location l_j is calculated based on Clear Channel Assessment statistics of the channel transmit time (*CHTXtime*$(\cdot)$) and channel receive time (*CHRXtime*$(\cdot)$). Each of the above mentioned Clear Channel Assessment statistics parameters are vendor implementation specific - however, they are calculated based on different Clear Channel Assessment and PHY states indicators BUSY, IDLE, TX, RX [15]. Accordingly, the activity factor is given by

$$\alpha_d^{(l_j)} = \left[\frac{CHRXtime(t+\tau) + CHTXtime(t+\tau)}{\tau} - \frac{CHRXtime(t) + CHTXtime(t)}{\tau} \right] \times 100, \quad (7)$$

where *CHRXtime*$(\cdot)$ and *CHTXtime*$(\cdot)$, respectively, denote the total time in milliseconds that the radio has spent on receiving data and the total time in milliseconds it has spent on transmitting data. The values are obtained during the PHY receive and PHY transmit procedures. *CHRXtime*$(\cdot)$ is calculated as summation of the periods between PHY-RXSTART indication and PHY-RXEND indication, while *CHTXtime*$(\cdot)$ is calculated as a summation of the periods between PHY-TXSTART indication and PHY-TXEND indication [15].

3) Re-transmission rate per user device: (in %) is calculated based on user-level statistics data as

$$\Delta_{retr,k} = \frac{N_{retr,k}(t+\tau) - N_{retr,k}(t)}{N_{pack,k}(t+\tau) N_{pack,k}(t)} \times 100, \quad (8)$$

where $N_{retr,k}(t)$ is the total number of retransmissions for the kth user device at time instant t (e.g. vendor-specific implementation *InternetGatewayDevice. LANDevice.*$\{i\}$. *WLANConfiguration.*$\{i\}$. *AssociatedDevice.*$\{i\}$. *X_BL_TxRetries*) and $N_{pack,k}(t)$ is the total number of packets transmitted out of the interface for the kth user device at time instant t given by *InternetGatewayDevice. LANDevice.*$\{i\}$. *WLANConfiguration.*$\{i\}$. *AssociatedDevice.*$\{i\}$. *Stats. PacketsSent* [17].

4) Error rate per user device: (in %) is calculated as

$$\Delta_{err,k} = \frac{N_{err,k}(t+\tau) - N_{err,k}(t)}{N_{pack,k}(t+\tau) - N_{pack,k}(t)} \times 100, \quad (9)$$

where $N_{err,k}(t)$ is the total number of inbound failed packets for the kth user device at time instant t (e.g. vendor-specific implementation *InternetGatewayDevice. LANDevice.*$\{i\}$. *WLANConfiguration.*$\{i\}$. *AssociatedDevice.*$\{i\}$. *Stats. X_BL_TxFailed*).

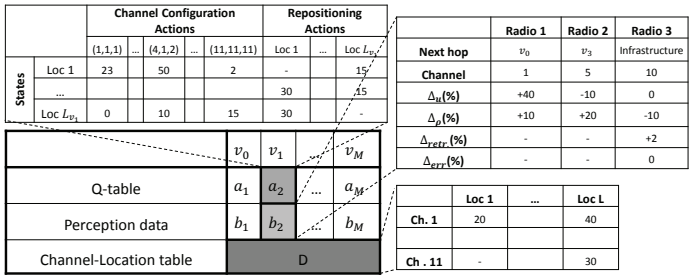


Fig. 3: Knowledge Base Design.

Remark 1 (Perception). The re-transmissions and error rate per user device give an insight to the severity of interference level. High level of interference consequently results in higher error and re-transmissions rates for the users impacted by the interference. The channel utilization metric gives an insight to the contention level, since the activity factor provides information on how much the radio traffic load contributes to that contention level. The value of the ratio of the activity factor to the channel utilization of the channel assigned to the radio d is used as a perception control variable along with the channel utilization value to trigger channel optimization. When this ratio has a very low value and the channel utilization value is higher than a certain threshold, the perception stage will detect the current network state as sub-optimal, resulting in an evaluation of the current state in the network. In order to avoid false alarms, the perception stage is responsible to correct values of the activity factor for radios that have connections among themselves.

For an example, assume that the link e_{ij} is formed of two nodes v_i and v_j on radio $d \in D_i$, where node v_i is a parent node. If the parent node has other connected devices on the same radio d as node v_j, then a high activity factor of the radio d of the parent node may contribute to a high channel utilization of the radio $d^{'}(=d) \in D_j$ at node v_j. In case that the activity factor of radio $d^{'} \in D_j$ has a very low value, but its channel utilization value is very high, it will consequently trigger the channel optimization. We note here that the parent node mostly contributes to v_j's channel utilization. This is a false alarm and it is necessary to modify the activity factor of the radio $d^{'} \in D_j$ at node v_j to the activity factor of radio $d \in D_i$ of its parent node.

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): Beside channels optimization, G-RL agent aims to place each node v_i at an optimal location. Thus, the state

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

$s \in S$ 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): G-RL agent takes two types of actions: channel configuration $A^{(c)}$ and EXT re-positioning $A^{(l)}$ with action set $A = A^{(c)} \cup A^{(l)}$. Since each node $\{v_i | \forall i \in [0, M]\}$ is equipped with $card(D_i)$ radios, we define channel configuration actions for that node as the set of all possible combinations of the radios as $a_c = [h_{d_1}, h_{d_2}, h_{d_{card(D_i)}}]$, where $|A^{(c)}| = N^{|D_i|}$ and $D_i = \{d_1, d_2, ..., d_{card(D_i)}\}$. On the other hand, each re-positioning action $a_l \in A^{(l)}$ changes the location of node v_i and results in a state transition. Finally, for each node v_i we define the combined action set $a = \{a_c, a_l\}$.

Reward (R): Instantaneous reward at time instant t in the state s for a selected action a at node v_i is calculated by Eq. (5) as

$$r_t(s, a, v_i) = \sum_{k=M+1}^{M+U} R_k. \tag{10}$$

We define the reward at the network level because applying an action a at node v_i impacts performance of whole network. In temporal difference Q-learning, the cumulative reward $Q_t(s, a, v_i)$ is calculated using the previous Q-value and the instantaneous reward as [21]

$$\begin{cases} Q_t(s, a, v_i) := Q_t(s, a, v_i) + \eta\Delta \\ \Delta = r_t(s, a, v_i) + \gamma \max\limits_{\forall a \in A} Q_{t+1}(s', a', v_i) - Q_t(s, a, v_i) \end{cases} \tag{11}$$

where $Q_t(s, a)$ is the cumulative reward at state s when action a is applied at time t. Parameters η and γ, respectively, are the learning factor and discount rate with values between 0 and 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^*. During the early stages of learning, when the KB is empty, the G-RL agent has to explore in order to discover the unknown environment. Subsequently, the KB is populated and the agent can retrieve and start exploiting the gained experience to pick an action that has the highest reward. 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 [21].

