A Distributed Dynamic Frequency Allocation Algorithm

Behtash Babadi and Vahid Tarokh

School of Engineering and Applied Sciences
Harvard University

Outline of Topics

1. Intoduction
2. The Algorithm
3. Main Results
4. Simulation Results
5. Time-varying Case
6. Conclusion

Outline

1. Intoduction
2. The Algorithm
3. Main Results
4. Simulation Results
5. Time-varying Case
6. Conclusion

Introduction

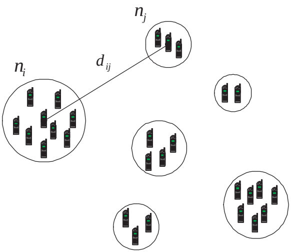
- In many emerging wireless networks no central frequency allocation authority is naturally available.
- Examples are
  - Ad hoc Networks
  - Cognitive Radios
- Optimal frequency allocation requires full knowledge of the spatial distribution profile of the network nodes.
- This makes distributed frequency allocation an important but mostly unchartered territory in wireless networking.
- Objective: Dynamic assignment of the frequency bands to the users in the network in order to minimize the interference.

System Model

- Various networks are naturally clustered (e.g., combat scenarios, WLAN Hotspots, WPAN.)

- In this light, we can partition the network elements into a union of clusters.

- We assume that the clusters are already formed in a specified manner.



- N clusters, c_i, $i = 1, \cdots, N$, where each cluster has a cluster head responsible for managing some of the network functions. d_{ij} denotes the distance between the cluster heads of c_i and c_j.

Interference Model

- We assume that the updates are taking place at times $t_1, t_2, \cdots$.
- The interference experienced by c_i caused by all the other clusters is

  $$I_{c_i}(N, \{d_{ij}\}, l) = \sum_{j \neq i} \frac{KP_0}{d_{ij}^{\eta}} \delta(s_i(l), s_j(l))$$

  where l denotes the update time t_l, $l = 1, 2, \cdots$.
- The aggregate interference of the network at time l is

  $$I(N, \{d_{ij}\}, l) = \sum_i I_{c_i}(N, \{d_{ij}\}, l) = \sum_i \sum_{j \neq i} \frac{KP_0}{d_{ij}^{\eta}} \delta(s_i(l), s_j(l))$$

- Note that this channel model is not necessary for the convergence of our algorithm (our algorithm works with any other channel model as long as it is reciprocal).

Similar Problems and Existing Approaches

- There are a number of proposed solutions to similar problems in different contexts (graph coloring, iterative waterfilling, etc.)
- These approaches have either of these drawbacks:
  - Excessively simplifying the interference models
  - Not fully decentralized
  - Require too much information exchange between autonomous nodes/clusters
  - Too complex to implement
  - Suffer from all the above shortcomings

Similar Problems and Existing Approaches (cont.)

- C. Peng, H. Zheng, and B. Y. Zhao [2006] propose that secondary users choose their spectrum according to their information about their local primary and secondary neighbors.
- Nodes are the vertices of a graph and any two interfering nodes are connected with an edge. This turns the problem into the graph multi-coloring problem.
- A sub-optimal solution to the graph multi-coloring, using an approximation algorithm to the graph labeling problem.
- Drawbacks:
  - Not fully decentralized
  - The interference model is excessively simplified
  - Too much message-passing among the nodes
  - High complexity

Similar Problems and Existing Approaches (cont.)

- Similar works in the context of Digital Subscriber Lines (DSL).
- W. Yu, G. Ginis, and J. M. Cioffi [2002] have proposed a method of iterative waterfilling in order to solve the problem of optimal PSD shaping in DSL applications.
- Each user must know a weighted sum of the PSD of the other users (interference), in order to do waterfilling.
- Drawbacks:
  - High computational complexity.
  - Nash equilibrium point does not necessarily correspond to the optimal answer.
    - For instance, in a two-user scenario, if both users start with a flat PSD initially, iterative waterfilling does not change their PSD.
    - This is clearly a Nash equilibrium point, but is far away from the optimal answer.

Similar Problems and Existing Approaches (cont.)

