UDC 654.165
doi:10.15217/issn1684-8853.2015.3.77

AN EFFICIENT CROSS-LAYER AWARE MAPPING OF VOIP CALLS IN WIRELESS OFDMA SYSTEMS Part II. Mapping Algorithms and their Performance

Y. Ben-Shimol[a], *PhD, Electrical Engineering, Professor, benshimo@bgu.ac.il*
I. Kitroser[a], *PhD, Electrical Engineering, kitroser@bgu.ac.il*
[a]*Ben-Gurion University of the Negev, POB 653, 1, Ben Gurion St., Beer Sheva, 74105, Israel*

Purpose: *This work address and solve the problem of efficient broadcasting of resource allocations descriptors for VoIP traffic in mobile OFDMA-based wireless systems.* ***Methods:*** *We show that the mapping overhead is substantially reduced by using semi-persistent allocations and that by taking advantage of the periodicity of VoIP frames, generated by a multi-phase vocoder. To handle the impact of mobility on the characteristics of the wireless channel we utilize a cross-layer decision approach to track the channel conditions and predict the expected mobile user behavior such that the system may under-react to channel changes in some cases.* ***Results:*** *The development efforts in this paper were directed towards developing efficient heuristic solutions since the problem is NP hard. Two heuristic algorithms with low computational complexity of* $O(n^2)$ *are presented and their performance gains are compared against a simple persistent allocation approach. Extensive simulations show that these mapping algorithms reduce the allocation overhead substantially in OFDMA-based systems such as WiMAX and LTE. The main advantage of the proposed algorithms is their ability to support multiple codecs, or a single codec with different operational modes, that result in different packet sizes and different periods. In addition, our proposed algorithms cluster the assignments together hence enabling the BS to allocate unused resources to other traffic types.* ***Practical relevance:*** *The proposed algorithmic solutions are simple enough to be implemented in practical OFDMA systems such as WiMAX and LTE to allow an efficient use of bandwidth for high quality VoIP sessions in of mobile users.*

Keywords *— Cross-Layer Design and Optimization, Resource Allocation and Interference Management, Mobile Multimedia Technology, 3.5G and 4G Technologies.*

Introduction

An important application in future IP based wireless communication networks is voice over IP (VoIP), which allows real-time voice calls between mobile stations (MSs). Most VoIP applications generate traffic of fixed size packets at a constant rate. Therefore, the traffic of a single VoIP session is classified as constant bit rate (CBR) traffic type and requires a fixed number of uplink (UL) resources on a periodic basis. This suggests that with a naive resource allocation scheme the up-link map (UL-MAP) includes allocation descriptors (or information elements — IEs) for VoIP streams, thus generating overheads at a constant rate. A typical voice call lasts several minutes on the average, while its packet period is usually up to several tens of milliseconds. This suggests that the total amount of overhead for each VoIP call is usually very large.

In part 1 of this paper [1] we underlined the necessary steps that are required for the implementation of a full cross-layer solution for the mapping overhead problem in the presence of mobility. Here we address the problem of the decision algorithm and present several new algorithms and an extension to the minimum overhead algorithm (MOA) [2] to efficiently support VoIP traffic under dynamic channel conditions. The algorithms use an efficient way to handle mobile channels and variations of the modulation and coding schemes (MCS) by using the finite state Markov chain (FSMC) model that was presented in [1], for approximating and tracking the channel states.

We present a number of algorithmic solutions for persistent allocation, which fully complies with the messaging mechanism presented in [3, 4] (see [1] for a detailed discussion). Our proposed algorithms solve the problem of how to map slots to MSs such that the overall overhead is decreased. In this work we generalize the model given in [2] to support dynamic channels where the channel may change rapidly (i.e., mobile users in urban environment). This generalization enables us to model both codecs supporting silent suppression or adaptive rate and/or changes in the state of the mobile channel (and hence, supporting also variable MCS). Our proposed model enables us to uniformly represent an adaptive multi-rate (AMR) codec [5] with multiple states or multiple coders with different VoIP traffic parameters where none of the reviewed solution handles the multiple vocoder case. We show that our proposed algorithms reduce the mapping overhead for the dynamic cases as mentioned above, while keeping a simple and manageable resource allocation mechanism. In addition, the suggested algorithms cluster all VoIP assignments together, hence enabling easier resource management for other traffic types.

The rest of the paper is formed as follows: section “Problem definition” presents the problem and notations for the allocation problem; section

"Mapping schemes" presents the mapping algorithms for the mobile channels scenario along with the numerical results for the performance evaluation of the given algorithms. Summary and conclusions are given at the end of the paper.

Problem Definition

The definition of the efficient VoIP mapping problem is:

Definition. Given a sequence of allocation tables $A[k]$, $k = 0, 1, ..., K$, and a set of requests $R = \{r_1, r_2, ..., r_q\}$, find a set of non-overlapped allocations $S = \{S_1, S_2, ..., S_q\}$ (i.e., $S_i^k \cap S_j^k = \varnothing$, $\forall i \neq j$), such that the mapping overhead is minimized.