D. Knowledge Base

The agent applies the following four stages on the KB: 1) Retrieve the most relevant case, in the KB, to the currently sensed information; 2) Reuse the retrieved case or relative experience to solve the sensed problem; 3) Revise the KB by updating the actions or fitness values of the stored cases; 4) Retain the new experience (e.g. new case) in the KB to be used in the future. The proposed framework in Fig. 2 implements these stages as described in Section V. The knowledge base stores the three types of tables for each node v_i, $i \in [0, M]$ as shown in Fig. 3.

Perception table: stores all the information related to the connectivity in the network and the parameters calculated in the perception phase. 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 changes in utilization and activity factor denoted by $u_d^{(l_j)}$ and $\alpha_d^{(l_j)}$, respectively. With regard to connected users, the changes in retransmissions and error rates, denoted by $\Delta_{retr.}$ and $\Delta_{err.}$, respectively, are also stored.

Q-table: this table saves the Q-values for each possible action a in state s calculated by Eq. 11 [18].

Channel-Location table: the channel utilization of all available channels N and at all candidate locations L is kept at each time slot. Entries are set only for channels that were sensed at a certain location. Otherwise, the entries remain empty.

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

V. Guided RL Agent Design

The G-RL agent unifies both learning and autonomous decision making. G-RL agent after applying an action at certain node v_i, checks the nodes that are connected to that node (radio-table), applies the corresponding actions for those nodes if it is necessary to keep connectivity (this propagates through whole wireless system). After configuration of whole wireless system is done, the agent waits for environment feedback obtained by sensing and perception blocks. On basis on that feedback, the agent calculates Q-value for the action it took at node v_i and updates Q-values in Fig. 3 for all actions that this action of node v_i invoked. The main stages of the proposed method, i.e. ICALO, are summarized in **Algorithm 1**.

Remark 2 (Exploration/exploitation balance). The agent goal is to balance between exploring the environment to gain more information, and exploiting the knowledge base 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 G-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 [6]. In essence, the agent explores the environment when 1) the observed change in the reward values is insignificant, or 2) the learning cost is low due to the absence of user traffic. On the contrary, exploitation is applied when 1) large (positive) variations in the reward values are detected, or 2) interference or contention problems are perceived by Sensing function as indicated in Fig. 2. During both stages, the agent is aware of the following domain knowledge:

Spectral Correlation: Overlapping channels in a wireless system[2] 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.

[2]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).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

Algorithm 1: Guided VDBE – Softmax Q-Learning

Input : Knowledge Base (Q-table, Perception data and Channel-Location data);
Output : Action a^*;

1 **Define:** Max. channel utilization: u_{thr}, max. re-transmission rate: $\Delta_{retr._{thr}}$, max. error rate: $\Delta_{err._{thr}}$, min. signal level: $RSSI'$, and target Q-value: q';
2 **for** $v_i \in V$ **do**
3 /* Policy 1: select type of action */
4 **if** $RSSI \leq RSSI'$ *OR* $max(Q) < q'$ **then**
5 $a^* = OptimizeLocation$;
6 **end**
7 /* Policy 2: zero-cost guided exploration */
8 **while** $U == 0$ **do**
9 $c_{old} = c_{inf}$;
10 **if** $min(Q) == 0$ **then**
11 Calculate $\beta_a = \sum_{i=1}^{I} \sqrt{\sum_{d=1}^{D} (c_{a,d} - c_{i,d})^2}$;
12 $a^* = argmax\{\beta_a\}\ \forall a$;
13 **end**
14 /* All actions are visited before */
15 **else**
16 a^* = Random uniform selection ;
17 **end**
18 Apply a^* to c_{inf}; Sensing and Perception;
19 Update Q-table and Channel-Location Table;
20 Switch back $c_{inf} = c_{old}$;
21 **end**
22 /* Policy 3: modified soft-max */
23 **if** $u_d > u_{thr} \& \frac{\alpha_d^{(l_j)}}{u_d} \ll 1$ *OR* $\Delta_{retr.,r} > \Delta_{retr._{thr}}$ *OR* $\Delta_{err.,r} > \Delta_{err._{thr}}$ **then**
24 **end**
25 Calculate $\varepsilon(s)$ using Eq. 21
26 **if** $Uniform(0,1) \leq \varepsilon(s)$ **then**
27 $\rho(s,a) = min\{\rho_o(s,a), \rho_u(s,a)\}$;
28 $\rho_{max} = max(\rho(s,a))$;
29 $\rho_{min} = 0.9 \times \rho_{max}$;
30 Calculate Euclidean distance for each action a versus current action iff $\rho(s,a) > \rho_{min}$ and multiply it with channel diversity factor of action a, i.e. calculate factor κ_a ;
31 $a^* = argmax\{\kappa\}$;
32 **end**
33 **else**
34 $Q_{max} = max(Q(s,a))$;
35 $Q_{min} = 0.85 * Q_{max}$;
36 Calculate Euclidean distance for each action a versus current action iff $Q(s,a) > Q_{min}$ and multiply it with channel diversity factor of action a, i.e. calculate factor κ_a ;
37 $a^* = argmax\{\kappa\}$;
38 **end**
39 /* Policy 4: Decision Making - Control Stage */
40 **if** $Q(s,a^*) = 0$ *OR* $Q(s,a^*)! = 0 \& Q(s,a^*) > 1.15 * q_{curr}$ **then**
41 apply action a^*
42 **end**
43 **else**
44 keep current configuration $a^* = NULL$
45 **end**
46 Update cumulative reward Q-value using Eq. 11;
47 **end**
48 return a^*

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

The algorithm is detailed as follows.

A. Selecting the Type of Action

The first decision performed by the agent is to leverage the perception stage to decide either to explore or exploit. Using the perception data, the agent monitors the system performance by checking the changes in contention, interference and coverage levels at current extender location. In particular, the $RSSI$ value on 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 (Lines 4-6). In the case of poor coverage, channel exploration at such a location is unnecessary and thus a re-positioning action must be selected. For instance, at poor coverage locations the throughput is already low and channel optimization cannot improve the throughput – time wasting to learn at those locations. 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 in Fig. 3 with the corresponding channel utilization of the last channel configuration.

In case of 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 (Lines 4-6).

B. Zero-Cost Knowledge-Driven Exploration

The second policy performs greedy exploration, yet with zero learning cost, since it is followed in one of the cases:

1) No users (i.e. $U = 0$ in Line 8 of **Algorithm 1**) are actively associated or
2) Users are connected but not requesting traffic (Line 8).

