Cog Gap: Cognitive and Opportunistic Gateway Access for Wireless Mesh Networks

Panlong Yang, Guihai Chen
Computer Science Department
Nanjing University
plyang@computer.org, gchen@nju.edu.cn

Baijian Yang
Department of Technology
Ball State University
byang@bsu.edu

Deke Guo, Yunhao Liu
Computer Science & Engineering Department
Hong Kong University of Science and Technology
guodeke@gmail.com, liu@cse.ust.hk

***Abstract*— The performance of conventional gateway access optimization techniques deteriorate dramatically when traffic load is dynamic. In this article, we propose a novel gateway access algorithm called 'Cog Gap', which is a cognitive method and is designed for Wireless Mesh Networks to maximize the network utilization. The proposed Cog Gap utilizes a destination-hub access model, where multiple gateway nodes are connected by wired links, and packets from a source node can be sent from any connected gateway nodes to increase transmission opportunities. In Cog Gap, we use the Hidden Markov Model (HMM) and the expectation maximization method to handle uncertain traffic pattern and the loss of probing results. A traffic allocation algorithm is then proposed to optimize dynamic multi-gateway access. By modeling the route state determination and transition, we transform the opportunistic gateway access problem into a Markov decision process (MDP) problem. A heuristic and adaptive algorithm named hindsight optimal is used in solving MDP. Simulation results have proven that the proposed Cog Gap algorithm can make full use of the transmission opportunities and does not incur noticeable protocol overhead.**

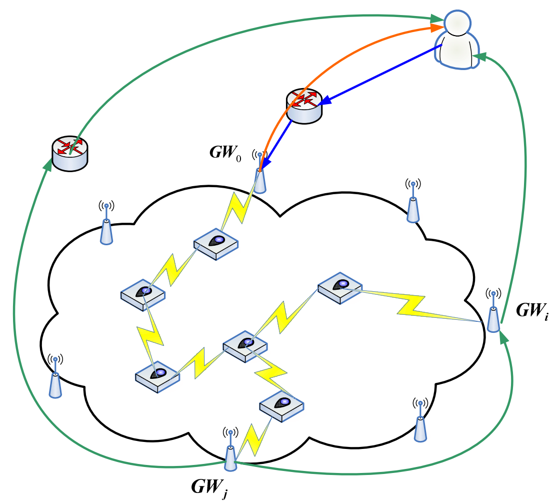

Fig. 1. Destination Hub based gateway access model

I. Introduction

Wireless mesh networks is a promising solution to the "last mile access" problem and mesh routers can achieve higher network throughput due to the existence of multiple communication channels. In this work, we focus on building a gateway access algorithm in wireless mesh networking system by leveraging a cognitive and opportunistic methodology.

Two general design goals for gateway access algorithm are network efficiency and traffic adaptivity. Because all nodes in network tend to send packets to gateway nodes, an efficient gateway access method should avoid the congestion [1] [8] [9] on the gateway nodes. For networks with dynamic traffic, loads vary on different routes. A carefully designed access method should be able to adapt to the traffic load to opportunistically maximize the network utilization.

We have three observations about the gateway access algorithm in Wireless Mesh Networks. First, for a wireless node, there are not much differences as far as which in gateway node it picks to send packets when traffic is not heavy. This is because gateway nodes are connected by wire-line or high speed radio links and the protocol overhead is trivial and the reassembling process at each gateway can ensure the data integrity. As shown in Fig.1, each gateway Gw_i are connected by wired network. Data packets generated by each node v_i can be delivered to any gateway Gw_j, and a reassembling process is running on each gateway. We denote this access mode as a "Destination Hub", because all the gateway nodes are connected as if they were plugged into a logical Ethernet hub.

The next observation is that traffic patterns dominate the throughput of gateway access [1] [8] [11]. Also, traffic patterns are unknown in a real deployed wireless mesh network. Optimal models such as linear programming and mixed integer programming are not working in a bandwidth constrained wireless network due to the big iteration overhead [18]. When dealing with the stochastic traffic patterns and multiple routing choices, a probing process is needed before directing traffic onto different routes.

The last observation is that transmission opportunities vary on different routes. It is therefore necessary for a source node to have a mechanism to send packets by concurrently using multiple efficient routes to maximize the throughput between a source node and a destination node.

Base upon those three observations, we propose our 'Cog Gap' algorithm, which applies a cognitive route level optimization to maximize the opportunities of transmitting packets among different routes to the gateways. We assume the unknown network traffic can be modeled by observing traffic patterns. That is, probing traffic load on different routes to gather information about traffic states and then making inference with a built-in hidden Markov model. In addition, to accurately probe network states with low cost, we employed the proposed interference model on IEEE 802.11 DCF in Qiu's work [6], which has been validated and proven to be an effective modeling technique [5].

Contributions of this study are as follows:

1) We propose a new gateway access model, called "Destination Hub". With this gateway access model, it is possible to have more routes to reach a destination node. In addition, this new model enables dynamic gateway access with variable traffic patterns. We also analyze route probing from a perspective of the optimal stopping theory.
2) The "Cog Gap" applies the IEEE 802.11 DCF [6] modeling to reduce the probing overhead and improve the probing accuracy. It also takes advantages of a validated MAC layer model, and creatively adopts a hidden Markov model (HMM) to deal with the uncertainties in traffic patterns.
3) Our proposed gateway access algorithm is a purely decision based mechanism, which has much less protocol overhead than conventional methods. Routing decisions are made in a decentralized manner by using the inference model of HMM and the validated MAC model.

