A Prediction and Learning Based Approach to Network Selection in Dynamic Environments

Xiaohong Li[1], Ru Cao[1], Jianye Hao[2(✉)], and Zhiyong Feng[2]

[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
[2] School of Software, Tianjin University, Tianjin, China
jianye.hao@tju.edu.cn

Abstract. The heterogeneous property in the next generation wireless network arises challenges of network selection problem. Existing approaches are mainly implemented in static network environments while cannot handle unpredictable dynamics in practice. In this paper, we propose a prediction and learning based approach, which considers both the fluctuation of radio resource and the variation of user demand. The network selection scenario is modeled as a multiagent coordination problem, in which a population of rational agents compete to maximize their benefits with incomplete information (no prior knowledge of network bandwidth and other users' demands). Terminal users adaptively adjust their selections in response to the gradually or abruptly changing environment. The system is shown to converge to Nash equilibrium, which also turns out to be both Pareto optimal and socially optimal. Extensive simulation results show that our approach achieves significantly better performance compared with two existing approaches in terms of load balancing, user payoff and the overall bandwidth utilization efficiency.

1 Introduction

The next generation wireless network is envisioned as a heterogeneous network (HetNet) environment consisting of a variety of overlapping radio networks (e.g., WPAN, WLAN, WMAN) with various technologies [3]. Within the HetNet environment, there is an overwhelming growth in the number of terminal users and their varying bandwidth demands, meanwhile, network resource is limited and may change dynamically due to the interferences from intrinsic or extrinsic factors (noises, channel interferences, natural disturbances, etc.). How to achieve a good balance between increasing user demand and dynamically changing radio resource in multi-user, multi-provider HetNet environments is challenging.

To tackle this challenge, effective techniques are required to help select the most appropriate network from all available candidates to satisfy specific requirements. Commercial solutions usually involve rudimentary static network selection policies (e.g., always select the WLAN, always select the cheapest or the fastest network) [11]. However, varying network characteristics and user preferences are omitted, which may often result in lower quality of service (QoS). Many traditional methods in research literatures use multi-attribute decision making algorithms (e.g., SAW, TOPSIS, ELECTRE, AHP&GRA) to evaluate and rank

A. Lintas et al. (Eds.): ICANN 2017, Part I, LNCS 10613, pp. 92–100, 2017.
https://doi.org/10.1007/978-3-319-68600-4_12

candidate networks in a preference order to guide the selection process [7]. This may cause congestion when all users connect the so-called "best" network. Learning based methods are promising candidate solutions to model the network selection problem. A channel selection and Routing approach is proposed in [1] which models the problem as Markov decision process to design the method of learning the best resource allocation policies. Q-learning is used in [12] to maximize the total reward in network selection decision. Reinforcement learning model is used in [6] to find the best strategy to maximise the reward function expressed in terms of call blocking and call dropping probabilities. Unfortunately, the above works suffer from the following two limitations: (1) requiring too much state information (the number of users, future bandwidth, etc.) as a prior, which is costly or impractical; (2) only focusing on the static resource without considering changing characteristics in practical environments.

To address the above problems, we model the network selection process as a multiagent coordination problem, in which a population of rational terminal users compete to select the "best" access networks to satisfy their varying demands with incomplete information (no prior knowledge of changing bandwidth and other users' demands) within a dynamic HetNet environment. Our approach is user-centric but does not require any central controller or additional communications between users. The only information available to users is the previous load and provided bandwidth of their connected networks. In addition, our approach is robust against failures of users: when they occasionally join or leave, the system can self-organize quickly and adapt to a newly created environment.

Simulation results show that the system guarantees convergence towards Nash equilibrium, which is also proved to be Pareto optimal and socially optimal. Extensive results demonstrate that our algorithm enables users to adaptively adjust their selections in response to the change of bandwidth, and it significantly outperforms either the learning or non-learning based approach in terms of load balance, user payoff and the overall bandwidth utilization efficiency.

2 Network Selection Problem Definition

In HetNet environments, radio resource may loss or be disturbed in transmission process by various factors (network topology, routers, base stations, noises, channel interferences, etc.) which greatly impact network performances [4,10]. Therefore, the available bandwidth of each base station allocated from its core network dynamically changes and is less than the nominal value due to many influence factors. In such dynamic environments, we assume that each user can only have access to the state information of the base station it connected from completed interactions and is lack of prior knowledge of any other networks or terminal users. The cooperation between the user and its connected base station is helpful and does not infringe upon any other's interest.