In particular, the agent will pick a channel configuration action, compute its reward value and store the cumulative reward in the Q-table (Lines 10-12) to maximize the gained knowledge. As such, for every possible action that is not applied before (i.e. with zero reward value in Q-table), the total Euclidean distance, to all previously visited actions, is calculated by β_a as a sum of Euclidean distances between action a and the all previously applied actions given by upper limit of I stored in the Q-table as indicated in Line 11 of **Algorithm 1**. Here, $c_{a,d}$ is the channel configuration of radio d when applying action a. The optimal action, from the exploration perspective, is the one with maximum total Euclidean distance.

In particular, the agent checks for the number of connected users and their total amount of traffic to explore the set of actions (Lines 8-21). If no traffic is requested from the users,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

the agent decides to explore the environment by switching the radio to a channel that was not explored before (i.e. with no entry in the Q-table) or has low likelihood value calculated by (11). 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 (Lines 14-16). 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 KB, and the channel configuration is switched back to the former value (Line 20).

C. Exploiting Spectral Correlation

There are two basic methods for balancing exploration and exploitation: ϵ-greedy and Softmax, each with its own drawbacks [21]. Having said that, we adopted a strategy for policy proposed in [24], named as Value-Difference Based Exploration combined with Softmax action selection (VDBE – Softmax). The reasons are twofold: VDBE-Softmax selects the exploration actions only in situations when the knowledge about the environment is uncertain, indicated by fluctuations in Q-values during learning, and the second reason that the policy can outperform ϵ-greedy, Softmax and VDBE policies in combination with on-policy and off-policy learning algorithms such as Q-learning.

In the case of perceiving interference or contention problems (i.e. at least one condition is satisfied in Line 23), the third policy is triggered (Lines 24-38). The policy is jointly defined based on knowledge management (with KB), VDBE–Softmax [24] and spectral correlation. We mention here that by sensing RL agent has some knowledge about its environment, which is stored in *Channel-Location* table and Radio-table of each node $v_i \in V, i \in [0, M]$. By utilizing this knowledge, RL agent modifies action probabilities, so the proposed action selection policy is given as

$$\pi(s) = \begin{cases} \rho(s,a) & \xi < \varepsilon(s) \\ arg \max\limits_{\forall a \in A(s)} Q(s,a) & otherwise, \end{cases} \tag{12}$$

where $\rho(s, a)$ and ξ, respectively, denote the G-RL action selection policy described in Section V-C1 and a uniform random number over the interval [0, 1]. The $\varepsilon(s)$ 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) G-RL 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

$$\rho(s_t = s, a_t = a, v_i) = \min\left[\rho_o(s,a), \rho_u(s,a)\right] \tag{13}$$

by using the BSmax probability $\rho_o(s, a) = Pr\{a_t = a|s_t = s\}$ and the environment probability $\rho_u(s, a, v_i)$ that 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 by a normalized exponential function (i.e. Boltzmann distribution) as follows:

$$\rho_o(s,a) = \frac{e^{\frac{Q(s,a)}{Temp}}}{\sum_{b \in A(s)} e^{\frac{Q(s,b)}{Temp}}}, \tag{14}$$

where T is a positive parameter called temperature starting with a large value and decreases with time. High temperatures cause all actions to be nearly equiprobable (more exploration), whereas low temperatures cause greedy action selections (more exploitation), while $\rho_u(s, a, v_i)$ is determined as follows

$$\rho_u(s,a,v_i) = \frac{\Sigma(s,a,v_i)}{\Theta(s,a,v_i) + \Xi(s,a,v_i) + \Delta(s,a,v_i)}. \tag{15}$$

In the above expression, $\Sigma(s, a, v_i)$ denotes the impact of channel diversity given as

$$\Sigma(s,a,v_i) = 1 + \sum_{d \in D_i} \sum_{d' \in D_i, d \neq d'} |h_d - h_{d'}|, \tag{16}$$

where h_d denotes the selected channel for the given link $e_{i,j}$ within the action set a_c. This sensor gives a higher weight to actions with operating channels having larger separation, and vice versa. This means that the action with the same channel tuned on all radio interfaces has the lowest probability of self-interference due to multiple radio interfaces per device. $\Theta(s, a, v_i)$ denotes an observed channel utilization on the node v_i for all radios given as

$$\Theta(s,a,v_i) = \sum_{d \in D_i} u_{h_d}^{l_i}. \tag{17}$$

$\Xi(s, a, v_i)$ 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, as

$$\Xi(s,a,v_i) = \sum_{j, e_{ij} \in E} |u_{h_d}^{l_i} - u_{h_d}^{l_j}| \times 100. \tag{18}$$

Finally, $\Delta(s, a, v_i)$ denotes the impact of interference and contention from overlapping channels given as

$$\Delta(s,a,v_i) = \frac{\sum_{d \in D_i} \sum_{h \in N, |h - h_d| \leqslant 5, h \neq h_d} (5 - |h - h_d|) u_h^{(l_i)}}{50}, \tag{19}$$

where h denotes a channel within the set of total number of available channels defined by N. By experimental studies the developed heuristics is adopted in Eq. (18) and Eq. (19). Eq. (19) denotes the contention impact for 2.4 GHz band. Thus, the constant value of 5 in the numerator of Eq. (19) is a consequence of the fact that there are only 3 non-overlapping channels in 2.4 GHz band. The second step is finding the maximal probability $\rho_{max} = \max(\rho)$ and on basis of it calculating the minimal allowed probability as (assuming 10% error tolerance to avoid oscillations of action selection).

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 $\kappa_{a'}$ for each a' as

$$\kappa_{a'} = \left(1 + \sum_{d \in D_i} \sum_{d' \in D_i, d \neq d'} |h_{d,new} - h_{d',new}|\right)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

$$\times \sqrt{\sum_{d \in D_i} (h_{d,current} - h_{d,new})^2}, \qquad (20)$$

where $h_{d,new}$ denotes the new channel index which will be assigned to radio d if action $a^{'}$ is applied, while $h_{d,current}$ denotes the current applied channel index at radio d. The first factor in the brackets denotes the channel diversity of the channel configuration defined by channel action $a^{'}$, since the second denotes Euclidean distance of the channel configuration defined by channel action $a^{'}$ from the current applied channel configuration 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 due to overlapping properties in wireless spectrum. Thus, the G-RL agent gives a higher probability to actions that have higher Euclidean distance from the currently applied action. In the last step, the G-RL agent picks the action a^* that has highest κ value.