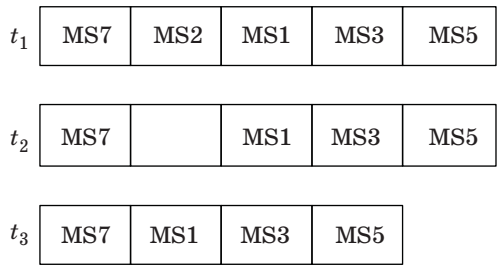
For the problem presented above we assume the following:

1) the solution needs to use the mechanism of persistence allocation. That is, the understanding that a user uses the last description for the allocated resources until (if at all), there is a change made by the base station (BS), and this change is presented as an information element (IE);

2) variations in the physical channel of each user are tracked continuously by a FSMC model. It is assumed that approximations for the states probabilities, transitions probabilities are known;

3) the metric for efficiency is the number of IEs used by the system.

Assumption 2 above implies that in cases of changes in channel state that need to be followed by reallocation for a user there is unallocated space (name a hole) that needs to be used in order to increase the efficiency of using the system resources. We provide a family of solutions and an explicit modification to the MOA such that the dynamic nature of the channel is taken into consideration. The main idea is to track the channel behavior of the MS and provide a predictor of the channel state changes probabilities and rates, which then can be used by the mapping algorithm. Such measures can indicate the stability of the channel, which can be used as an allocation grouping criteria to reduce the number of holes in the allocation table. See Fig. 1 [3] for illustration of hole-filling operation, and later see the entropy mapper that is presented in Algorithms 2 and 3. In addition, we use the predictor of channel behavior, since rate assignments are later related to channel states. This also has implication on how to quantify the cost of state change in term of system resources and hence decides whether it is cost effective to perform an MCS state transition or just ignore it (see the Extended MOA (EMOA) mapper that is presented later).



■ ***Fig. 1.*** Illustration of resource hole and resource shifting for five resources. At t_1 there are 5 sub-bursts. At $t_2 > t_1$ MS2 is deleted, thus generating a hole. At $t_3 > t_2$ MS1, MS3 and MS5 are shifted to the right to fill the hole

Mapping Schemes with Dynamic Channels

In the static case presented in [2] the baseline mapping algorithm treated the VoIP data as regular data without using a persistent approach. Hence, the basic mapper's overhead was directly related to the system load. In this work, we extend the MOA semi-fixed allocation mapper (SFAM) algorithm presented in [2] to efficiently support a dynamic channel scenario. We use the channel profile (as recommend in [6]) of a mixture of static, pedestrian and mobile users. In addition, to align with current IEEE 802.16 and 3GPP LTE traffic models, we use AMR as a primary voice codec with two working profiles. In order to evaluate our algorithms, we provide a simple persistent mapping algorithm, which complies with the persistent allocation mechanisms in IEEE 802.16-2009 [3] and 802.16m-2011 [4].

Simple persistent mapper

The simple persistent mapping (SPM) algorithm is a straightforward implementation of persistent allocation mechanism supported by [3, 4]. As stated in the problem definition, one of our goals is also to reduce the effective area of the overall VoIP allocations, meaning that the allocation area needs to be condensed, rather than letting it to spread over the entire resource domain. A possible motivation for this is to support other types of traffic, which may require consecutive resources for allocations. Essentially, this means that we would like to perform persistent allocation while eliminating resource "holes" and for this purpose we will use the resource shifting mechanism as supported by [3, 4].

The inputs to the SPM algorithm are lists of new and active requests and a current frame number k. We denote a request r with period p_r in *NewReguests* and *ActiveReguests* by a and n, respectively. The list is sorted according to the requests allocation slot index. The output of the simple persistent algorithm is the UL-MAP that is represented by a list of information elements (IEL). We use the function *Allocate*(r, s_i, k, IEL) (depicted in Algorithm 1) for allocating request r at slot s_i in frame k. The func-

tion allocates the required slots in line 2 and updates the index of next available slot for allocation in line 7. In lines 4–5, the function allocates a new UL-MAP allocation element m_r^k only if there was a change in the number of calculated for the last allocation for this request.

The SPM algorithm is depicted in Algorithm 2. The SPM algorithm starts by scanning the *ActiveRequests* for changes in request size due to an MCS change or due to collision with other active requests in this frame. Each such request is moved to the *NewRequests* list. Next, we search for holes between the active request allocations. Each such hole is eliminated either by filling a new request or by using the resource shifting procedure, which shifts all subsequent active requests. The computational complexity of SPM follows from the number of required operations to map an instance of requests. For any on-line mapping of a given set of requests, at least one operation is required for mapping each request. Therefore, the complexity of any mapper is at least $O(n)$ and this holds for the standard mapper. The complexity of the proposed mapping algorithms can be separated into operations that take place before and during the mapping process. For the pre-mapping operations, assuming that the mappers' input is not ordered with relation to the periodicity of the requests, $O(n \cdot \log n)$ operations are required to sort the requests according to their periodicity. The worst case scenario for SPM of mapping one request may require considering all other input requests, therefore mapping all requests requires $O(n^2)$ operations.