- R. Cendrillon, J. Huang, M. Chiang, and M. Moonen [2006], [2007] consider the problem for a DSL system with N users and K tones.
- The achievable bit-rate of user n is

$$R^n \triangleq \sum_{k=1}^{K} \log \left(1 + \frac{s_k^n}{\sum_{m \neq n} \alpha_k^{n,m} s_k^m + \sigma_k^n}\right)$$

  where s_k^n is the transmission power of user n over tone k, $\alpha_k^{n,m}$ is the normalized cross-talk channel between users n and m and σ_k^n is the noise level of tone k for user n.
- The optimization problem is:

$$\max_{s_k^n, \forall n,k} \sum_i w_i R^i \quad s.t. \quad \sum_k s_k^n \leq P^n, \forall n$$

  for a given set of $0 \leq w_1, \cdots, w_N \leq 1$ such that $\sum_i w_i = 1$.
- This problem can be solved iteratively in a centralized fashion and converges to the optimal values.
- However, it is very complicated due to being centralized.

Similar Problems and Existing Approaches (cont.)

- Very hard to solve in a decentralized manner.
- The optimization problem is relaxed based on introducing a virtual user with fixed thresholds.
- The throughput of the virtual user (from the viewpoint of user n) is

$$R^{n,ref} \triangleq \sum_k \log \left(1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k} \right)$$

  where $\tilde{s}_k$ is a fixed power assignment over tone k for the virtual user, $\tilde{\sigma}_k$ is the noise over tone k, and $\tilde{\alpha}_k^n$ is the cross-talk channel of user n and the virtual user over tone k.
- The relaxed optimization problem for each user n:

$$\max_{s_k^n \forall k, w^n} w^n R^n + (1 - w^n) R^{n,ref} \quad s.t. \quad \sum_k s_k^n \leq P^n$$

  where the maximization is jointly over w^n and s_k^n, $k = 1, \cdots, K$.

Similar Problems and Existing Approaches (cont.)

- Each user solves the relaxed optimization problem locally across different tones. The knowledge of a weighted sum of the PSD of the other users (interference) is required.
- The convergence is proved only in high SNR regime.
- The achievable region resulted by $\sum_i w_i R^i$ over all the values of $0 \leq w_i \leq 1, \forall i$ such that $\sum_i w_i = 1$ is close to the achievable region of the optimal centralized solution.
- No one-to-one correspondence between the points of the achievable regions of the optimal (centralized) and decentralized algorithms.
- The algorithm does not necessarily converge to optimal values.

Similar Problems and Existing Approaches (cont.)

- For the case of asynchronous transmission (in the presence of ICI), the optimization can not be separated across the tones. They have therefore used heuristic optimization approaches with no convergence guarantee.
- Drawbacks:
  - Simplified model for the coupling of the users
  - Stringent constraints for the uniqueness of the Nash equilibrium point
  - The convergence is only proved in high SNR regime
  - No guarantee on the optimality

Contributions of Our Work

- Our proposed dynamic frequency allocation algorithm is fully distributed
  - No information exchange between autonomous devices is needed
  - No knowledge of the existence of other autonomous entities is required
- The proposed algorithm is simple and has low computational complexity.
- It can be used in conjunction with any realistic wireless radio channel model such as those commonly employed in wireless standards (e.g., Hata model, Okumura model, etc.)
- Convergence of this algorithm to a sub-optimal solution is proved
- We have established performance bounds showing that this sub-optimal solution is near-optimal under various practical node activity models.

Assumptions

- At each time slot for any cluster *at most* one user is transmitting and one user is receiving.
    - Alternative scenarios are possible, e.g., users transmit and receive through the cluster head.
    - Can be relaxed to any reciprocal channel model between clusters.
- The distances between clusters are much larger than the size of clusters and bounded below by a distance δ.
- The rate of change of the spatial distributions of the clusters in the network and the underlying channels is much less than the processing/transmission rate.

Assumptions (cont.)

- Each user transmits with power KP_0, where K is a function of frequency.
  - This assumption can be relaxed.
- Path loss with exponent η. No shadowing and fading is assumed.
- r different accessible transmission bands, $b_1, \cdots, b_r$.
- At time t, the ith cluster is in state $s_i(t) \in \{1, 2, \cdots, r\}$, corresponding to the index of the transmission band it is using.
- Performance metric: Aggregate interference of the network.