In summary, we propose a new framework on gateway access in wireless mesh network, which takes advantage of varying network traffic for transmission opportunity. Considering the partially known characteristics of the network traffic, we make our decisions on a MDP (Markov Decision Process) model. The opportunistic transmissions, however, will successfully utilize the throughput gap and add beliefs of our MDP model as observation results.

The rest of the paper is organized as follows. In Section II, related work on gateway access algorithms are introduced, followed by the system model and problem formation in Section III. Next, in Section IV, we propose a model driven method on route level probing. In Section V, we propose a MDP based approach on gateway access problem. We further describe on our cognitive routing method in Section VI, and present the simulation results in Section VII. Last, we conclude our work and future directions in Section VIII.

II. Related Work

Gateway nodes provide seamless interconnections between wireless nodes and wired routers. They have great impact on the performance of a heterogeneous network. Accessing algorithms to maximize the utilization of the gateway nodes have been extensively studied recently [1] [11] [12]. Those traditional methods assumes network traffic is known beforehand, and gateway nodes can be placed everywhere in the network region as needed, which is normally not the case in real deployments.

Scheduling wireless data in transmitting queue is discussed in multi-carrier wireless data systems [12]. Because different carriers have different transmitting rates, a scheduling algorithm is needed in order to enhance the performance of wireless access point. Most, if not all, scheduling algorithms are unable to optimize dynamic traffic in multi-hop wireless networks where traffic pattern is unknown.

Multipath algorithms are primarily used to recover route failures in self-organizing networks. Recently, the algorithms are also applied in traffic engineering to balance traffic load among multiple routes toward the same destination [15] [16]. In gateway accessing mechanisms, however, multiple routes towards the same destination node could still lead to congestion, and make the performance even worse.

Congestion aware routing protocols could effectively avoid congestion on routes toward the gateway nodes, such as the models proposed in [3] [5] [6]. In congestion aware approaches, predictable performance and load balancing can be achieved, but such approaches do not have feedback or traffic modeling mechanisms, which result in its incapabilities of handling dynamic traffic.

Cognitive methods are widely employed in exploring channel states [19] [20], but it heavily relies on the probed results across multiple channels. We believe that cognitive methods are also useful in packet routing in wireless networks. T. salonidis et al [3] propose a method to identify high throughput path in 802.11 meshed networks. With a probing mechanism to collect information on packet loss rate and channel busy time, link level throughput can be evaluated and extended to the route level.

Opportunistic access methods, although having been widely used in MAC layer, can still be applied in routing layer. Opportunistic routing protocols was first proposed by Biswas and Morris [21] to effectively exploit the network diversity among users. That paradigm needs to have many nodes to cooperate during routing process, and the scheduling efficiency largely depends on network traffic and the network topologies.

Transmission opportunities change dynamically on different routes. Selecting a number of wrong paths will largely increase the delay and seriously affect the system performance due to the feedback latency [22]. Therefore, transmission scheduling and rescheduling should quickly adapt to the route probing results. Indeed, we need to strike a balance between probing cost and its accuracy. We keep all these challenges in mind in the process of building our gateway access algorithm.

III. system model and Problem Description

In this section, we introduce the gateway access model and route probing model. In our "destination-hub" based network, gateway access problem is transformed into a MDP problem, where states of different routes are modeled, and state transitions are executed according to evaluated reward functions.

A. Gateway Access Model

Network Model: We suppose there is a gateway set $\Omega = \{Gw_1, Gw_2, ..., Gw_k\}$, where gateway nodes are connected via wired network. We further assume that, the communication cost between gateway nodes can be omitted. Nodes deployed in network form a node set V, and for a given node $v_i \in V$, there exists at least one route to gateway nodes. The route set $\Phi(v_i)$, consisting of k routes from v_i to gateway can be denoted as $\Phi(v_i) = \{\mathcal{R}_i^1, \mathcal{R}_i^2, ..., \mathcal{R}_i^k\}$. In MAC layer, we use IEEE 802.11 DCF as the MAC protocol, and the RTS/CTS interference model is applied in our network model. We also assume that, routing messages are completely known to each node, as many proposals have pointed out [23] [24].

The primary goal of this study is to explore the traffic diversity in a dynamic network scenario. The packet reassembling process at each gateway nodes is not the focus of our work and is therefore not included. Routing algorithms that consider both mobility and stability of routes towards gateways are outside the scope of this paper.

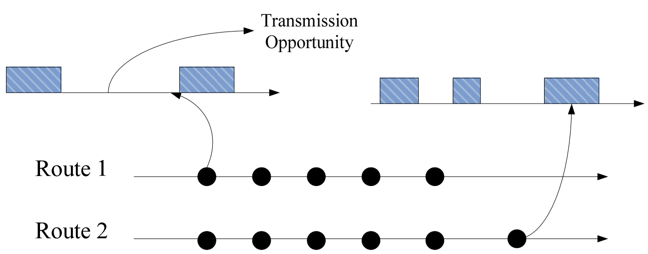

Fig. 2. Opportunistic Transmission Model on Different Routes

Route States Probing Model: As shown in Fig. 2, at time t, there are different transmission opportunities on each route. The fundamental reason for the transmission opportunities is the stochastic behavior of network traffic. In order to take advantages of these opportunities, the source node need to periodically probe multiple routes to the destination. Probing process is done by sending a short probing packet and receiving correspondent ACK packet. Round trip time (RTT) on different routes are ranked in ascending order. Suppose $k' < k$ different routes have been probed for transmission, the RTT values can be ranked as $d^{(1)} \leq d^{(2)} \leq ... \leq d^{(k')}$. We assume for each source node, there are at least one primary route to gateway nodes. The difference between primary and secondary route is that transmission on secondary route should avoid collision on primary route. We assume that, the packet delays are mainly produced by the deferring time in MAC layer, and round trip delay is the sum of deferring time on each links j in route set.