Algorithm 1. Allocate
Input:{r, k; s_i, IEL}

1. $m_r^k = \emptyset$
2. $S_r^k = \left\{ s_i, s_{i+1}, \ldots, s_{\bar{N}_r - 1} \right\}$
3. remove r from *NewRequests* or *ActiveRequests*
4. **if** $S_r^k \neq S_r^{k-p_r}$ **then**
5. $m_r^k \leftarrow \left\{ s_i, \bar{N}_r \right\}$
6. **end if**
7. $s_i \leftarrow s_i + \bar{N}_r$
8. $\text{IEL} = \text{IEL} \cup \left\{ m_r^k \right\}$

The Entropy Mapper

We note that in the case of a dynamic environment, MSs may change their MCS profiles, hence creating holes. In such cases, the simple persistent algorithm will try to fill the holes with a new request or perform resource shifting. Resource shifting invokes an additional system cost since it requires an additional MAP IE. As stated earlier in the section discussing general channels, we are interested in channel dynamics and managing a FSMC to track channel conditions by estimating each state's steady state probability π_k [given by equation (12) in [1]]. As can be seen from Fig. 2, *a*–*d*, the state probabilities change and become more diverse as the channel conditions degrade. This means that for mobile users, the state transition probabilities increase, therefore the number of the required slots for allocation varies more frequently.

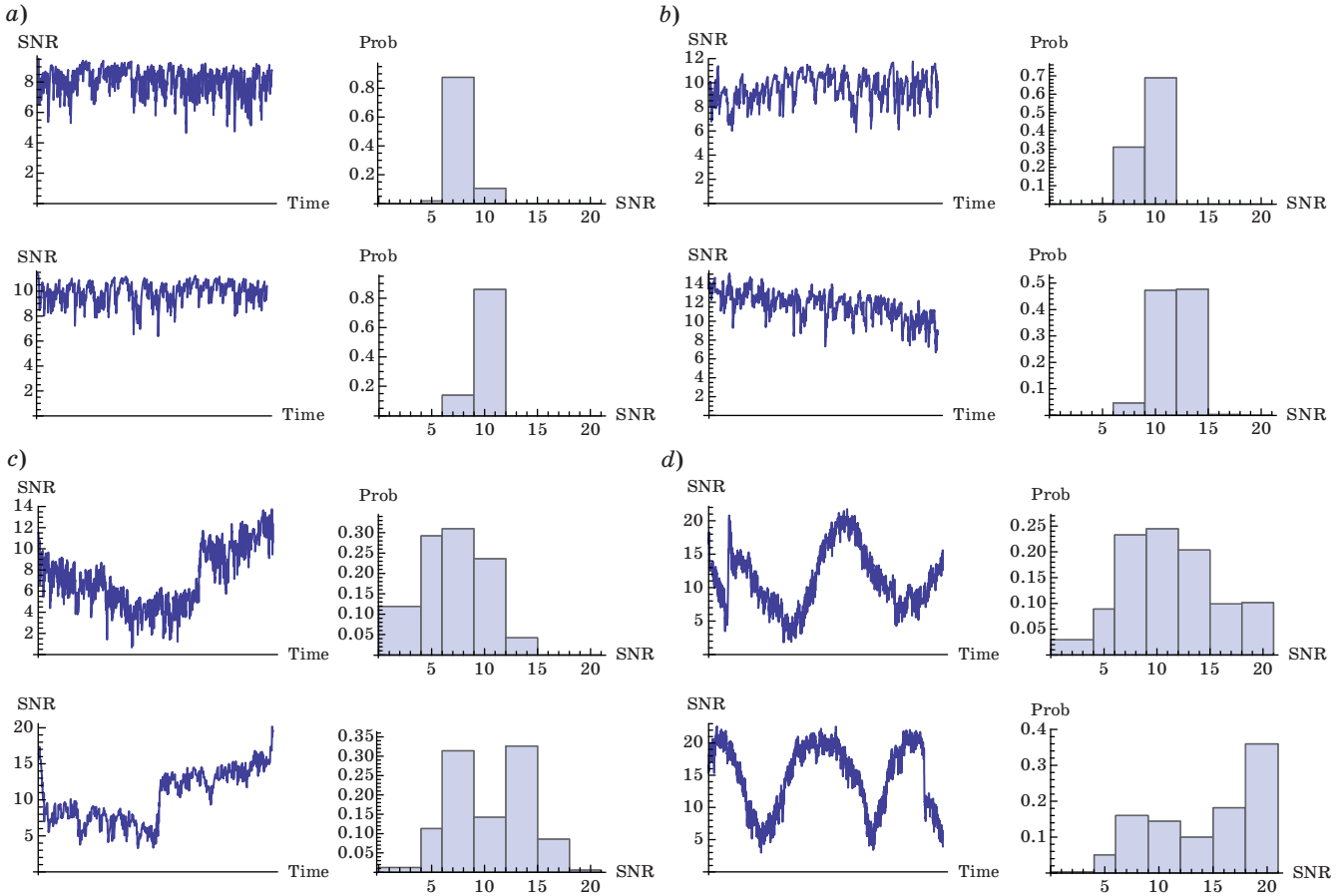
Our objective is to identify such MSs and reduce the number of operation required to condense the allocation region due to frequent MCS changes. By grouping such MSs at the end of the allocation region we decrease the number of holes that are created due to MCS changes, or at least decrease the number of shift operations. We use an entropy measure to