2) Exploration Probability $\varepsilon(s)$: The state-dependent exploration probability $\varepsilon(s)$ is calculated using the difference in Boltzmann distribution between the last two cumulative rewards:

$$\begin{cases} \varepsilon_{t+1}(s) = \psi(s) f(s,a,\sigma) + [1 - \psi(s)]\varepsilon_t(s), \\ f(s,a,\sigma) = \frac{1 - e^{\frac{-|\eta \Delta(s,a)|}{\sigma}}}{1 + e^{\frac{-|\eta \Delta(s,a)|}{\sigma}}} \end{cases} \qquad (21)$$

where σ and $\psi \in [0,1]$, respectively, denote a positive constant called inverse sensitivity and the influence of the selected action on the state-dependent exploration probability. 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 a way that low values cause full exploration at small value changes while high values of σ cause a high level of exploration only at large value changes.

D. Decision Making - Control of Convergence

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 (Lines 40-45). In the case that 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 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 the network oscillating between the same states.

Remark 3 (Convergence Analysis). Q-learning has been well studied in the literature and, under mild assumptions, has been proven to converge to the desired optimal solution [22]. The original Q-learning algorithm uses a stochastic iterative update to determine the optimal Q-values [21]. The update-rule for Q-learning is given by Eq. (11). We consider the entire wireless network as a single RL-agent, while the reward is calculated on the network level. Due to multiple extenders in the neighboring environment an action is applied by G-RL agent taking into account the entire network state. In that way, the correlation among multiple extenders is taken into consideration. Since there are a finite set of actions (available channels and limited number of locations) and a finite set of states (we are assuming that number of locations is limited), G-RL agent converges to optimal (steady-state) point as illustrated in the following section. However, we note here that the optimality point may change with the dynamics of the environment. In dynamic environment, general RL-agent suffers from oscillations and it fails to converge [21]. In order to provide the convergence of G-RL agent, we exploited both the spatial and spectral correlations, while further oscillations are reduced by the decision control policy (i.e. Lines 40-42 in **Algorithm 1**). For two-hops network in dynamic environment, we proved.

VI. Performance Evaluation

In this section, we first describe the simulation results with network simulator ns-3 and then, the experimental results are presented.

A. Network Simulator ns-3

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. 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. SNR based ideal rate adaptation is used and the MAC protocol is IEEE 802.11.

In each 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: $\varepsilon_{EXT}(0) = 1$, $\varepsilon_{mAP}(0) = 1$, $Temp = 50$, $\sigma = 100$, $\psi_{EXT} = \frac{1}{121}$, $\psi_{mAP} = \frac{1}{11}$, $\eta = 0.7$, $\gamma = 0$, $\tau = 2$, $u_{thr} = 60(\%)$, $RSSI' = -60$ dBm, $\Delta_{err.} = 0.005\%$, $\Delta_{retr.} = 50\%$. The network area considered is size of 20×10 meters. All the nodes of our network (mAP, EXT and client devices) are placed within this area. For the purpose of this simulation, we consider an apartment consisting of 8 rooms as given in Fig. 4(a). Additionally, APs belonging to neighboring external networks may be placed outside of this network area. In all tests, every node (internal or external) was placed in an enclosing area of 30 m × 20 m. Note that all interfering external APs transmit at a rate of 5 Mbps to an associated node placed outside the apartment.

We divide the testing process into three phases to highlight different aspects of our approach: 1) Speed of convergence to near-optimal throughput; 2) Comparison of steady-state throughput to other channel-assignment schemes; 3) Resilience to dynamic network conditions.

1) Speed of convergence to near-optimal throughput: ICALO takes time in trying out different channel assignments and locations before arriving at a final state (steady-state). Therefore, it is important to understand how the network

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

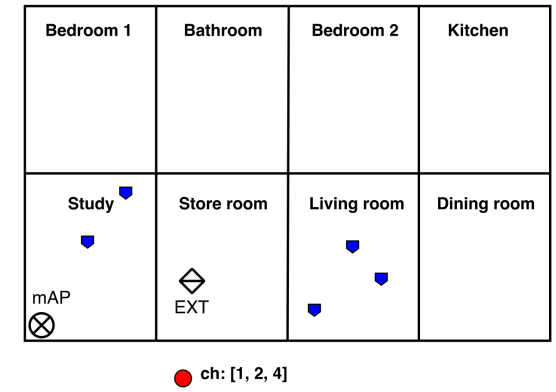
throughput will be affected during this period. In this experiment, we try to demonstrate this behavior of ICALO and more importantly, its speed of convergence to the steady-state throughput. Short time periods for channel switching and repositioning are not considered in the analysis of results as they make no effect on the convergence behavior (other than act as small delays).

We consider a family of five living in the apartment, three in the living room and two in the study. This is visualized in Fig. 4(a). The mAP and EXT are initially placed as indicated in Fig. 4(a). Then, we introduce a single external (non-managed) node to act as external interference to our network. The fronthaul and backhaul channel of the EXT is initially set to 2 and 6, respectively. The external AP channel is set to 2. The variation of per-user throughput versus time when running ICALO in this scenario is given in Fig. 4(b). Here, we see that the network reaches near-optimal throughput at its steady-state in around 26 seconds, the convergence time of ICALO (indicated by the dashed line). Note that this is just the initial convergence time; as ICALO learns, the convergence time will drop (see later in Fig. 7(b)). From the more pronounced peaks and valleys before the dashed line in Fig. 4(b), we can get an idea of how many changes in the channel configuration of the EXT occurred before the steady-state (for this particular arrangement of nodes, there was no EXT re-positioning suggested by ICALO). The actual number of channel changes to reach the steady-state was 11.

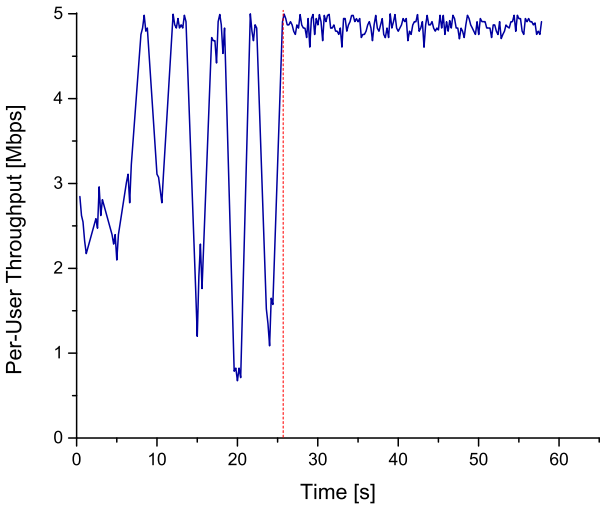
To get a more general idea on the convergence times and the number of configuration changes (location changes plus channel changes) to reach steady-state, we conduct 50 tests with the same configuration as in Fig. 4(a), except that the EXT is placed at a random location within the apartment in each test. The fronthaul and backhaul channels of the EXT is set to 3 and 7, respectively, while the external AP is in channel 3. We observed that for 90% of tests, the convergence time is less than 36 s – all tests converged within 43 seconds. The mean convergence time was 23.6 s with a standard deviation of 7.9 s. Analysis of configuration changes until steady-state revealed that the mean number of configuration changes is 9.4 and that the standard deviation is 2.5. These numbers further validate what we observed in Fig. 4(b).