Given a node v_i in network, the j^{th} route state at time t can be denoted as $S(R_i^j(t))$. If $S(R_i^j(t)) = 1$, it means the route R_i^j is saturated and cannot be used to transmit packets any more. Otherwise, it means the route is unsaturated, and can bear more packets for transmission. Because the route probing process can only return the RTT values, we need a function to map time delay space $\vec{T}$ to network congestion state $\mathcal{S}$. That is, $f : \vec{T} \times \vec{M} \rightarrow \mathcal{S}$, where $\vec{M} \subset \vec{T} \times \vec{S}$ is the mapping matrix.

Definition: Transmission GAP of a route i at time slot t is denoted as $\varepsilon_i(t) = g_i^* - g_i(t)\{for \quad g_i^*(t) > g_i(t)\}$, where $g_i^*(t)$ is the optimized fair traffic allocation on route i, and $g_i(t)$ is the current traffic load.

Unfortunately, achieving a reasonable and accurate mapping matrix between RTT and routes is difficult. Appealing to model proposed in [6], we build our own mapping function, which considers traffic state and saturated throughput (capacity) on each routes.

B. Traffic Pattern and Schedule

Firstly, we assume that time t is discrete, where $t \in \{0, 1, 2, ...\}$. Given a node v_i in network, we assume there are totally k routes between v_i and destination hub Ω. We also assume discrete traffic types, where transmission rates on different routes toward gateway are categorized into m types, and ranked in descending order. That is, for $i < j$, we have $\lambda_i > \lambda_j$, where $1 \leq i < j \leq m$. For a traffic type i, we assume that the arrival sequence is unknown. $A_i(t) = 1$ denotes the event that traffic of type i arrives at time slot t, and we have $A_i(t) \in \{0, 1\}$, For all traffic types, we denote $\vec{A(t)}$ as the traffic pattern at time t. With the progress of t, traffic load on various routes are dynamically changing. A gateway access schedule $\pi(t)$ is to handle $\vec{A(t)}$, in which the arriving traffic loads are scheduled to different routes $R_{v_i}^j$ to fully utilize network resources. $\pi(t)$ is a mapping from $A(t)$ to $[0, 1]$, that is $\pi : A(t) \rightarrow [0, 1]$. We denote the schedule vector as $\pi(t) = \{\pi_1(t), \pi_2(t), ..., \pi_k(t)\}$, and also $\sum_{1 \leq i \leq k} \pi_i(t) = 1$.

C. Model the Gateway Access as MDP

We model our gateway access problem as a MDP. In a more generalized form, a MDP can be denoted in a four tuples $\langle S, I, T, R, \rangle$, where S is the set of states; I is set of available actions; T is the probability distribution on state transitions; and R is the reward function. For simplicity, the time step t is not presented on the four tuples. In the following sections, we use these notions $\langle S, I, T, R, \rangle$ instead of $\langle S(t), I(t), T(t), R(t), \rangle$.

State Space: The state space of MDP can be denoted as $\vec{S}$, which is a state vector of different routes. $\{s_{ij}\}_{n \times k}$ denotes the traffic load of different source, where s_{ij} denotes the traffic load on j^{th} route from source node v_i.

Action Space: The set of actions can be denoted as $I = \{0, 1\}^{(m,k)}$, where the elements of I, $a_{ij} = 1$ means traffic type i will be transmitted on the j^{th} route, and $a_{ij} = 0$ means no traffic is directed to route j.

State Transition: The first component of state **S** depends on the MAC layer protocol and traffic patterns in network, which will be discussed in section IV.

Reward function: The reward values, as we aim to improve network throughput, can be denoted as

$$R(\vec{s}, \vec{a}) = \vec{\epsilon}(t) \cdot \vec{I}(t)$$

which indicates the the capacity gap we are making use of. That is, we will be rewarded as the transmission gaps on routes are explored and used. Because it is only a one step reward, we can extend the reward function to a finite horizon. That is, given an initial state $\vec{s_0}$, the system adopt a action $\vec{a_0}$. After a horizon of H steps, the accumulated reward is given by:

$$W_H(\vec{a_0},...,a_{\vec{H}-1}) \equiv \sum_{k=0}^{h-1} R(\vec{s_k})$$

where $a_{\hat{H}}$ is the latest control action that impact network throughput, and $\hat{H} = H - d^{(c^*)}$, which is the latest control action that affect route states and $d^{(c^*)}$ is the smallest probing RTT among all observations.

IV. COG THE GAP - MODEL DRIVEN ROUTE LEVEL PROBING

In this section, we propose a cognitive route level probing method, which is driven by a validated modeling approach [6] [5]. Two issues need to be solved for building a model driven probing. The first issue is the unknown traffic pattern. This uncertainty hinders us from exploring the ensured traffic load on each route, hence the transmission gap. The second issue is the inaccurate and incomplete probed RTT values. Inaccurate RTT results is due to the fact that different routes have difference in RTT variance. Incomplete RTT results exist because it is wise not to frequently probe the networks to avoid congestion.

We use the hidden Markov model (HMM) for traffic pattern modeling. When the traffic patterns and distributions are known, a fair and optimized traffic allocation method is proposed. The RTT values under the allocated traffic pattern are evaluated for cognition on the transmission opportunity, that is, the capacity gap. Expectation maximization (EM) method is used in dealing with the incomplete data, and a Markowitz model is applied to make choices on different RTT values. According to the returned and refined RTT values, we evaluate the transmission gap, which is useful for the next step MDP modeling and computation. We also discuss the mapping function between RTT and route capacity on a validated IEEE 802.11 DCF model [5] [6].