Outline

1. Intoduction
2. The Algorithm
3. Main Results
4. Simulation Results
5. Time-varying Case
6. Conclusion

Main Algorithm

Main Algorithm: *Clusters scan all the frequency bands* $b_1, \cdots, b_r$ *in an asynchronous manner over time. Each cluster chooses the frequency band in which it experiences the least aggregate interference from other clusters.*

- The cluster head scans all the frequency bands and estimates/measures the interference it experiences in each frequency band.
- The cluster head chooses the new transmission frequency band.

An Example of The Update Process

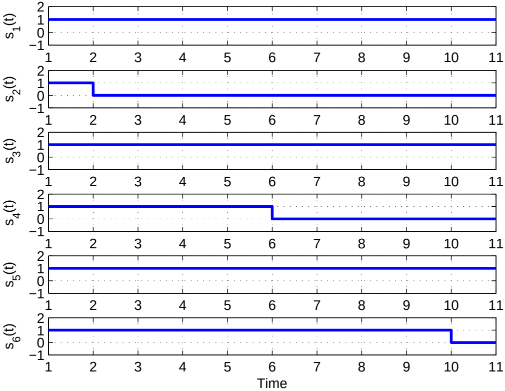


Figure: States vs. time for 6 clusters located equidistantly on a line

- The algorithm converges to the optimal configuration in this example.

Outline

1. Intoduction
2. The Algorithm
3. Main Results
4. Simulation Results
5. Time-varying Case
6. Conclusion

Convergence

Theorem

Given any reciprocal channel model, the Main Algorithm converges to a local minimum in polynomial time in N.

Outline of the Proof

- Clearly, $I(N, \{d_{ij}\}, l) \geq 0$ for all l.
- Before an update at time l, we have

$$I(N, \{d_{ij}\}, l) = \sum_{h} \sum_{c_m \in C_h, m \neq i} I_{c_m,h}(N, \{d_{ij}\}, l) + 2I_{c_i,j}(N, \{d_{ij}\}, l)$$

- c_i switches from band b_j to b_k. Therefore,

$$I(N, \{d_{ij}\}, l+1) = \sum_{h} \sum_{c_m \in C_h, m \neq i} I_{c_m,h}(N, \{d_{ij}\}, l) + 2I_{c_i,k}(N, \{d_{ij}\}, l)$$

- Clearly, $I_{c_i,k}(N, \{d_{ij}\}, l) \leq I_{c_i,j}(N, \{d_{ij}\}, l)$, due to the decision criterion.
- Therefore, $I(N, \{d_{ij}\}, l+1) \leq I(N, \{d_{ij}\}, l)$.

Outline of the Proof (cont.)

- $I(N, \{d_{ij}\}, l)$ is a positive non-increasing function of l. Therefore, the algorithm converges.
- The least amount of decrement in each update step is $O(\frac{1}{N^\eta})$.
- The maximum of aggregate interference if $O(N)$.
- The algorithm converges in polynomial time in N.

Upper Bound on the Performance

Theorem

Let $I_a(N, \{d_{ij}\})$ *denote the aggregate interference of all the clusters corresponding to the state of the algorithm following convergence and* $I_w(N, \{d_{ij}\})$ *to be the aggregate interference for the worst case interference scenario (all clusters transmitting in one frequency band), then*

$$I_a(N, \{d_{ij}\}) \leq \frac{1}{r} I_w(N, \{d_{ij}\})$$

- Proof outline:
  After convergence, c_i is in a frequency band, say $k \in \{1, 2, \cdots, r\}$ such that $I_{c_i,k}(N, \{d_{ij}\}) \leq I_{c_i,j}(N, \{d_{ij}\})$, for all $j \neq k$. Therefore,

$$rI_a(N, \{d_{ij}\}) = r \sum_i I_{c_i,k}(N, \{d_{ij}\}) \leq \sum_i \sum_j I_{c_i,j}(N, \{d_{ij}\}) = I_w(N, \{d_{ij}\})$$

Lower Bound for the Optimal Strategy (For Linear Arrays)

Definition

A Linear Array is an array of clusters for which all the clusters are co-linear (lie on a line).