2.1 Multiagent Network Selection Model

In practice, each user makes independent decisions based on its local information only. However, actions taken by users influence the actions of others indirectly. Therefore, we model the problem as a multiagent coordination problem, in which a population of rational users located in the same or different service areas with no information about others learn to compete to maximize their payoffs given that available bandwidth varies dynamically. Formally, the multiagent network selection is modeled as a 6-tuple $< BS, B_k(t), U, b_i(t), A_i, P_i(t, \mathbf{a}) >$, where:

- $BS = \{1, 2, ..., m\}$ is the set of available base stations (BS).
- $B_k(t)$ denotes the provided bandwidth of BS $k \in BS$ at time t.
- $U = \{1, 2, ..., n\}$ is the set of terminal users.
- $b_i(t)$ denotes the bandwidth demand of user $i \in U$ at time t.
- $A_i \subseteq BS$ is the finite set of actions available to user $i \in U$, and $a_i \in A_i$ denotes the action (i.e., selected base station) taken by user i.
- $P_i(t, \mathbf{a})$ denotes the expected payoff of user $i \in U$ by performing the strategy profile $\mathbf{a} = \{a_1, ..., a_i, ...a_n\} \in \times_{j \in U} A_j$ at time t.

There are n users competing for m base stations in the system. The detail definition of payoff based on the joint strategy profile $\mathbf{a}$ can be expressed as,

$$P_i(t, \mathbf{a}) = \frac{w_i(t, \mathbf{a})}{b_i(t)}, \quad w_i(t, \mathbf{a}) = \begin{cases} b_i(t), & \sum_j b_j(t) \leq B_{a_i}(t) \\ \frac{B_{a_i}(t) \cdot b_i(t)}{\sum_j b_j(t)}, & otherwise \end{cases} \tag{1}$$

where $w_i(t, \mathbf{a})$ is the perceived bandwidth (a theoretical value without considering the transmission loss) of user i at time t, and $j \in \{j \in U | a_j = a_i,\ a_j, a_i \in \mathbf{a}\}$ is the user who connect the same base station with user i.

2.2 Theoretical Analysis

Nash equilibrium (NE) is the most commonly adopted solution concept in game theory. Under a NE, no player can benefit by unilaterally deviating from its current strategy [11]. Underlying the multiagent network selection problem, a NE is reached when there is no overload on any base station (this situation is shown in later experiments). Under this condition, users' perceived bandwidth equals to their demands and all users' payoffs reach maximum. Therefore, no one is willing to change its strategy given that others' strategies are unchanged.

Definition 1. $\mathbf{a}^* \in \times_{i \in U} A_i$ *is a Nash equilibrium if for all* $k \in BS, \sum_j b_j(t) \leq B_k(t)$, *where* $j \in \{j \in U | a_j = k, a_j \in \mathbf{a}^*\}$.

However, a NE may not be desirable in general since it may not necessarily correspond to the maximization of the system-level payoff. Fortunately, any NE in our model is also Pareto optimal and socially optimal [11]. The two properties guarantee both the system's stability and system-level optimization.

Theorem 1. *Nash equilibrium, Pareto optimality and Social optimality are equivalent in the multiagent network selection problem.*

Proof. It can be deduced that if profile $\mathbf{a}^*$ is a NE, each user's payoff reaches maximum and cannot be further increased. Therefore, it's impossible to find another outcome under which no user's payoff is decreased while at least one user's payoff is strictly increased. This proves that $\mathbf{a}^*$ is Pareto optimal. In addition, $P_i(t, \mathbf{a}^*) = \max P_i(t, \mathbf{a}) \Rightarrow \sum_i P_i(t, \mathbf{a}^*) = \max \sum_i P_i(t, \mathbf{a}), \forall \mathbf{a} \in \times_{j \in U} A_j$. The sum of all users' payoffs reaching maximum means $\mathbf{a}^*$ is also socially optimal.

3 Multiagent Network Selection Strategy

A user's network selection strategy consists of two steps: selection and evaluation. In selection procedure, the user learns to choose the best candidate network to satisfy its special demand. Once the selection procedure is completed, evaluation procedure will be triggered to update its strategy.