A. Modeling Traffic Arrival Distributions

For each node in a network, the data transmission rate is observable, while the model is not known. We use HMM to model distributions of traffic arrival patterns. For applications that traffic arrival rates are not known, the EM can be used to infer a HMM heuristically [30] [31]. In order to avoid abuse of notations, a HMM model is denoted as a tuple $< Q^H, T^H, \Lambda^H, \Pi^H >$, where Q^H is a finite set of states, and T^H is the probability distribution for next state in Q, Λ is a mapping from Q to the probability of a task arrival, Π^H is a distribution of the uncertain initial state over Q^H. A hidden state sequence $< q_1, q_2, ..., q_k >$ is generated by the distribution function Π^H.

We assume that at each observation step, X_t is generated by a $K - dimensional$ real value hidden state H_t, and the distribution is generated by:

$$P(H_{1:T}, X_{1:T}) = P(H_1)P(X_1|H_1)\prod_{t=2}^{T} P(H_t|H_{t-1})P(X_t|H_t)$$

Due to the incomplete data of our probing results, hidden variables exist in our model, the likelihood log function can be formed as:

$$\mathcal{L} = \log P[Y|\theta] = \log \sum_X P[Y, X|\theta]$$

To derive the EM algorithm for inferring HMM parameters, we need to compute the log probability of hidden variables and observations:

$$\log P(H_{1:T}, X_{1:T}) = \log P(H_1) + \sum_{t=1}^{T} \log P(X_1|H_1) + \sum_{t=2}^{T} \log P(H_t|H_{t-1})$$

As the traffic patterns are $k - dimensional$ values, the transition probability can be computed as:

$$P(H_t|H_{t-1}) = \prod_{i=1}^{m}\prod_{i=1}^{m} \Psi_{ij}^{H_{t,i},H_{t-1,j}}$$

where Ψ_{ij} is the probability of transition from state i to j,

$$\log P(H_t|H_{t-1}) = \sum_{i=1}^{k}\sum_{j=1}^{k} \log H_{t,i}H_{t-1,j}\Psi_{ij}$$

Given the transition probability distributions, we can compute $\Lambda = \{\gamma_1(t), \gamma_2(t), ..., \gamma_k(t)\}$, which is a belief vector, where $\gamma_i(t)$ is the conditional probability given decision and observation history.

B. Traffic Allocation

We suppose that, for each source node, there is a primary route to gateway node. This assumption is easily satisfied because the many routing protocols for wireless networks can provide an optimal route to destination.

The throughput of each link can be modeled according to the conversion function, as proposed in [5]:

$$\vec{\mathbf{g}} = \mathbf{R} \cdot \vec{\mathbf{f}}$$

where $\vec{\mathbf{g}}$ is the vector of end to end throughput, $\vec{\mathbf{f}}$ is the vector of traffic demand, and $\mathbf{R} = \{R_{id}\}_{n \times m}$ is the routing matrix.

The problem of maximizing total end-to-end throughput can be formulated into a stochastic non-linear optimization problem (SNLP). Different with the model proposed in [5], the traffic patterns are distributed over serval candidate forms. Distributions and transitory probabilities have been computed by HMM and EM method, which have been presented in section IV-A.

$$\max \quad \sum_d f_d$$

$$subject \quad to \begin{cases} R_i\vec{f} \le F_i(\tau), & \forall i \in L; \\ G_i(\tau) \le 0, & \forall i \in L; \\ 0 \le f_d \le f_d^*, & \forall d; \\ 0 \le f_d^j \le f_d^{j*}, & \forall j; \\ P_{ij}^d \to \pi_i, & \forall i, j; \\ 0 \le \tau_i \le 1, & \forall i \in L;. \end{cases}$$

where τ is the traffic rate, $G_i(\tau) < 0$ is a necessary condition for transmission on each link, as indicated in [5].

Algorithm Heuristic Two-stage Programming Algorithm

Compute First-stage Programming results, and the allocation factor $\vec{x}$
for $i := 1$ to s do **do**
  Compute Second-stage Programming results, and $\chi^s = d^s \upsilon^s$
  $\tilde{\chi} = \max \chi, \sum_{s=1}^{r} p_s \chi^s$
end for
if $\tilde{\chi} = \chi$ **then**
  The first-stage result
else
  The second-stage result
end if

Fig. 3. Two-Stage Matching Algorithm Description

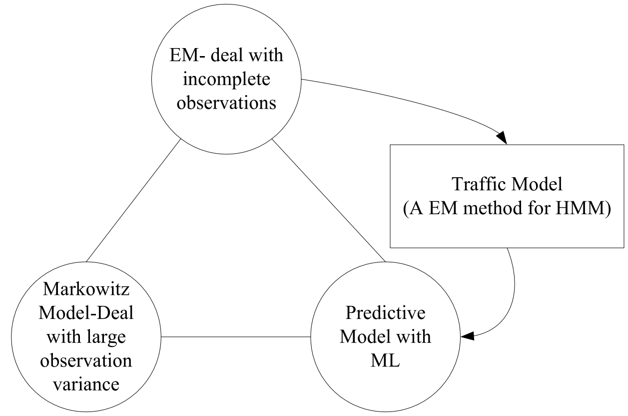

Fig. 4. Flow Chart of the ”Cog Gap”

$F_i(\tau)$ is the maximal throughput bearable on link $i \in L$, and L is the link set, given by [5].

We use ”two-stage” programming [34], a heuristic algorithm for solving stochastic programming models. The goal of our optimization model is modified as:

$$\max[\sum_d x_d + \sum_s \sum_d p_s x_d(s)]$$

Where p_s denotes the probability of each traffic pattern s, and $x_d(s)$ denotes the allocated traffic for transmission at state s.