Theorem

For a Linear Array of clusters in $[0,(N-1)d]$, we have

$$\lim_{N\to\infty} \frac{1}{N} I_o(N,\{d_{ij}\}) \geq \frac{1}{r^\eta} 2\zeta(\eta) \frac{\tilde{P}}{d^\eta}$$

where $I_o(N,\{d_{ij}\})$ is the aggregate interference of the optimal strategy, $\tilde{P}$ is the transmission power of one node and $\zeta(\eta)$ is the Riemann zeta function.

Outline of the proof

- Motivation: The optimal frequency band assignment strategy is not known for general linear arrays. We therefore try to lower bound the aggregate interference of the optimal assignment.
- It can be shown that the minimum aggregate interference of any linear array of N clusters in $[0,(N-1)d]$, is higher than that of the corresponding *uniform* linear array, where the N clusters are located in $[0,(N-1)d]$ equidistantly, for N large enough.
- The righthand side corresponds to the minimum normalized aggregate interference of a uniform linear array.
- Uniform linear array achieves the bound, for N large enough.
- The inequality holds for any linear array.

The case of $r = 2$

Theorem

If $r = 2$ and $\eta \geq 2$, then the optimal strategy for a uniform linear array is the alternating assignment of the two frequency bands, for any N.

- Outline of the proof:
  - For $\eta \geq 2$, there can not be any 3 successive clusters in the same frequency band, in the optimal configuration.
  - There is a sequence of changes in the assignments for any given configuration, which results to the alternating assignment and in each step, the aggregate interference decreases.

Performance Result for Linear Arrays

Corollary

For a linear array in $[0, (N-1)d]$, *we have*

$$\frac{I_a(N, \{d_{ij}\})}{I_o(N, \{d_{ij}\})} \leq \frac{r^{\eta-1}}{\left(\frac{d_{min}}{\min\{d_{max}, d\}}\right)^{\eta}}$$

as $N \to \infty$.

- Proof outline:
  - Combining the previous theorems.
  - It is a worst-case bound. Applicable to any linear array.

Remarks

- The optimal strategy is non-trivial for a general uniform array of clusters.
- The alternating strategy for finite N and $r > 2$ seems to be the optimal strategy, although it is not trivial.
- The Riemann zeta function in the lower bound expression is merely a consequence of the path loss model and the fact that at each time slot, only one transmitter and one receiver are active in each cluster.

Outline

1. Intoduction
2. The Algorithm
3. Main Results
4. Simulation Results
5. Time-varying Case
6. Conclusion

Simulation Setup

- Simulations for $r = 2$, $d = 1$ and $\eta = 2$.
- We let all the clusters to be in frequency band b_1 initially.
- The update starts from the leftmost cluster all the way to the rightmost and continues in a circular fashion. We repeat the updates until the convergence is achieved.
- The ith cluster is distributed uniformly on the interval $[i - 0.5 + G, i + 0.5 - G]$, where G denotes the guard-band to restrict the minimum distance of two adjacent clusters.

Simulation Results: 1D Linear Arrays

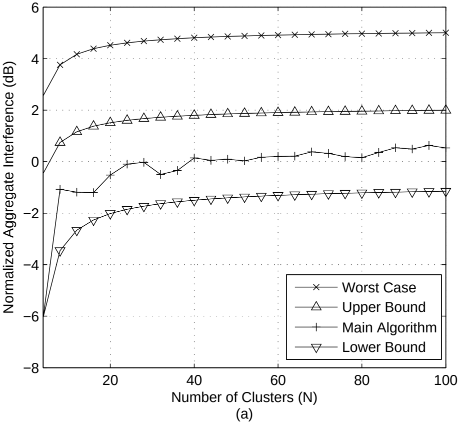


Figure: Normalized Aggregate Interference (dB) vs. N for a uniform linear array

The algorithm performs within 1.5 dB of the optimal strategy (significantly better that the upper bound of 3 dB obtained in the performance results).