2) Comparison of steady-state achievable throughput: We compare the steady-state throughput of ICALO with that of three other channel assignment approaches – namely single channel assignment, Common Channel Assignment (CCA) [13] and Connected Low Interference Channel Assignment (CLICA) [29] for two different scenarios. In each scenario, we place the client devices in a constant arrangement of locations, and randomly change the initial position of the EXT 50 times and measure the steady-state per-user throughput of ICALO along with that of the other approaches. Hence, each experiment consists of 200 tests – 50 for each approach. Note that within a given experiment, the initial location of the EXT, the channels of the interfering external APs, and the positions of the client devices are kept constant, so as to facilitate a fair comparison.

In each individual test, we assign all possible channels to the EXT fronthaul and backhaul, and consider the throughput



(a) Scenario



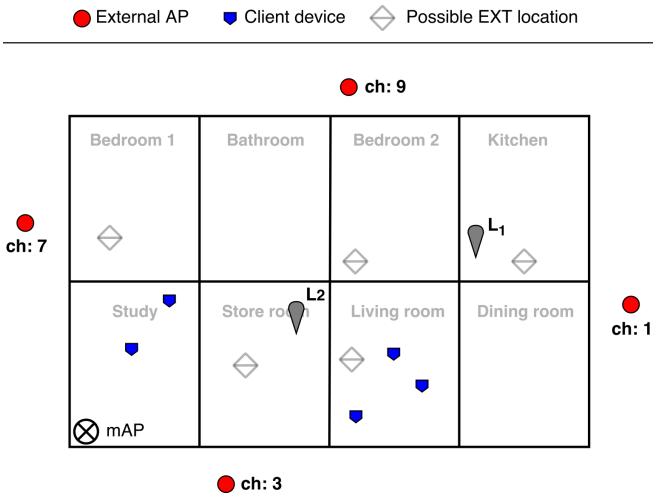
(b) Per-user achievable throughput

Fig. 4: Convergence study in dynamic network conditions with contention.

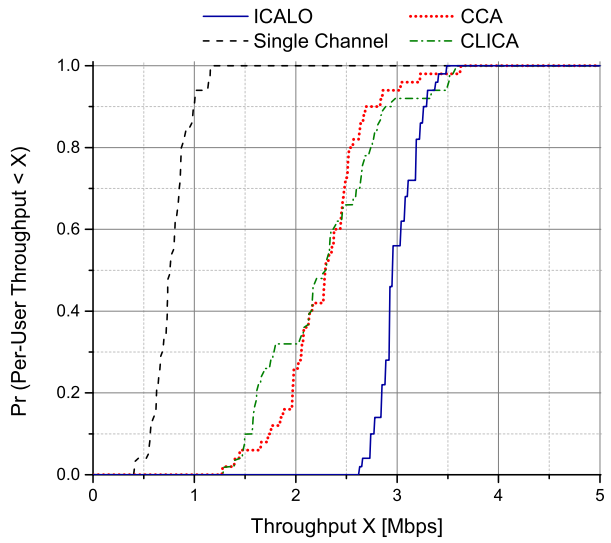
of the channel that produced the highest throughput as the throughput for that test. This is to get the best possible throughput for each test under the constraint of using a single channel (in single channel assignment, it is to be assumed that there is a single available channel). The essence of CCA is to assign the same set of channels for each radio of every node in a WMN, to have the maximum possible level of inter-node connectivity while having channel variation to reduce interference. To get a high throughput under this premise while maintaining fairness, we assign a random couple of orthogonal channels to the EXT fronthaul and backhaul in each test. To construct the conflict graph in CLICA, we consider the physical model, which assigns edge weights based on the value of certain network physical parameters as presented in [29] and originally proposed in [30] (the alternative protocol model does not capture interference due to overlapping channels).

In both scenarios (Experiment 1 in Fig. 5(a) and Experiment 2 in Fig. 6(a)), we consider congested environment with four external APs. The respective channels of the external APs in Experiment 1 are 1, 3, 7, 9, while in the Experiment 2 the channels of external APs are 1, 2, 4 and 8 (the operating channel of a external AP is given next to that AP in the figures). In each of the experiments, the initial position of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

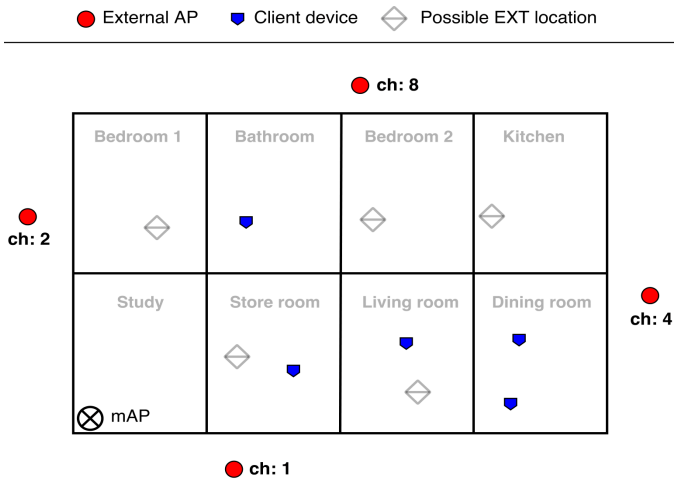


(a) Scenario

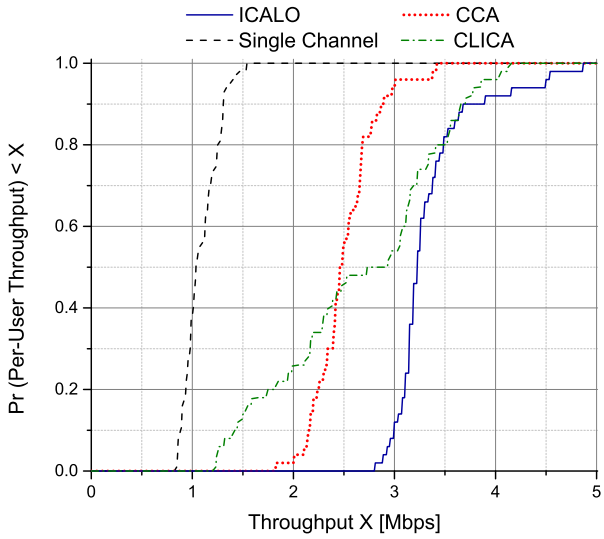


(b) CDF

Fig. 5: Comparison of ICALO with state-of-the-art methods and pre-defined backhaul interference.