Algorithm 2. SPM
Input:{*NewRequests*, *ActiveRequests*, k; IEL}

1. $slot = 0$; $\text{IEL} = \emptyset$; $r = \emptyset$;
2. **for** $a_i \in ActiveRequests$ **do**
3. **if** $\bar{N}_{a_i}^k \neq \bar{N}_{a_i}^{k-p_{a_i}}$ OR $\exists a_j \neq a_i, \left\{ S_{a_i}^k \cap S_{a_j}^k \neq \emptyset \mid a_j \in ActiveRequests \right\}$ **then**
4. remove a_i from *ActiveRequests* and prepared a_i to *NewRequests*;
5. **end if**
6. **end for**
7. **while** $s_{a_1} = slot$ **do**
8. *Allocate* (a_1, k, *slot*, IEL)
9. **end while**
10. **if** $ActiveRequests \neq \emptyset$ **then**
11. $hole = s_{a_1} - slot$
12. **else**
13. **goto** Step 22
14. **end if**
15. $r \leftarrow \max \left\{ n_i \in NewRequests \mid \bar{N}_{n_i} \leq hole \right\}$
16. **if** $r \neq \emptyset$ **then**
17. *Allocate* (r, k, *slot*, IEL);
18. **else**
19. *ShiftResources* {*slot*, *hole*, IEL};
20. **end if**
21. **goto** Step 7;
22. **while** $NewRequests \neq \emptyset$ **do**
23. *Allocate* (n_1, k, *slot*, IEL)
24. **end while**
25. **return** IEL

identify the dynamics of each MCS. Let $\pi_0(t)$, ..., $\pi_K(t)$ denote the state probabilities of a FSMC at time instance t, where each state corresponds to an MCS level. The entropy $H_i(t)$ of MS i at time t is defined as

$$H_i(t) = -\sum_{k=0}^{K} \pi_k(t) \log \pi_k(t). \qquad (1)$$



■ ***Fig. 2.*** Two examples of SNR behavior in time and the FSMC histogram for: *a* — static MSs; *b* — Pedestrian MSs (3 km/h); *c* — Vehicular MSs (30 km/h); *d* — Vehicular MSs (120 km/h)

We note that given two MSs i, j with $H_i(t) \geq H_j(t)$, the channel of MS i is "more dynamic" than that of MS j. The entropy measure provides a simple yet effective improvement to the simple persistent allocation simply by sorting the new requests in increasing order of their entropy measure. Hence, we can define the Entropy Mapper algorithm as depicted by Algorithm 3.

Algorithm 3. Entropy
Input:{*NewRequests*, **H**, *ActiveRequests*, *k*; IEL}
1. Sort *NewRequests* according to **H**;
2. IEL ← *SPM* {*NewRequests*, *ActiveRequests*, *k*};

The argument **H** is a vector of all corresponding $H_i(k)$ values for a current time frame k. It is assumed that each MS calculates this value and that there is a feedback mechanism to report this value to the BS. The complexity of the Entropy Mapper algorithm is same as for SPM algorithm, that is, $O(n^2)$.

Extended MOA

The next algorithm we present here is an extension to the MOA described in [2]. The EMOA modifies the MOA in order to handle dynamic channels. The main idea is to track the channel using the FSMC and maintain both the state probabilities $\pi_0(t)$, ..., $\pi_K(t)$ and the transition probabilities [according to (12) and (13), respectively, given in [1]]. It is assumed that the FSMC states are sorted in a non-decreasing order according to the supported rates r_i, $i = 0, ..., K$, hence $r_0 \leq r_1 \leq ... \leq r_K$. If there is no MCS change, that is, no state transition in the corresponding FSMC, then the MS's request is treated regularly by MOA. But if there is an MCS change, for example from i to j, then the algorithm calculates a cost function to decide whether to comply with the channel conditions and change the MCS or not. The cost function compares the loss due to the decision to remain in state i compared to the overhead required to change to MCS j. The main motivation here is that if the current change is merely a jitter, that is, the channel will return to state i fast enough, then the loss due to staying in the same MCS is tolerable comparing to the overhead induced by the MOA to re-map the request.