Simulation Results: 1D Linear Arrays

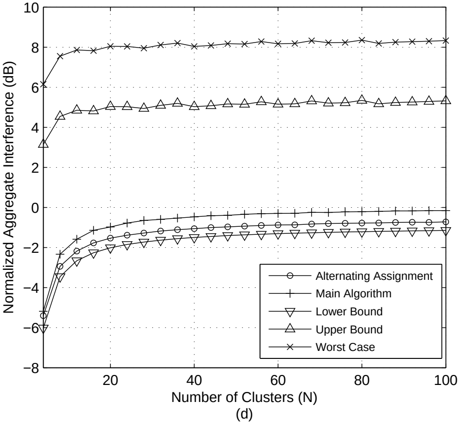
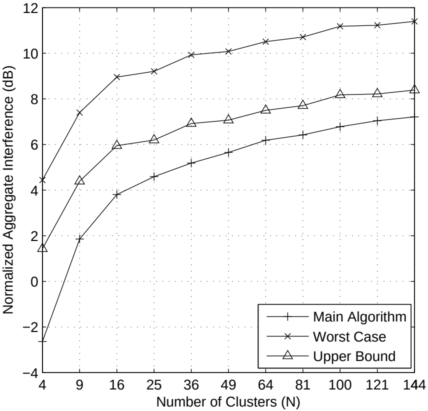


Figure: Normalized Aggregate Interference (dB) vs. N for a non-uniform linear array with $G = 0.02$

Here, the assumption on the size of the clusters being much smaller than the distances between them is relaxed ($G = 0.02$). The algorithm still performs within 1.5 dB of the optimal strategy (significantly better that the upper bound of 3 dB obtained in the performance results).

Simulation Results: 2D Array



(a)

Figure: Normalized Aggregate Interference (dB) vs. N for a non-uniform 2D array with $r = 2$ and $G = 0.05$ per dimension

The clusters are distributed uniformly around the sites of the integer lattice $\mathbb{Z}_2$. No analytical bounds established. The algorithm still performs within the derived upper bound.

Simulation Results: 3D Array

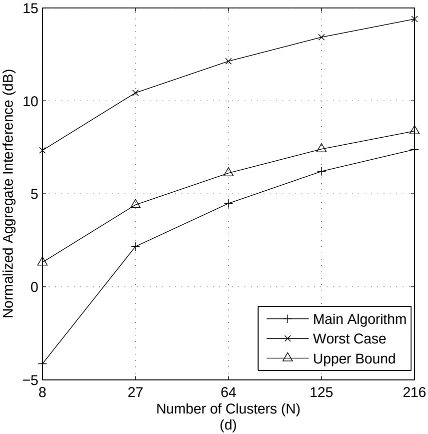


Figure: Normalized Aggregate Interference (dB) vs. N for a non-uniform 3D array with $r = 4$ and $G = 0.05$ per dimension

The clusters are distributed uniformly around the sites of the integer lattice $\mathbb{Z}_3$. No analytical bounds established. The algorithm still performs within the derived upper bound.

Outline

1 Intoduction

2 The Algorithm

3 Main Results

4 Simulation Results

5 Time-varying Case

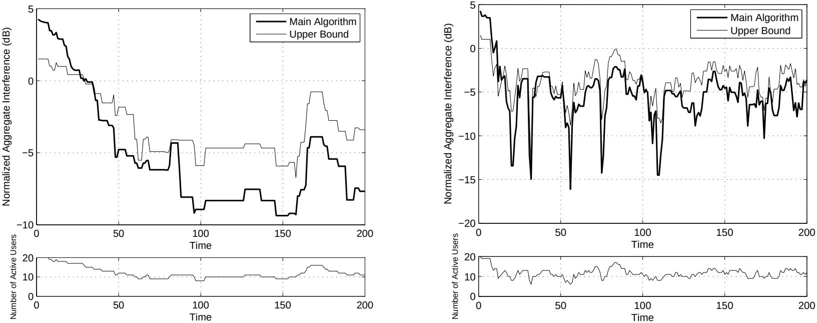
6 Conclusion

Time-varying setup

- The clusters go on and off over time, according to a two-state Markov model.
- For cluster c_i, $i = 1, 2, \cdots, N$, we consider an activity indicator state $a_i(l)$, such that $a_i(l) = 1$ and $a_i(l) = 0$ correspond to being active and inactive at time l, respectively. Let $P_0^{c_i}(l)$ and $P_1^{c_i}(l)$ be the probability of c_i being in activity indicator state 0 and 1 at time l, respectively. The evolution of the probabilities is given by:

$$\begin{pmatrix} P_0^{c_i}(l+1) \\ P_1^{c_i}(l+1) \end{pmatrix} = \begin{pmatrix} \alpha & 1-\alpha \\ 1-\alpha & \alpha \end{pmatrix} \begin{pmatrix} P_0^{c_i}(l) \\ P_1^{c_i}(l) \end{pmatrix}$$

Simulation Results: Time-varying setup



(a)

(b)

$\alpha = 0.99$

$\alpha = 0.95$

The algorithm still performs within the upper bound, almost all the time.