3.1 Selection

Algorithm 1 summarizes the selection procedure for user $i \in U$. For each available base station $k \in BS$, the user checks whether it can satisfy its special demand (Lines 1–11). If the user sends a connection request to a base station with no historic information, which is the standard case at the beginning of the life-cycle, this unpredictable base station will be added in a spare list for a later decision. Otherwise, the user predicts the possible bandwidth and load on the base station. If the predicted load plus the demand is below the predicted bandwidth, this base station is added to the list of candidates (Lines 7–9). Then the user evaluates if any candidate base station is expected. There might be three cases. In the case where the list of candidate base stations predicted having adequate bandwidth available is not empty (Line 12), the "best network selection" is determined by the following policy: the base station with most expected free bandwidth is chosen as the most appropriate connection currently. In particular, in the case there is no available candidate, the user will randomly explore one from all unpredictable base stations and gather its state information (Line 17). There might be an exceptional case that no base station is generated from the algorithm (Line 19). In this case, the original base station is used and $flag$ is set into -1.

Each user maintains a historic information table $table_k = (h_0, ..., h_p), (0 \leq p < m)$ for each connected base station k. The table is composed of up to m items $h_j = (t_j, load_j, bw_j)$, comprising observed time t_j, load $load_j$ and bandwidth bw_j. The oldest item will be overwritten if already m items are recorded. Load prediction mechanism employs time series forecasting techniques to predict future load value based on records of this table. It involves three major steps:

- Create predictor set. Each user keeps a set of r predictors $P(A,k) = \{p_i | 1 \leq i \leq r\}$, which is created from some predefined set in evaluation procedure (following case 1), for each available base station k. Each predictor is a predictive function from a time series of historic loads to a predictive load value.
- Select active predictor. One predictor $p^A \in P$ is called active predictor, which is chosen in evaluation procedure (following case 2, 3), used in load prediction.
- Make prediction. Predict the possible load of the base station via the active predictor and the historic load records.

A similar prediction mechanism can also be adopted to bandwidth prediction.

3.2 Evaluation

Evaluation procedure introduced in Algorithm 2 is divided into three cases based on the selected base station.

Algorithm 1. Selection

```
1: for all k ∈ BS do
2:   if table_k = ∅ then
3:     push k in unpredList
4:   else
5:     predLoad ←LoadPredict(p^A)
6:     predBW ←BWPredict()
7:     if predLoad + b_i ≤ predBW then
8:       push k in candList
9:     end if
10:  end if
11: end for
12: if candList ≠ ∅ then
13:   for all cand ∈ candList do
14:     availBW = predBW − predLoad
15:   end for
16:   seleBS ← argmax_{k∈BS}(availBW)
17: else if unpredList ≠ ∅ then
18:   seleBS ←random(unpredList)
19: else
20:   seleBS ← lastBS // stay at last BS
21:   flag = −1
22: end if
```

Algorithm 2. Evaluation

```
1: if predictorSet = ∅ then
2:   create predictorSet for seleBS
3:   p^A ←random(predictordSet)
4:   update(table_seleBS)
5: else if flag = −1 then
6:   for all k ∈ BS do
7:     delete h ∈ table_seleBS with a probability
8:   end for
9: else
10:  for all p ∈ predictorSet do
11:    predLoad ←LoadPredict(p)
12:    r_p = 1 − |load−predload| / load
13:    Q_p = (1 − α)Q_p + αr_p
14:  end for
15:  p^A ←BoltzmanExploration(predictordSet)
16:  //abruptly changing case
17:  if |B_seleBS − predBW| > Δ then
18:    clear(table_seleBS)
19:  end if
20:  update(table_seleBS)
21: end if
```

Case 1. If the selected base station is visited for the first time (Line 1), the user will create a new predictor set for this base station and record its load and bandwidth information into the corresponding record table. All predictors in the set are chosen randomly from a predefined set, hence users' predictor sets may be different from each other. The predefined set contains multiple types of forecasting functions which differ in window sizes (e.g., average method, linear regression, exponential smoothing, etc.) [2]. Different types of predictors are suitable for different situations and environments.

Case 2. If $flag = -1$, it implies that currently historical records recommended no appropriate base station (Line 5). In the case, some old records need to be removed from the table according to a probability distribution relative to their lifetime to get more up-to-date information for further predictions.

Case 3. The general situation is that the user switched to a previously visited base station. The evaluation mainly involves two aspects: assessing the performance of all predictors (Line 10) and dealing with the case of abruptly changing bandwidth (Line 17). The assessment of predictors is resorted to Q-learning. Specifically, Q-function in our approach is defined as the following equation,

$$Q_p(t) = (1-\alpha)Q_p(t-1) + \alpha r_p(t-1), \qquad r_p = 1 - \frac{|load - predload|}{load}$$

where $p \in predictorSet$ denotes the predictor, $Q_p(t)$ is the Q-value of p, α is the learning rate, and r_p is the observed reward which denotes the predictive accuracy of p. The predictor which forecasted a more exact value receives a higher reward, else receives a lower reward.