Let $\vec{x}$ be a first-stage solution, and let $\vec{\chi} = \vec{c}\vec{x}$. For scenario s, let υ^s be a second-stage myopic solution, and let $\chi^s = d^s \upsilon^s$. Let $\tilde{\chi} = \max\{\chi, \sum_{s=1}^{r} p_s \chi^s\}$. If $\tilde{\chi} = \chi$, then return and $(\vec{x}, \mathbf{0}, \mathbf{0}, ..., \mathbf{0})$; otherwise, return $(\mathbf{0}, \upsilon^1, ..., \upsilon^r)$ and $\sum_{s=1}^{r} p_s \chi^s$. Pseudo codes of two-stage programming are listed in Fig. 3. For each stage, we use the iterative linear programming method proposed in [5] to solve this problem.

C. Fully Utilizing RTT Values

In order to ”cog the gap”, there are three challenges on route level opportunity probing. First, the observed values through route probing are not sufficient. Frequent probing would lead to large protocol overhead. On the other hand, long observation time would accordingly loose opportunities for transmission. Second, RTT values would possibly have large variance incurred by traffic variations. Routes with smaller RTT variance, should be preferred over the larger ones. Third, RTT values are the sum of deferring time on each link, so there is still a gap between the evaluated capacity and the real value.

Fig. 4 illustrates the relationship between modules for throughput probing.

Expectation Maximization: To address the issue of incomplete data, we use EM method, which works as follows. Let $\mathbf{y}$ denote incomplete data consisting of values of observable variables, and $\mathbf{z}$ denote the missing observations. The likelihood formula would be much more convenient if mixture components that “generated” the samples were known (see example below). The conditional distribution of the missing data $\mathbf{z}$ given the observed can be expressed as:

$$p(\mathbf{z}|\mathbf{y}, \theta) = \frac{p(\mathbf{y},\mathbf{z}|\theta)}{p(\mathbf{y}|\theta)} = \frac{p(\mathbf{y}|\mathbf{z}, \theta)p(\mathbf{z}|\theta)}{\int p(\mathbf{y}|\hat{\mathbf{z}}, \theta)p(\hat{\mathbf{z}}|\theta)d\hat{\mathbf{z}}}$$

An EM algorithm iteratively improves an initial estimate θ_0 by constructing new estimates θ_1, θ_2 etc, which can be computed according to the following equation:

$$\theta_{n+1} = \arg\max_{\theta} \mathcal{G}(\theta)$$

where $\mathcal{G}(\theta)$ is the expected value of the log-likelihood. $\mathcal{G}$ is given by

$$\mathcal{G}(\theta) = \sum_z p(z|y, \theta_n) \log p(y, z|\theta)$$

Markowitz Method: To offset the negative effects of large variance RTT values, we use the notion “efficient frontier” proposed by Markowitz method. The “efficient frontier” is used to maximize the return for a given risk. It is given by an optimal model:

$$\max E[R] = \sum_{i=1}^{k} R_i X_i$$

$$\sum_{i=1}^{k} \sum_{j=1}^{k} \sigma_{ij} X_i X_j = V$$

$$\sum_{i=1}^{k} X_i = 1$$

The R_i is the returned rewards of RTT, as we prefer the route with lower RTT values, the reward value can be denoted as follows:

$$R_i = 1 - \frac{RTT_i}{\sum_{1 \le j \le k} RTT_j}$$

$\sigma_{ij} = cov(X_i, X_j)$ and V is the bounding on sum of covariance of selected values.

Maximum Likelihood: We need an accurate mapping between RTT values and traffic patterns. Since we have the distributions of traffic patterns, and the RTT values are known, we can maximize the probability of traffic pattern sequence from the observing results. Thus, a maximum likelihood mapping from RTT values to traffic patterns can be given by:

$$\gamma_t(i) = P[(q_t = S_i)|O, \lambda]$$

that is the probability of being in state S_i at time t, given the observation O and model λ. If the traffic patterns and the traffic scheduling are given, the RTT values can be evaluated as indicated in [5], given by:

$$RTT_i = \sum_{j \in R_i} W_j = \sum_{j \in R_i} \frac{g(j)}{\tau(j)}$$

Where $j \in R_i$ denotes each link j on route R_i, and $\frac{g(j)}{\tau(j)}$ is ratio between throughput of link j and allocated traffic of link j. We'd to maximize the expected number of correct states. We use the Viterbi algorithm [31], an algorithm modifying the optimization goal, to find the state sequence

$$\delta_{t+1}(j) = [\max_j \delta_t(i) a_{ij} \cdot b_j(O_{t+1})]$$

where $\delta_t(i) = \max P[q_1 q_2 ... q_t = i, O_1 O_2 ... O_t | \lambda]$. For the given observation sequences of the RTT values, we can make an inference on the route states given by $\delta_t(i)$.

V. A Decision Based Gateway Access Approach

In this section, we introduce our solution to the MDP model for dynamic spectrum access. We use a technique called "Hindsight Optimization" [26], which is used to heuristically solved the the problem of calculating "Q-value" [27].

According to the MDP model described above, we can formulate our spectrum access problem as follows: Goal of optimization is to achieve a policy maximizing the objective function

$$V_H^*(s_0) = max[R(x_0, a_0), , R(\vec{s}_{H-1}, \vec{a}_{H-1}]$$

As the time progresses, actions are triggering with the dynamic network states. The policy can be denoted as $\pi = \{\mu_0, \mu_1, \mu_2, ...\}$, where $\mu_k : \vec{s} \to \vec{a}$. The policy π is markov if it is only related to the current state and transition probabilities.