Assumptions

- The convergence of the algorithm is not guaranteed under the mentioned time variations.
- We assume that the clusters update their frequency band asynchronously according to the same temporal statistics.
- In a continuous approximation, let $I(t)$ denote the aggregate interference of the network at time t.
- The update process is modeled by a Poisson process of rate $\frac{1}{\Delta T}$, *i.e.*, each cluster updates its frequency band with a rate $\frac{1}{N\Delta T}$.
- For the moment, we assume that all the N clusters are active.
- We associate $\epsilon_i = -1$ and $\epsilon_i = 1$ to clusters in band b_0 and b_1, respectively.
- All the following analysis is valid for steady state (near equilibrium) and assuming that the number of users switching on/off in each time slot is much less than the total number of users.

Dynamics of the Algorithm for $r = 2$

- An active cluster experiences an interference of $\frac{1}{2}(I_i - \sum_{j \neq i} \frac{\epsilon_j(t)}{d_{ij}^{\eta}})$ or $\frac{1}{2}(I_i + \sum_{j \neq i} \frac{\epsilon_j(t)}{d_{ij}^{\eta}})$ depending on the band it is using, where I_i is the worst case interference experienced by cluster c_i.
- We define the band b_j to be *appropriate* for cluster c_i, if c_i is assigned in band b_j in the optimal strategy.
- If a total of M users c_i, $i \in \{i_1, \cdots, i_M\}$ are not in the *appropriate* frequency bands, the aggregate interference will be

$$I(t) = I_a + \sum_{k=1}^{M} |\sum_{j \neq i_k} \frac{\epsilon_j(t)}{d_{i_k,j}^{\eta}}|$$

  where I_a is the target performance of the algorithm.
- Assuming Ergodicity, we have

$$\mathcal{E}[I(t)] - \mathcal{E}[I_a] = M\mathcal{E}\Big[|\sum_{j \neq i} \frac{\epsilon_j(t)}{d_{ij}^{\eta}}|\Big]$$

  where $\mathcal{E}$ denotes the ensemble average.

Dynamics of the Algorithm for $r = 2$

- For any update, the average change in $\mathcal{E}[I(t)]$, will be

$$\Delta\mathcal{E}[I(t)] = \frac{\rho M}{N}\mathcal{E}\Big[\Big|\sum_{j\neq i}\frac{\epsilon_j(t)}{d_{ij}^{\eta}}\Big|\Big]$$

  where ρ is a geometrical constant showing the effective number of interacting neighbors to a cluster including itself (this is a linearization near the equilibrium point).

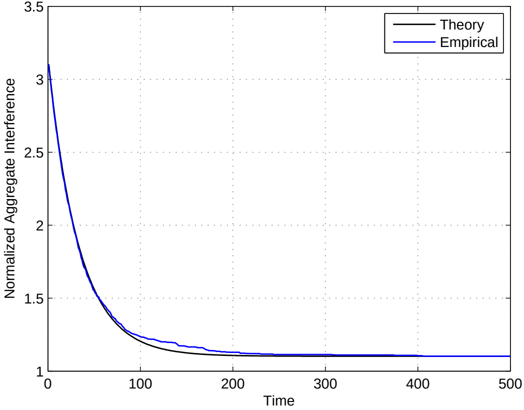
- Combining the above equations we get

$$\frac{\Delta\mathcal{E}[I(t)]}{\Delta T} = -\frac{\rho}{N\Delta T}(\mathcal{E}[I(t)] - \mathcal{E}[I_a])$$

- Spatial ergodicity is assumed in the derivation of the above dynamics.
- On the time scale of the updates, using the ansatz $N\Delta T \triangleq \tau$ one can write

$$\frac{d\mathcal{E}[I(t)]}{dt} = -\frac{\rho}{\tau}(\mathcal{E}[I(t)] - \mathcal{E}[I_a])$$

Simulation Results: Dynamics of the Algorithm for a Uniform Linear Array of 100 clusters



- A theoretical estimate for ρ in this case is 3, since every cluster has two nearest neighbors.