In our approach, Boltzmann exploration mechanism [5] is adapted to explore the active predictor. The probability x_p of selecting predictor p is given by $x_p(t) = \frac{e^{Q_p(t)/T}}{\sum_k e^{Q_k(t)/T}}$, where the *temperature* $T > 0$ balances the tradeoff between exploration and exploitation.

The above process works well in the environment with static or gradually changing bandwidth. However, it is slightly different in abruptly changing case. When detecting that the difference between the observed bandwidth and predicted value of the base station in the last selection is larger than a threshold Δ, the user will consider it encounters abruptly changing environments (Line 17). At catastrophe points, all historic records are invalid and may mislead to inaccurate predictions in future. In order to eliminate the adverse influence and achieve rapid re-convergence, the record table should be cleared out. Then the latest information is added as the only valid record for a later prediction.

4 Performance Evaluation

Parameter settings of our simulated scenario are given in Table 1. We consider a variety of HetNet environments consisting of up to 900 users. On BS_0 and BS_1, the provided bandwidth changes gradually and abruptly. On BS_2, the provided bandwidth keeps static. All experimental results are averaged over 50 independent runs. We make comparisons with two existing multi-user network selection algorithms [8,9] in following aspects.

Table 1. Parameter settings

Access tech	Network rep	Base station	Max bandwidth	User demand
WLAN	Wi-Fi	BS_0	25 Mbps	32 kbps–128 kbps
WMAN	WiMAX	BS_1	50 Mbps	
CDMA Cellular Network	4G	BS_2	5 Mbps	

Load Balancing Analysis. Figure 1 depicts the load situations on the three base stations in static and dynamic environments. We observe that under our prediction and learning based algorithm (PLA), initially, all users randomly selecting their base stations results in high levels of overload or underload on different base stations. However after a few learning interactions, network bandwidth becomes well-utilized without being overloaded. Meanwhile, the load (i.e., total bandwidth demand) on each base station dynamically changes with the amount of provided bandwidth. This implies that the system converges to NE and achieves load balance among the three base stations. It is worth to note that the jitter on BS_2 is because users are trying to join or leave this base station in response to the abrupt changes on the other two base stations.

RAT selection algorithm (RATSA) [8] is similar to the best response, where the user always selects the network with maximum expected bandwidth allocated. Future provided bandwidth, user number on a base station and the number of past consecutive migrations are required as prior knowledge. A user switches its base station only if the value of allocated bandwidth from another base station divided by currently perceive bandwidth is higher than a given threshold η. For fair comparisons, we set $\eta = 1.5$ which gives RATSA the best performance. The comparative figure shows an unbalancing phenomena that too much unmet demand on BS_0 and BS_2, but too little utilization on BS_1 over some time. This indicates that users cannot sense the dynamic environment and adjust their strategies timely. We also simulate the network selection scenario using another Q-learning based approach (QLA) [9]. In QLA, it can be observed that users are trying to adapt to the changing environment. However, it takes a long time to get close to the varying bandwidth and cannot achieve complete load balance.

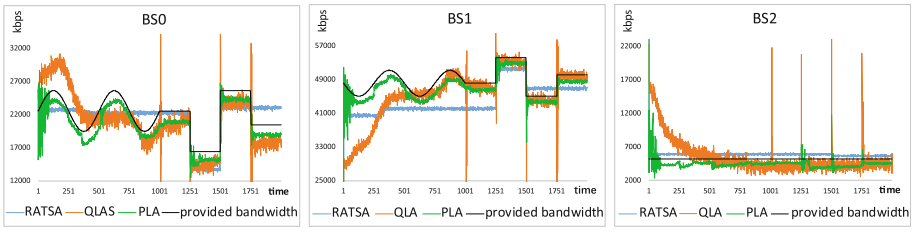

Fig. 1. Load situations on three base stations.