A. Analysis on Structure of Optimal Results

Optimal results of MDP models, although are usually not achievable due to the explosion of computation on large state space, is still meaningful to solve our problem heuristically. Fix a large H, we focus on finite-horizon reward and follow a standard approach in solving MDP [27] [28]. We then start the analysis from the fixed horizon model. For a given initial state $\vec{s}$, let

$$V_H^*(\vec{s}) = \max_\pi V_H^\pi(\vec{s})$$

That is the "Q-value", where

$$Q_k(\vec{s}, \vec{a}) = R(\vec{s}) + E(V_{k-1}^*(\vec{s})'), \qquad k = 1, 2, 3, ..., H$$

is the utility function of action $\vec{a}$ at state $\vec{s}$, where k is the number of time step. As mentioned in Bellman's equations [29]

$$V_H^*(\vec{s}) = \max_{\vec{a} \in A} \quad Q_H(\vec{s}, \vec{a})$$

and a policy $\pi^* = \{\mu_0^*, \mu_1^*, ...\}$ is optimal if it satisfies the following equation for all k.

$$\mu_k^*(\vec{s}) = \arg\max_{\vec{a} \in \mathbf{A}} Q_H(\vec{s}, \vec{a})$$

In practice, computing Q_H is polynomial to the size of state space. There are two problems need to be solved. The first problem is that evaluating effects of the proposed action is difficult and costly. We need an appropriate model to validate our actions to avoid channel congestion. The second problem is that the computation for very large state space is extremely hard. We need an efficient algorithm to deal with the exponential increase of the decision tree [32].

B. Hindsight Optimization Technique

The hindsight optimization technique [27] [26] is based on traffic models, which can be viewed as stochastic predictions for future network behaviors. The hindsight optimization algorithm heuristically evaluates $\hat{Q}(\vec{s}, \vec{a})$. Considering our objective:

$$\vec{a}^* = \arg\max_{\vec{a} \in \mathcal{A}} Q(\vec{s}, \vec{a})$$

Computation of $Q(\vec{s}, \vec{a})$ is carried out by estimating $\hat{Q}(\vec{s}, \vec{a})$. In short, $\hat{Q}(\vec{s}, \vec{a})$ is a sampled evaluation of $Q(\vec{s}, \vec{a})$ given by

$$\hat{Q}_n(\vec{s}, \vec{a}) = \frac{1}{n} \sum_{t=1}^{t=n} W_t^*(\vec{s}, \vec{a})$$

$$\hat{Q}(\vec{s}, \vec{a}) = R(\vec{s}) + E_{S_1, .., S_H} \max_{\vec{a}_1, ..., \vec{a}_H} W_{H-1}((\vec{a})_1, ..., (\vec{a})_{H-1})$$

where

$$W_{H-1}(\vec{a}_1, \vec{a}_2, ..., \vec{a}_{H-1}) = \sum_{k=1}^{H-1} R(\vec{s}_k, \vec{a}_k)$$

and $W_t^*(\vec{s}, \vec{a})$ is the hindsight optimal value for trace t. As mentioned in in [26], the $\hat{Q}(\vec{s}, \vec{a})$ is an upper bound on $Q(\vec{s}, \vec{a})$.

Given the estimate $\hat{Q}(\vec{s}, \vec{a})$ of $Q(\vec{s}, \vec{a})$, the hindsight optimization makes evaluation on different actions. As what we are concerning is to rank a series of actions, the upper bound estimation can be arbitrarily loose without effecting the results [27].

In our problem, the action space is somehow continuous. This uncountable state space makes direct evaluation impossible and we cannot make a good estimation on $\hat{Q}(\vec{s}, \vec{a})$. We applies the method proposed in [26] to handle continuous actions.

Algorithm Gradient Search Algorithm

```
Initialize a(0)
for k = 1, 2, ..., do
    ∇_a Q̂_n((s), a) = (1/n) Σ_{t=1}^{n} ∇_a W_t^*(s, a)

    if ||∇_a Q̂_n((s), a)|| ≤ ε then
        break
    end if
end for
```

Fig. 5. Pseudo Code of Gradient Search Algorithm

At each decision step, as the action vector is continuous, we can use the searching algorithm to maximize $\hat{Q}(\vec{s}, \vec{a})$. The pseudo code on our proposed search algorithm are shown in Fig. 5.

VI. Discussions

We will discuss two important issues in this section, probing cost and probing errors.

A. Gateway Access and Probing Cost

The aforementioned decision based gateway access mechanism does not consider the cost on channel probing. We are attempting to prove the existence of optimal stopping time on route states' probing from the perspective of cooperative game. Appealing to work of Zheng et al [25], we define the return rate of gateway access probing, where $\{N_1, N_2, ..., N_L\}$ denotes the stopping times for route probing, which means the probing process is stopped after having probed L channels.

$$r_L = \frac{\sum_{l=1}^{L} R_{N_l} T}{\sum_{l=1}^{L} T_{N_l}} \rightarrow lim_{L\rightarrow\infty} \frac{E[R_{N_l} T]}{T_{N_l}}$$

R_N is the stopping reward variable, T_N is the duration of the N^{th} probing, and N is the stopping time.

As indicated in [33], if the optimal stopping point exists, the following equation should be satisfied:

$$E[\sup_n Z_n] < \infty \quad and \quad \limsup_n Z_n = -\infty$$

where $Z_n \triangleq R_{(n)}T - xT_n$, $T_n \triangleq \sum_{j=1}^{n} K_j \varrho + T$, where K_j denotes the number of probing times, ϱ is the average duration for each probing, and n is the number of routes being probed.