Let $COST_{IE}$ denote the cost of a MAP-IE for indicating an allocation. Also, let $E_{i,k}$ denote the average number of steps to first enter state k when starting

from state i. $E_{i,k}$ is calculated by solving the expression for the hitting time [see [1] for expression (15)]. We can see that if $|i - k| \le 1$ and there is a bound of maximum allowed $L \le K$ steps, we need to solve $\min\{\lfloor L/2 \rfloor, K - k\}$ equations for the case of $i = k + 1$ and $\min\{\lfloor K/2 \rfloor, k\}$ equations for $i = k - 1$. Finally, let $\overline{COST}_{IE}$ denote the average cost in terms of MAP-IEs added in the MOA due to a hole created by changing an active request to a new request due to an MCS change. In MOA, such a hole may be filled by using another active request, which must be re-allocated and may create an additional hole.

Equation (2) denotes a cost function that quantifies (in terms of slots) the relative cost for user m when the system moves from system state i to system state j, using the ratio between the predicted system loss if the state change is ignored (not changing the specific allocation of the user) and the system loss due to re-allocation of the user due to the state change:

$$Cost_{i,j}(m) = \frac{F(i,j) \cdot E_{j,i}}{Slots\left(2\left(COST_{IE} + \overline{COST}_{IE}\right)\right)}. \quad (2)$$

$F(i, j)$ is a function of the change direction:

$$F(i,j) = \begin{cases} Slots\left(PKT_{bits}(m)\right)_{SNR_i} - \\ -Slots\left(PKT_{bits}(m)\right)_{SNR_j}, & i < j \\ Slots\begin{pmatrix} \bar{N}\left(P_j\left(PKT_{bits}(m)\right)\right) - \\ -\bar{N}\left(P_i\left(PKT_{bits}(m)\right)\right) \end{pmatrix}, & i > j \end{cases}, \quad (3)$$

where $Slots(B)_{SNR}$ is the number of slots required to transmit B bits with a given SNR and $Slots(B)$ denote the corresponding number of slots using the SNR for the UL-MAP message. $\bar{N}\left(P_i\left(PKT_{bits}(m)\right)\right)$ defines the number of bits required on the average to transmit a voice packet of size $PKT_{bits}(m)$. Here, we assume that not changing the system state results with a higher bit error rate (BER). Hence, on the average, the system may suffer from packet loss and will be required to retransmit the voice packet, which will result in system's goodput loss. We calculate $\bar{N}(\cdot)$ as follows: Let p^i denote the system bit error probability in state i with $SNR = SNR_i$. Than,

$$P_i\left(PKT_{bits}\right) = 1 - \prod_{i=1}^{PKT_{bits}} \left(1 - p^i\right) \quad (4)$$

denotes the packet error probability of a packet with PKT_{bits} bits. Note that since the system state is not changed, $P_j(PKT_{bits})$ represents the packet error probability of a packet with PKT_{bits} where the system is in *state i* and with $SNR = SNR_i$.

The probability of m consecutive packet transmissions (due to errors) is obtained from a geometric probability thus, the average number of transmitted packets is be given by

$$N\left(P_i\left(PKT_{bits}\right)\right) = \sum_{m=1}^{\infty} m \cdot P_i^m\left(PKT_{bits}\right). \quad (5)$$

Taking a practical bound of no more than $T_{\max}$ retransmissions (thus maximal number of transmissions equals $T_{\max}$) we get:

$$\bar{N}\left(P_i\left(PKT_{bits}\right)\right) = \sum_{m=1}^{T_{\max}} m \cdot P_i^m\left(PKT_{bits}\right). \quad (6)$$

The modified MOA calculates $Cost_{i,j}(m)$ for each user m at a transition change $i \to j$ in order to decide whether to change the state or not. The decision basically compares the cost of changing the state or not. If $Cost_{i,j}(m) > 1$ than the user's MCS state is changed, otherwise the system remains at the current MCS level.