Convergence. In our approach, the system takes a learning phase to achieve convergence, i.e., when there is no overload on any base station (can be observed in Fig. 1), the system converges to Nash equilibrium, which is also Pareto optimal and socially optimal (Definition 1, Theorem 1). In gradually changing case, once it converges to equilibrium, the state sustains over time. We call it first-convergence and the average first-convergence time exponentially increases with the user number from 860 to 900. Specially, in abruptly changing case, when

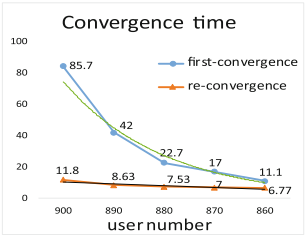

Fig. 2. Convergence time

Alg.	BW Utilization	Switching Rate	User Payoff
PLA	0.947379	0.007388	0.998231
RATSA	0.897151	0.000509	0.954789
QLA	0.902609	0.200071	0.957789

Fig. 3. Comparisons of average performances

encountering catastrophe points, the equilibrium is broken but re-converges in a number of steps. The average re-convergence time linearly varies with the number of users (see Fig. 2).

User Payoff, Switching Rate and Bandwidth Utilization. Comparison results of the three approaches in terms of average user payoff, switching rate and bandwidth utilization are presented in Fig. 3. We observe that over 2000 interactions, PLA outperforms RATSA in average user payoff and bandwidth utilization. The average switching rate of PLA is slightly higher because users are trying to switch their connections to respond to the dynamics to get higher payoffs in the initial phase and at catastrophe points. As for QLA, although we can sense it is trying hard to adapt to the dynamic environments, it gives bad performance of any of the three criteria compared to PLA.

5 Conclusions

In this paper, a prediction and learning based approach is presented to tackle the network selection problem with changing bandwidth in HetNet environments. The performance of the approach is investigated under various conditions and aspects. Extensive experimentations show that the system ideally converges to Nash equilibrium, which also turns out to be both Pareto optimal and socially optimal. Furthermore, our approach significantly outperforms state-of-the-art approaches in terms of load balance, user payoff and bandwidth utilization.

Acknowledgements. This work has partially been sponsored by the National Science Foundation of China (No. 61572349, No. 61272106), Tianjin Research Program of Application Foundation and Advanced Technology (No.:16JCQNJC00100).

References

1. Barve, S.S., Kulkarni, P.: Dynamic channel selection and routing through reinforcement learning in cognitive radio networks. In: IEEE International Conference on Computational Intelligence & Computing Research, pp. 1–7 (2012)
2. Brockwell, P.J., Davis, R.A.: Introduction to Time Series and Forecasting. STS. Springer, Cham (2016). doi:10.1007/b97391

3. Charilas, D.E., Panagopoulous, A.D.: Multiaccess radio network enviroments. IEEE Veh. Technol. Mag. **5**(4), 40–49 (2010)
4. Jain, K., Padhye, J., Padmanabhan, V.N., Qiu, L.: Impact of interference on multi-hop wireless network performance. Wireless Netw. **11**(4), 471–487 (2005)
5. Kianercy, A., Galstyan, A.: Dynamics of Boltzmann q learning in two-player two-action games. Phys. Rev. E **85**(4), 041145 (2012)
6. Kittiwaytang, K., Chanloha, P., Aswakul, C.: CTM-based reinforcement learning strategy for optimal heterogeneous wireless network selection. In: Computational Intelligence, Modelling and Simulation (CIMSiM), pp. 73–78. IEEE (2010)
7. Martinez-Morales, J.D., Pineda-Rico, U., Stevens-Navarro, E.: Performance comparison between madm algorithms for vertical handoff in 4G networks. In: Electrical Engineering Computing Science and Automatic Control (CCE), pp. 309–314. IEEE (2010)
8. Monsef, E., Keshavarz-Haddad, A., Aryafar, E., Saniie, J., Chiang, M.: Convergence properties of general network selection games. In: 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 1445–1453. IEEE (2015)
9. Niyato, D., Hossain, E.: Dynamics of network selection in heterogeneous wireless networks: an evolutionary game approach. IEEE Trans. Veh. Technol. **58**(4), 2008–2017 (2009)
10. Perkins, D.D., Hughes, H.D., Owen, C.B.: Factors affecting the performance of ad hoc networks. In: IEEE International Conference on Communications, vol. 4, pp. 2048–2052 (2002)
11. Trestian, R., Ormond, O., Muntean, G.M.: Game theory-based network selection: solutions and challenges. IEEE Commun. Surv. Tutor. **14**(4), 1212–1231 (2012)
12. Xu, Y., Chen, J., Ma, L., Lang, G.: Q-learning based network selection for WCDMA/WLAN heterogeneous wireless networks. In: 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), pp. 1–5. IEEE (2014)