(a) Scenario



(b) CDF

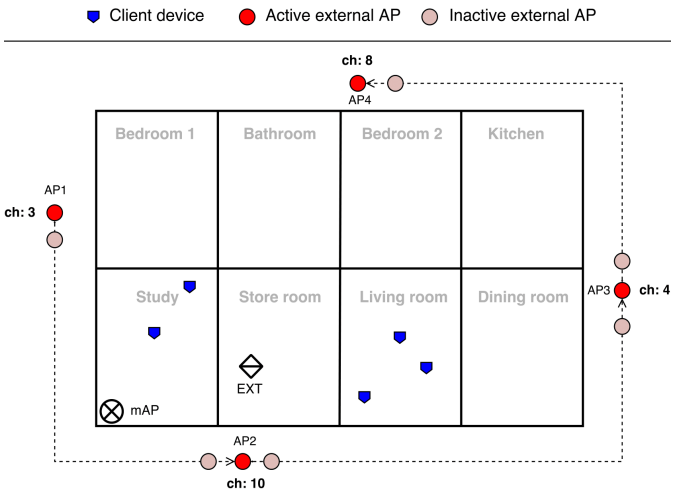
Fig. 6: Comparing ICALO performance with state-of-the-art and dynamic backhaul interference.

EXT is randomly set within the confines of the apartment. We place the client devices and the external AP in the positions as indicated in the figures. The initial channel at the fronthaul and backhaul of EXT is set to 3 and 7, respectively.

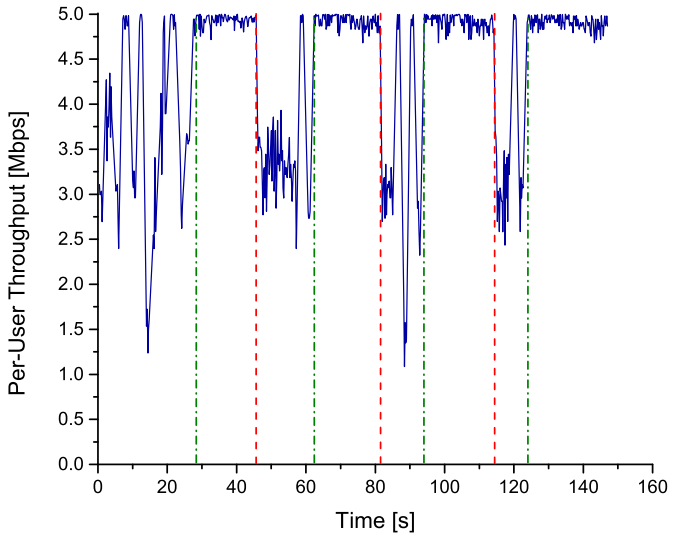
The results of Experiment 1 and Experiment 2 are shown in Fig. 5(b) and Fig. 6(b), respectively. For each experiment, the cumulative distribution function (CDF) of the steady-state throughput of each algorithm is plotted. With the increased and more complex external interference, the average steady-state throughput is much less than that of the case of a single interfering channel (Fig. 4(b)). Observing the results in Fig. 5(b) and Fig. 6(b), as expected the single channel assignment has the lowest achievable throughput. CCA performs much better as it eliminates (in our tests) inter-channel interference by choosing orthogonal channels. However, even in this case, it has no information about 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.

Low throughput results of CCA and CLICA in Fig. 5(b) and Fig. 6(b) were caused in scenarios where the EXT was placed in locations too far away or too near to the mAP. In such situations, no channel assignment can recover the degradation of throughput caused due to the poor location of the EXT. On the other hand, ICALO was able to alleviate this by re-positioning the EXT. For example, the lowest throughput for CCA and CLICA in Experiment 1 is 1.3 Mbps, where the EXT was initially positioned in location L_1 (see Fig. 5) – clearly a bad location for it considering the locations of client devices. By initially re-positioning the EXT to location L_2, ICALO was able to eventually obtain about $2\times$ improvement of steady-state throughput of 2.7 Mbps. This portrays the tight coupling between channel assignment and location of EXTs in the goal for throughput enhancement.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology



(a) Scenario



(b) Results of experiment

Fig. 7: Recovery of per-user throughput in dynamic network conditions by ICALO.

3) Resilience of ICALO to dynamic network conditions: With the use of wireless enabled devices continuing to grow at an astronomical rate, any deployment of a new wireless network should expect interference from neighboring external networks. Such scenarios are extremely dynamic – new devices will get added and existing devices will leave unpredictably. As such, modern networks should be resilient in the face of these effects and be able to recover from them quickly to reach peak performance. ICALO has an advantage in this respect as it keeps getting better as time passes, and is able to make smarter, faster decisions based on its ever-growing knowledge base.

To verify this claim, we simulate the scenario illustrated in Fig. 7(a): starting with AP_1 activated, remove and activate each of the APs, $\{AP_1, AP_2, AP_3, AP_4\}$ 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 AP_4 is left activated. $AP_1, AP_2, AP_3 and AP_4$ transmit at channels 3, 10, 4 and 8, respectively. The per-user achievable throughput variation for this scenario is shown in Fig. 7(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 from the figure that the network reaches near-optimal throughput at each stage after successfully recovering from the decline in throughput due to a sudden change in the external interference conditions. Note also that successive convergence time in seconds decreases as 28, 16, 12.5 and 10. 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 a little lag.

B. Testbed Evaluation

We practically evaluate the feasibility of the proposed framework by developing a full prototype. We consider an experiment with COTS devices running Linux Embedded Development Environment (LEDE) [25]. For the mAP, we use a TP-Link Archer 1750AC router with one 2.4 GHz radio configured to operate in two modes—ad-hoc (connecting EXT) and infrastructure (connecting user devices). The experiment includes 30 COTS wireless nodes with additional 15 wireless nodes that are not managed and serve as random interference and contention generators. We design EXT with two 2.4 GHz radios by combining two APs (TP-Link Archer 1750AC) in such a way that LAN interface of one is connected to the WAN interface of other. One AP operates in ad-hoc mode, while the other operates in infrastructure mode. Both the mAP and EXT are having a channel width of 20 MHz. The wireless repeating mode of Wireless Distribution System (WDS) is used to connect mAP and EXT. To test in a more challenging environment, we selected the 2.4 GHz band due to a larger number of neighbors that are not available on the 5 GHz band. We equip the EXT with a USB-to-audio adapter and speaker in order to enable cyber-user interface. By this interface, ICALO notifies an end-user when to re-position the EXT. At both the mAP and EXT, we host a part of the sensing logic which periodically reports network parameters, that is done by combination of Linux Shell and Python programming. The logic of the other blocks of ICALO is hosted on a MATLAB server that uses secure shell (SSH) to push new configurations to the wireless system.