Algorithm 4. EMOA
Input:{$NewRequests_j$, $PendingRequests_j$, $ActiveRequests_j$, k; IEL}

1. **for** $a_i \in ActiveRequests$ **do**
2. **if** $State_{a_i}^{k} \ne State_{a_i}^{k-p_{a_i}}$ **then**
3. Calculate $Cost\left(State_{a_i}^{k-p_{a_i}}, State_{a_i}^{k}\right)(a_i)$ according to equation (2);
4. **if** $Cost\left(State_{a_i}^{k-p_{a_i}}, State_{a_i}^{k}\right)(a_i) > 1$ **then**
5. move request a_i from *ActiveRequests* to *NewRequests*;
6. **else**
7. move request a_i from *ActiveRequests* to *PendingRequests;*
8. set $TS_{a_i} = k$; $E_{a_i} = E\left(State_{a_i}^{k-p_{a_i}}, State_{a_i}^{k}\right)$;
9. **end if**
10. **end if**
11. **end for**
12. **for** $a_j \in PendingRequests$ **do**
13. **if** $State_{a_j}^{k} = State_{a_j}^{TS_{a_j} - p_{a_j}}$ **then**
14. move request a_i from *PendingRequests* to *ActiveRequests;*
15. **else if** $TS_{a_j} - k > E_{a_j}$ **then**
16. move request a_j from *PendingRequests* to *NewRequests*;
17. **end if**
18. **end for**
19. IEL $\leftarrow MOA\{NewRequests, \{ActiveRequests \cup PendingRequests\}, k\}$

We note that if we don't change the MCS level, we calculate the actual cost of this decision by finding the actual overhead that resulted as a function of time. If the actual overhead exceeds the cost

of changing the state, we change the current state. If, at any given time, the system returns to state i, the process is stopped.

The EMOA is depicted in Algorithm 4. We added two parameters for each request r: TS_r and E_r. TS_r denotes a time stamp to tag the frame in which the request was moved from the list of active requests (*ActiveReguests*) to the list of pending requests (*PendingReguests*). E_r denotes the estimated number of steps to return to the original state that was used in the cost function. In addition, we assume that $State_r^k$ denotes the FSMC's state of the MS with request r at frame k. If the state was changed and the cost function is less than one, the algorithm records TS_r and E_r as defined above and moves the request to *PendingReguests*. For each request in *PendingReguests*, the algorithm compares the current state with the state before the change. If the MS returns to its original state, the request is moved back to *ActiveReguests*. Otherwise, if the MS is not in its original state for more than the predicted number of steps, the request is moved to the new requests list *NewReguests*. We note that the operation defined above is done for all requests independent of their period; hence we ignore the period of the lists, which is later used by the MOA. The complexity of the EMOA Mapper algorithm remains the same as for the MOA and SPM algorithm, that is, $O(n^2)$.

Numerical Results — Dynamic Case

System model

In order to study the performance of the suggested algorithms, we used a system-level simulator, which complies with the baseline configuration and simulation assumptions in the IEEE 802.16m evaluation methodology [6]. We used a single cell environment with a cell radius of 1.5 km. The major parameters are defined in tab. 1.

The MSs are uniformly dropped within the cell and we use random interval mobility model with cell wrap-around. We define two normalized offered load models per MS: one with a medium load of $a = 0.5$ Erlang and another with a high load of $a = 0.9$ Erlang.

This load model defines the average percentage of registered MSs that are in active session at each frame. Each MS, when finishing its current session, will choose the start time of its next session based on an exponential distribution with mean of $\lambda = Na/T$, where N is the number of registered MSs and T is the average call duration. We use this load model rather than just increasing the number of registered MSs with always-on sessions, since we want to simulate the dynamic behavior of entering or leaving requests, rather then just requests that transit between active and idle states.

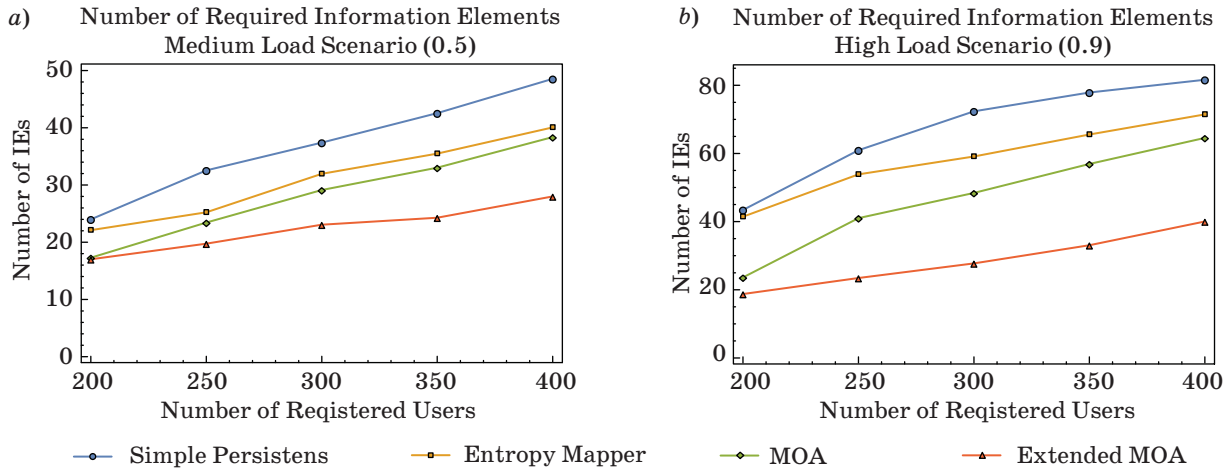
■ ***Table 1***. System parameters used for the evaluation of VoIP persistent allocation algorithms