With the increasing of n, the reward value is bounded by the maximal reward, and naturally $\limsup_n Z_n = -\infty$. We are to prove that, $E[\sup_n Z_n] < \infty$. Let $\tilde{\mathcal{K}}$ denote the average probing, and we have

$$E[\sup_n Z_n] = E[\sup_n R_{(n)}T - x\sum_{j=1}^{n} k_j \varrho - xT]$$

$$\leq E[\sup_n n(\frac{R_{(n)}T}{n} - x\tilde{\mathcal{K}}\varrho) - xT] \leq \infty$$

The existence of optimal stopping point will lead the gateway accessing to a new problem. If the probing cost is not negligible and the route states distribution is not fully known, the optimal stopping point would lead to partially observable route states for decision based model.

B. Considering Probing Errors

Probing results are analyzed according to RTT values, which is a measurement on network states. According to the results in [10], RTT values can be given by:

$$D_n = \Delta_n + U_n$$

where U_n is a Gaussian random variable with mean zero and variance σ. We use Neyman-Pearson detector for traffic state probing, which is given by:

$$\|\mathbf{Y}\|^2 = \gtrless_{H_1}^{H_0} \xi$$

where σ_0 and σ_1 are variance values for the route with and without capacity gap respectively. ξ is the error tolerant factor, which means that if the error rate is lower than ξ, the effects of performance reduction can be overlooked.

$$\begin{cases} \mathcal{H}_0(g(t) > 0), & \mathcal{Y}_i \sim \mathcal{N}(0, \sigma_0^2), i = 1, 2, ..., L \\ \mathcal{H}_1(g(t) < 0), & \mathcal{Y}_i \sim \mathcal{N}(0, \sigma_1^2), i = 1, 2, ..., L \end{cases}$$

$$\epsilon \triangleq Pr\{\|\mathbf{Y}\|^2 > \xi | H_0\} = 1 - \Gamma(\frac{L}{2}, \frac{\xi}{2\sigma_0^2})$$

where $\Gamma(L, x) = \int_0^x t^{L-1} e^{-t} dt$.

$$1 - \delta \triangleq Pr\{\|\mathbf{Y}\|^2 > \xi | H_1\} = 1 - \Gamma(\frac{L}{2}, \eta\frac{\sigma_0^2}{\sigma_1^2})$$

The errors in route state determination however, are dependent on RTT value distributions, state determination methods, and threshold etc.

VII. Evaluations

We implement the Cog Gap algorithm into GlomoSim [35], and conduct extensive simulation experiments to evaluate our design. In a gateway access system, there are many factors which would impact our algorithm, such as gateway placement, number of gateways, traffic patterns etc. We examine the impact of probing process on cognitive gateway accessing.

TABLE I
Simulation Parameters

Parameters	Value
Number of Wireless Nodes	50
Number of Gateway Nodes	5, 10, 15, 20, 25
MAC Protocol	IEEE 802.11 DCF Module in GlomoSim
Routing Protocol	FSR
Working Area	(1000, 1000) meters square
Communication Range	100 meters
Packet Length	512 bit
Channel Bandwidth	1M

A. Impact of Gateway Placement

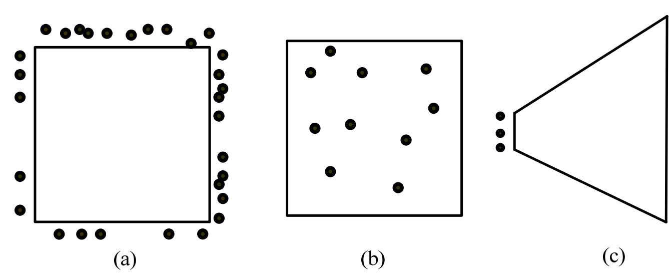

Fig. 6. Impact on gateway placement

Firstly, we first study the impact of gateway placement on network performance. As shown in Fig. 6(a), gateway nodes are deployed on a surrounded border. In Fig. 6(b), gateway nodes can be deployed in any points of closed area. Fig. 6(a)

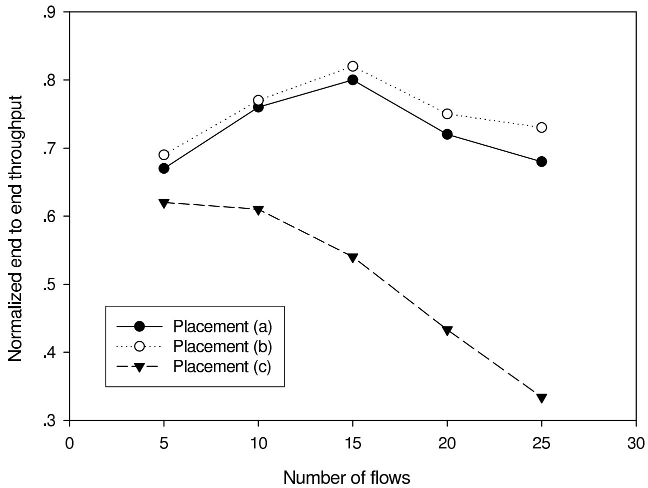

Fig. 7. Impact on number of flows

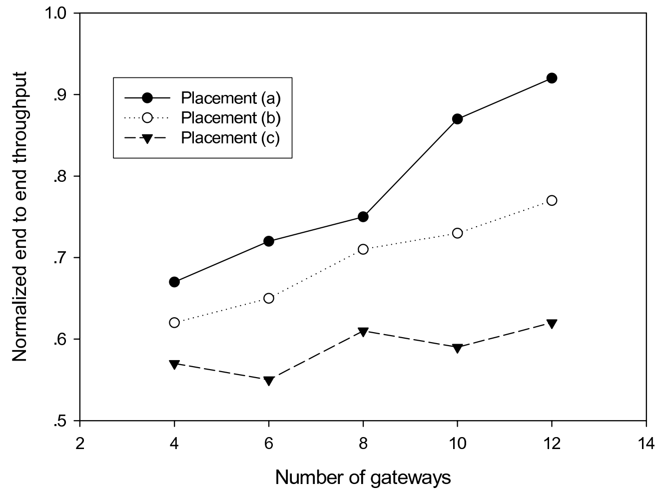

Fig. 8. Impact on number of gateways

and (b) are conventional in wireless networks, such as wireless sensor networks. Fig. 6(c) depicts that there are only a fraction of border can be used to deploy gateway nodes. The placement in Fig. 6(c) stands for the network with constrained area for gateway nodes.