Time-varying Statistics

- We can model the change in the number of active clusters by two Poisson counters of rate λ.
- Each cluster when activated, approximately experiences the instantaneous normalized aggregate interference of the network (Ergodicity).
- If we define $\mathcal{I}(t) \triangleq \mathcal{E}[I(t)]$ and $\mathcal{I}_a(t) \triangleq \mathcal{E}[I_a(t)]$, under the assumption of λ being small compared to $\frac{1}{\tau}$, we have the new dynamics in the Itō form as

$$d\mathcal{I}(t) = -\frac{\rho}{\tau}(\mathcal{I}(t) - \mathcal{I}_a(t))dt + \frac{4}{N}\mathcal{I}(t)(dN_+ - dN_-)$$

  where dN_+ and dN_- are two independent Poisson counters of rate λ, *i.e.*, $E(dN_+) = E(dN_-) = \lambda dt$.
- Under this model, the algorithm converges in mean to the target performance.

Steady State Analysis

- The variance equation associated with the dynamics is

$$\frac{dE(\mathcal{I}^2(t))}{dt} = -\Big(\frac{2\rho}{\tau} - \frac{32\lambda}{N^2}\Big)E(\mathcal{I}^2(t)) + \frac{2\rho}{\tau}\mathcal{I}_a^2$$

- In the steady state, the variance settles down to

$$\sigma_{ss}^2 = \mathcal{I}_a^2 \frac{16\lambda\tau}{N^2\rho - 16\lambda\tau}$$

given $\frac{16\lambda\tau}{N^2\rho} < 1$.

- According to our model for cluster activities, $\lambda = N^2(1-\alpha)/2\tau$. Thus, we get the following trade-off inequality

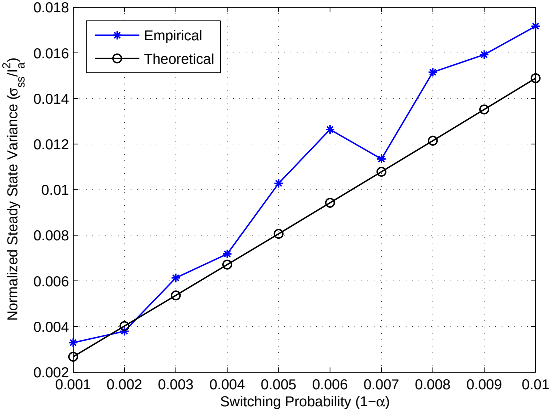
$$\frac{8(1-\alpha)}{\rho} < 1$$

in order to have a finite variance in the steady state.

Remarks

- The model gives a simple trade-off inequality for design purposes.
- The geometrical parameter ρ can be empirically estimated for different network topologies. However, theoretical estimates are possible.
- If $\lambda = O(N^{1-\epsilon})$ for some $\epsilon > 0$, as $N \rightarrow \infty$, the inequality always holds. Therefore, the algorithm converges in both mean and variance in the sub-linear regime.
- The analysis can be generalized to other statistical models for the activity of the clusters over time.

Simulation Results: Empirical vs. Theoretical Steady State Variance



- The model matches the empirical data (averaged over 500 ensembles).

Outline

1. Intoduction
2. The Algorithm
3. Main Results
4. Simulation Results
5. Time-varying Case
6. Conclusion

Conclusion

- Proposed a distributed algorithm for finding a sub-optimal frequency band allocation to the clusters in a network.
- Proved the convergence of the algorithm for any reciprocal channel model.
- Obtained performance bounds for one dimensional linear arrays of clusters (the algorithm outperforms the bounds).
- Evaluated the performance of the algorithm when clusters can be in sleep or active mode and go off and on according to time-varying statistics.

Future Work

- Finding performance bounds for more general network topologies.
- Generalization of the performance bounds to higher dimensions.
- Finding a modified algorithm for the scenario that clusters can be in sleep or active mode and go off and on according to time-varying statistics (Open loop control).
- Distributed control methods, to control the performance of the algorithm, in the presence of time-varying statistics (Closed loop control).