Parameter	Value
FFT Size	1024
CP	1/8
Operating Bandwidth	10 MHz
Duplexing scheme	TDD
Subchannelization	PUSC
Coding	Convolutional Turbo Coding (CTC)
Cell Radius	1.5 km
BS Antenna Height	32 m
MS Antenna Height	1.5 m
Carrier Frequency	2.5 GHz
BS TX Power	42 dBm
Number of BSs	1
Number of MSs	25–400
Path Loss Model	PL(dB) = 130.18+37.6log(R_{Km})
Lognormal Shadowing Standard Deviation	8 dB
Mobility	0–120 km/h
Channel Mix (ITU)	Static — 12 % Pedestrian B (3 km/h) — 56 % Vehicular A-1 (30 km/h) — 16 % Vehicular A-2 (120 km/h) — 16 %

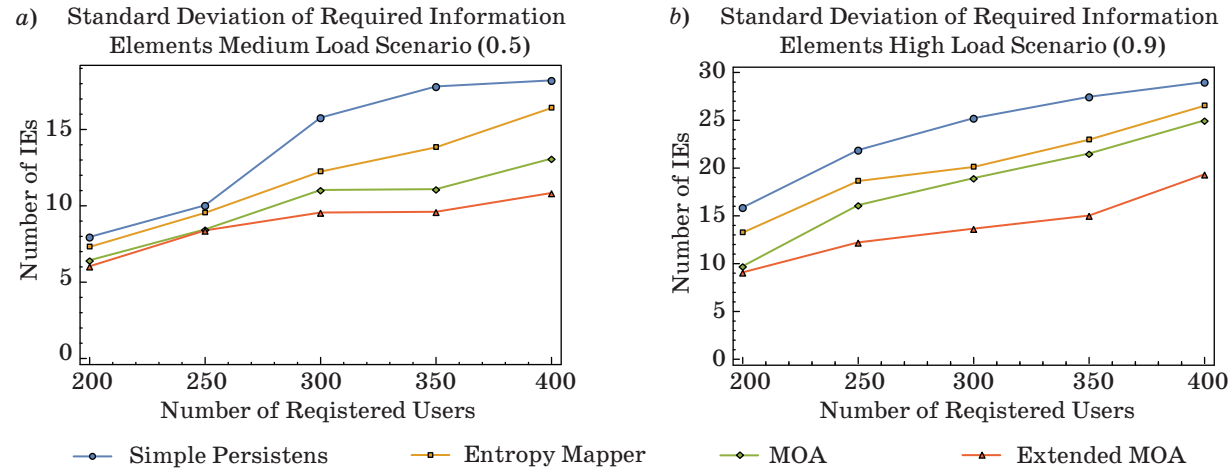
■ ***Table 2***. AMR 12.2 Codec parameters

Parameter	Value
AMR Codec Frame Length	20 ms
Voice Payload with RTP and 802.16 overhead	44 Bytes
Inactive Payload (with same overhead as voice)	18 Bytes
Average Call Time	~Exp(μ), 1/μ = 210 s
Talk Spurt length	~Exp(μ), 1/μ = 1026 ms
Silence length	~Exp(μ), 1/μ = 1171 ms

We consider a VoIP model, that contains periods of active talking spurts (ON periods) interleaved by silence periods (or OFF periods). Specifically, we consider the AMR codec [5], which is optimized for speech coding and was adopted as the standard speech codec by the 3GPP standardization group, widely used in GSM and is recommended for system evaluation of WiMAX systems [6]. Tab. 2 describes the parameters for the RTP AMR 12.2 codec with a voice activity factor of ~50 %.



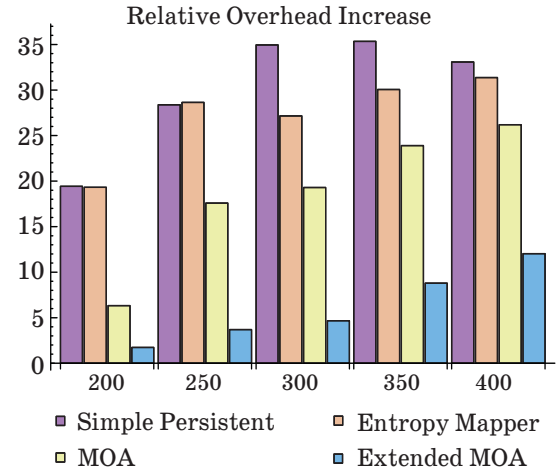
■ ***Fig. 3.*** Comparison of overhead between the VoIP mapping algorithms for a load of 0.5 (*a*) and 0.9 (*b*) Erlang