1) Self-optimization scenario: A validation of ICALO is done in the non-managed environment with a layout shown in Fig. 1, where mAP is placed at location l_0. The initial location of EXT is not pre-defined (ICALO will suggest one). We consider the worst-case scenario with always active users in single and multi-user scenarios. In case of a single-user scenario, the user is located at l_2 with 2K video demand. The RSSI from mAP at location l_2 is below -75 dBm and to serve this user, an extender is needed. In single-user scenario, ICALO firstly optimizes the location of the EXT and then searches for an optimal channel assignment. With regards to channel assignment, we compare ICALO (with the proposed G-RL agent) with an unguided RL (UG-RL) agent. In both cases, when ICALO or UG-RL agent decides on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2926409, IEEE Transactions on Vehicular Technology

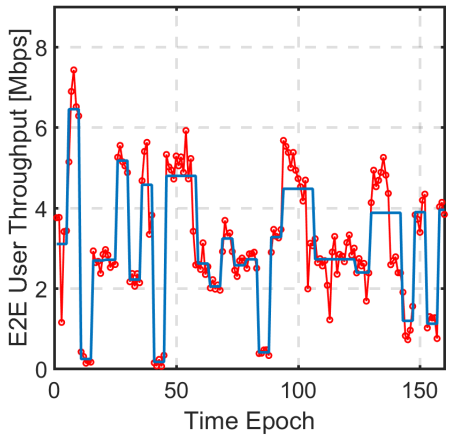
Loc/Ch	1	2	3	4	5	6	7	8	9	10	11
l_{mAP}	-	1e3	38	39	38	50	-	40	74	-	1e3
l_{EXT}	62	1e3	37	37	61	84	71	35	74	44	1e3

TABLE I: Channel - Location table

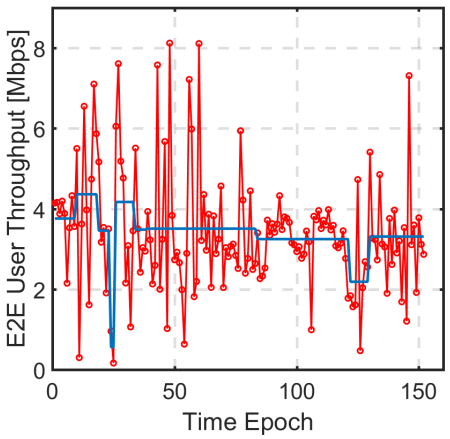
optimal channel combination, a hidden node is introduced at the backhaul link to test their responsiveness. In the multi-user scenario, the mAP and EXT each have two connected and active user devices. Parameters related to ICALO and the system are listed as: $\varepsilon_{EXT}(0) = 1$, $\varepsilon_{mAP}(0) = 1$, $Temp = 50$, $\sigma = 100$, $\psi_{EXT} = \frac{1}{121}$, $\psi_{mAP} = \frac{1}{11}$, $\eta = 0.7$, $\gamma = 0$, $\tau = 4$, $u_{thr} = 60(\%)$, $RSSI' = -65$ dBm, $\Delta_{err.} = 0.005\%$, $\Delta_{retr.} = 50\%$.

Location Optimization: The initial recommendation of the EXT's placement is mid-way between the locations of user device at l_2 and mAP at l_0 in Fig. 1, somewhere close to location l_4. After the initial placement of the EXT, by means of sensing and perception, ICALO validates the average RSSI level of the EXT received at the mAP's location (-70 dBm) and the RSSI level of the user device at the EXT's location l_4 (-44 dBm). Since the RSSI level of the EXT received at mAP is below $RSSI' = -65$ dBm, ICALO sends a voice notification to the user to reposition the EXT to a new location l_1, mid-way between the EXT's current location, l_4, and the mAP's location, l_0. After re-positioning the EXT to l_1, ICALO again validates the average RSSI level of the EXT received at the mAP's location l_0 (-56 dBm) and RSSI level of the user device at the EXT's current location l_1 (-58 dBm). Since the RSSI levels satisfy the RSSI constraint, ICALO can start searching for an optimal channel combination for backhaul and fronthaul links.

Unguided Channel Optimization: Unguided channel optimization relies only on BSmax probabilities without domain knowledge. That is, when $\xi < \varepsilon(s)$, only ρ_o is considered when selecting the next action (this is the classic Softmax exploration). As a consequence, the UG-RL agent requires a longer searching time to find an optimal configuration as illustrated in Fig. 8(a), with a high likelihood to apply channel combinations with poor performance. Thus, the wireless system experienced poor performance for a longer time in comparison with ICALO. Also, to find optimal channel combinations, UG-RL agent applies far more actions (higher learning cost) than ICALO, leading to the degradation of user experience due to many re-connections and delays for re-association of both EXT's and user devices. We note here that the channel combinations with poor performance due to high level of contention, and/or large errors caused by hidden nodes, require more time to establish connection between mAP and EXT, and also between user devices and mAP/EXT. This time (in range of several to tens of minutes) is referred to as a dead time in the wireless system, and it increases with higher channel utilization and/or interference. To reduce the dead time, a distributed logic at both the EXT and mAP is added (EXT is not visible to G-RL agent in the cloud) to reset the system configuration to so far best-known settings. As such, ICALO has a much smaller probability to visit actions with poor performance compared with a UG-RL agent.



(a) Instant Reward (UG-RL agent)



(b) Instant Reward (G-RL agent)

Fig. 8: Single-user Scenario

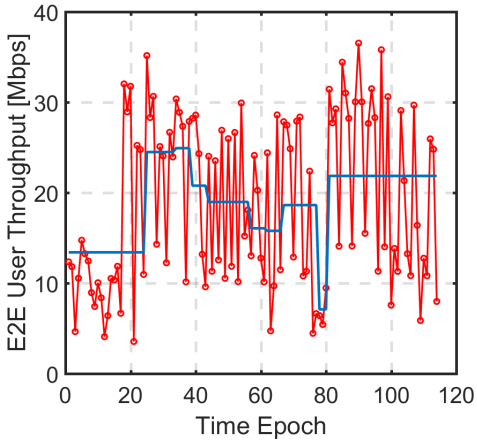


Fig. 9: Multi-user Scenario (G-RL agent)

2) Guided Channel Optimization for Single-user Scenario: Guided by the domain knowledge, the G-RL agent used in ICALO significantly decreases the search time for the optimal channel configuration as shown in Fig. 8 (b). To maximize the initial learning space of ICALO, the agent starts with non-overlapping channels (e.g. channel 3 for backhaul and channel 8 for fronthaul). After a new configuration is pushed, ICALO collects a number of sensing samples (4 in our case) with period $T = 4$ s before reasoning about applying a new action. Also, to avoid situations where a collection of a certain number of sensing samples lasts long, ICALO specifies the maximal time it will wait for collection as 120 s. Subsequently, with each new channel action, ICALO sends to the node the best-known channel action so far. This is necessary to avoid the channels which don't allow re-establishment of all links of a certain node in 30 s. For those channels, ICALO sets the channel utilization to 1000, to stress poor performance at those channels. The tested environment includes 61 and 72 non-managed neighbors sensed at mAP's location and EXT's location, respectively. The level of contention is very high for each channel and most of the channels are highly utilized (see Table I, which is updated by the perception block with each new sensing sample). By applying a new action (channel configuration), ICALO acquires knowledge about the utilization of the current and adjacent channels, and calculates hidden node and contention node impacts. In that way, ICALO keeps the average throughput in the wireless system approximately constant (4 Mbps) and only needs a very short period ($T = 40$ s) to learn the neighborhood. From $T = 40$ s, G-RL agent chooses between two channel combinations (4,10) and (3,7). After the G-RL agents stabilizes the wireless system, we add a hidden node at the backhaul at time instant 120 s, to test ICALO's responsiveness to a dynamic environment. An additional AP is placed at location l_3 which operates at channel 3 and has an associated active user device at saturated traffic load. Consequently, the utilization of channel 3 is increased from 38 to 73. Here, ICALO detected a very high level of error rate at the mAP, and takes only two iterations to avoid the hidden node problem as illustrated in Fig. 8(b).

3) Multi-users scenario: In this scenario, we consider only ICALO (G-RL). There are 2 user devices connected to the mAP with RSSI levels of -35 dBm and -56 dBm and 2 user devices connected to EXT with RSSI levels -48 dBm and -58 dBm. All the devices stream a 2K video. As shown in Fig. 9, ICALO performs very well in case with multiple users and avoids the channels with poor performance. It is worth noting that the tested environment is very dynamic, and during testing we observed that ICALO very quickly adapts to changes in the neighborhood.

VII. Conclusion

This paper presented ICALO, a self-optimization scheme for wireless extenders in a WMN which adopts an AI-driven learning framework. ICALO optimizes the operating channels and locations of extenders by striking a balanced trade-off between their backhaul and fronthaul performance, considering the impact of uncoordinated neighboring networks, learning cost and network dynamics. 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 a UG-RL. 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 NP-hard problem of joint channel assignment and location optimization of WMNs by guaranteeing low-cost learning and achieving near-optimal network configurations. The basic idea of the proposed ICALO framework is validated for a 2-hop network with both single and multiple stations by considering only the 2.4 GHz band. In most modern households, a single extender coupled with an mAP will cover the performance needs of an overwhelming majority of users. As such, the demonstrated performance of ICALO will serve as an important marker of the Quality of Service that can immediately be offered to these users through home wireless networks. 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).

References

[1] H. Gacanin and A. Ligata, "Wi-Fi self-organizing networks: Challenges and use cases," IEEE Communication Magazine, Network & Service Management Series, Vol. 55, Issue 7, pp. 158–164, July 2017.

[2] W. Peng, W. Gao, and J. Liu, "A Novel Perspective on Multiple Access in 5G Network: Framework and Solutions," IEEE Wireless Communications, open access, DOI: 10.1109/MWC.2019.1800315, 2019.

[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] H. Gacanin and M. Wagner, "Artificial Intelligence Paradigm for Customer Experience Management in Next-Generation Networks: Challenges and Perspectives," IEEE Network Magazine, Vol. 33, Issue 2, March/April 2019.

[5] 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.

[6] H. Gacanin, "Autonomous Wireless Systems with Artificial Intelligence," IEEE Vehicular Technology Magazine, Special issue on 6G: What is Next?, May 2019.

[7] 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.

[8] 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.

[9] 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. 206–218, Feb. 2013.

[10] A. Raniwala and T.-C. Chiueh, *Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network*, IEEE Conference on Computer Communications (INFOCOM), pp. 2223–2234, 2005.

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

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

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

[14] R. Atawia and H. Gacanin, "Self-deployment of future indoor wireless networks: An artificial intelligence approach," in Proc. IEEE GLOBECOM 2017, 2017, pp. 1–6, Singapore.

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

[16] Broadband Forum (DSL Forum TR-069), "CPE WAN Management Protocol", www.broadband-forum.org, May 2004.

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

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

[19] G. Gui, H. Huang, Y. Song and H. Sari, "Deep learning for an effective non-orthogonal multiple access scheme, IEEE Transactions on Vehicular Technology, vol. 67, no. 9, pp. 8440-8450, Sept. 2018.

[20] H. Zhu, Y. Cao, W. Wang, T. Jiang, S. Jin, "Deep Reinforcement Learning for Mobile Edge Caching: Review, New Features, and Open Issues," IEEE Network, Vol. 32, Issue 6, November/December 2018.

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

[22] S. P. Singh, T. Jaakkola, M. Littman, and C. Szepesvari, *Convergence results for single-step on-policy reinforcement-learning algorithms.* Machine Learning, vol. 38, no. 3, pp. 287–310, 2000

[23] R. Akl and A. Arepally. *Dynamic Channel Assignment in IEEE 802.11 Networks*. Proc. IEEE International Conference on Portable Information Devices (PORTABLE'07), 2007.

[24] 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).

[25] Linux Embedded Development Environment (LEDE) *https://lede-project.org/.*

[26] M. Alicherry, R. Bhatia and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," ACM MobiCom 2005, Aug. 28 – Sept. 2, 2005, Cologne, Germany.

[27] R. M. Karp. *Reducibility among combinatorial problems.* in Complexity of Computer Computations, R. E. Miller and J. W. Tathcher, Eds, New York: Plenum Press, pp. 85-104, (1972).

[28] W. K. Hale, *Frequency Assignment: Theory and Applications* Pin Proceedings of the IEEE, Vol. 68, No. 12. (1980)

[29] A. Adya, P. Bahl, J. Padhye, A. Wolman, Lidong Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," Proceedings of the IEEE International Conference on Broadband Networks (BroadNets), 2004.

[30] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu, "Impact of interference on multi-hop wireless network performance," ACM MobiCom, 2003.

[31] S. Karunaratne, R. Atawia, E. Perenda and H. Gacanin, "Joint Channel and Location Optimization of Wireless Networks with Artificial Intelligence," 2018 IEEE Global Communication Conference (IEEE Globecom 2018), December 2018, Abu Dhabi, UAE.