■ ***Fig. 4.*** Comparison of standard deviation between the VoIP mapping algorithms for a load of 0.5 (*a*) and 0.9 (*b*) Erlang

Discussion

Figures 3, *a* and *b* depict the average number of IEs included in the UL-MAP of the algorithms defined for the dynamic case. Compared to the static case, here we can see that there is more overhead, since the operational state of the MSs change more frequently. We can also see that the extended MOA provides considerable performance gain compared to the other algorithms in all load scenarios. In addition, the entropy mapper algorithm, which organizes the requests according to their channel dynamics and uses the simple persistent algorithm, provides a considerable overhead decrease compared to the simple persistent mapper. This is due to the fact that when all the dynamic MSs are grouped together, the probability of consecutive holes increases, and hence, these holes are easier to fill or require less resource shifting operations for removing them. In addition, we can observe that



■ ***Fig. 5.*** Relative overhead increase between medium and high loads for the proposed algorithms as function of system load

when the system dynamics increases (the high load scenario in Fig. 3, *b*), the performance differences between the algorithms increase. Figures 4, *a* and *b* depict the standard deviation of the UL-MAP IEs per scenario and algorithm. It can be seen that the variation coefficient for all the algorithms is between 30 to ~55 %. Also, the algorithms maintain the relative order of performance with the standard deviation of the overhead. Finally, Fig. 5, shows the relative overhead increase between the medium load and high load scenarios as a function of the number of registered MSs. It can be observed that the extended MOA has the lowest sensitivity factor to load increase, which enables it to maintain low load overhead compared to the other algorithms.

Conclusions

This paper discusses the mapping overhead problem in IEEE 802.16 OFDMA-based systems. This problem needs to be solved for practical systems such that the advantages of the OFDMA technology can be applied in future broadband wireless communication systems, both fixed and mobile. We are using the notion of semi-fixed allocations (which is similar to the persistent allocation) that enables the BS to discard IEs from the UL-MAP, thus reducing mapping overhead. Since the problem of reducing the mapping overhead is NP hard, as already was shown in [2], our effort was directed at developing efficient heuristic solutions. We extend the fixed model of [2] to support the dynamic case with a mixture of mobility models and VoIP codec with silent suppression. The dynamic case is mainly characterized by frequent MCS changes due to changes in channel states during the VoIP call. The efficiency of the algorithm may be enhanced when deciding not to change the allocation for a short change in channel conditions. In order to achieve this goal, we provided channel tracking and prediction model by employing the FSMC model and simple approximations of state probability and state transition probability. Two algorithms that extend the VoIP mapping for the dynamic case were presented and their performance gain compared to a simple persistent allocation approach was shown.

The main advantage of the proposed algorithms is their ability to support multiple codecs, or a single codec with different operational modes, that results in different packet sizes and different periods. In addition, our proposed algorithms cluster the assignments together hence enabling the BS to allocate unused resources to other traffic types. Our main contribution in this paper is to develop explicit mapping algorithms that efficiently reduce the mapping overhead of VoIP data usage. Our solutions present a cross-layer oriented approach, which track the channel state in order to predict the expected cost of channel change on the overall system cost. One of the main observations is that in some cases, the BS should under-react to changes in channel conditions since it is more rewarding from a system-wise perspective. The proposed algorithms conform to the mechanism of persistent allocation as defined in the IEEE 802.16 standard.

Our proposed algorithms were applied to UL-MAP only. Specific adaptation of the algorithms is required for adjusting them to support DL allocations. In addition, the proposed algorithms are not necessary limited to IEEE 802.16 based technologies, and future research is adopting them to LTE based systems.

References

1. **Ben-Shimol Y., and Kitroser I.** An Efficient Cross-Layer Aware Mapping of VoIP Calls in Wireless OFDMA Systems. Part I: Problem description and channel tracking. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2015, no. 2(75), pp. 90–97. doi:10.15217/issn1684-8853.2015.2.90
2. **Kitroser I., Chai E., and Ben-Shimol Y.** Efficient Mapping of Multiple VoIP Vocoders in WiMAX Systems. *Wireless Communications and Mobile Computing*, 2009, vol. 11, no. 6, pp. 667–678.
3. **802.16-2009** IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE 802.16-2009, May 2009.
4. **IEEE Standard** for Local and Metropolitan Area Networks. Part 16: Air Interface for Broadband Wireless Access Systems. Amendment 3: Advanced Air Interface. IEEE 802.16m-2011, May 2011.
5. **Mandatory** Speech Codec Speech Processing Functions; Adaptive Multi-Rate (AMR) Speech Codec; Transcoding Functions, 3GPP TS 26.090V10.0.0, March 2011.
6. **Draft** IEEE 802.16m Evaluation Methodology, IEEE 802.16m-07/037r2, December 2007.