With increased traffic loads, the normalized throughput varies on different gateway placement as shown in Fig.F̃IG:TRAFFIC-NUMBER. Placement (a) outperforms (b) and (c) with the increasing number of flows. Because more transmission opportunities exist in the placement (a) when a destination-hub based gateway access model is applied.

B. Impact of Gateway Number

Apart from the gateway placement, the number of gateway nodes is also an important factor. In our simulations, we vary the number of gateways to explore the impact of gateway number on our proposed algorithm.

As shown in Fig. 8, by increasing the number of gateways, throughput on each gateway placement mode increases accordingly. The placement (c), however, does not demonstrate a throughput improvement with the increasing number of gateway nodes, because the gateway nodes are clattered into a small region, which will lead to congestions in the end.

C. Impact of Traffic Patterns

We model different level of traffic loads and variations to illustrate the impact of traffic patterns. We use three types of traffic patterns, as shown in Table II. The second column lists the traffic patterns of each mode. The vector values denote the packet interval for each traffic pattern $\{\mathcal{TP}_1, \mathcal{TP}_2, \mathcal{TP}_3\}$ respectively, and the third column is the transition probability between each transmission rate. The vector values listed as $\{\mathcal{P}_{12}, \mathcal{P}_{23}, \mathcal{P}_{31}\}$, where $\mathcal{P}_{ij}$ denotes the probability transition from pattern i to j.

The three traffic modes are different on transmission opportunities. Mode 1 is traffic load heavy, and Mode 2, 3 will provide more opportunities for transmission on gaps comparing to the Mode 1. The difference between mode 2 and mode 3 is the traffic variations. In mode 3, there are more opportunities for network with light traffic load, and the probabilities on transition are different from Mode 2.

TABLE II
SIMULATION PARAMETERS

Traffic Mode	**Traffic Load**	**Transition Probability**
Mode 1	{0.5s, 1s, 2s }	{0.6, 0.2, 0.2}
Mode 2	{1s, 4s, 6s}	{0.1, 0.1, 0.8}
Mode 3	{2s, 4s, 6s}	{0.3, 0.5, 0.2}

As shown in Fig. 9, we can see that traffic mode 1, which is relative high in terms of traffic load, has less transmission opportunities than traffic mode 2 and 3. The network throughput of each placement is very close. While in mode 2 and 3, the placement (a) outperforms (b) and (c) in mode 2 and 3, with varying traffic load. Because there are more opportunities in the two modes, and using the ”Destination Hub”, more opportunities could be exploited for placement (a).

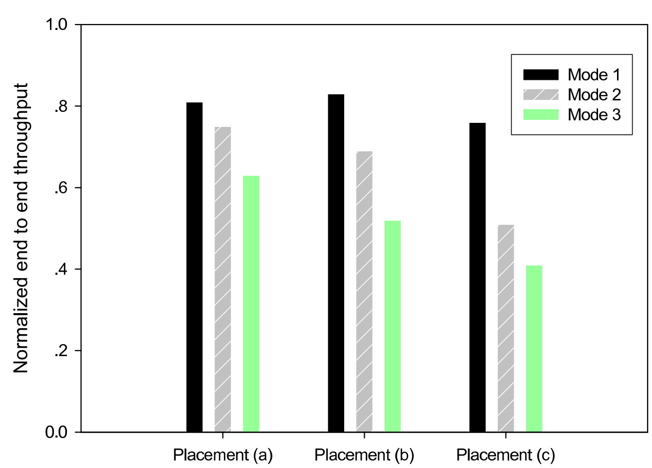

Fig. 9. Impact on traffic patterns

VIII. Conclusions and Future Work

The proposed destination-hub based gateway accessing algorithm paves the way for opportunistic route level traffic allocations. Transmission opportunities are exploited using a route level cognitive paradigm. By applying the IEEE 802.11 DCF model, the transmission opportunities on different routes are explored and evaluated. The traffic allocation is based on the decision dominated in the MDP model. In future work, the following issues need to be further studied:

1) The probing cost is not in full consideration. In section VI, we have proven the optimal stopping point. As only the partially probed route states are available, gateway accessing turns into a POMDP problem, which needs to be further studied.
2) The effects on errors, which is evaluated only according to RTT variance, are not fully addressed. Because the errors might exist in each modeling process, such as traffic modeling, route state determination, etc, we will study the problem to improve the accuracy.
3) The algorithm mainly concerns on theoretical results on opportunities exploitation and decision making. In future works, we need a systematic design on "Destination Hub" based gateway access protocol, with both "Cog Gap", and coordination process between gateway nodes.

References

[1] Joshua Robinson, Mustafa Uysal, Ram Swaminathan, and Edward Knightly. Adding Capacity Points to a Wireless Mesh Network Using Local Search. *In Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008.

[2] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement driven deployment of a two-tier urban mesh access network. *In Proceedings of ACM MobiSys*, Uppsala, Sweden, June 2006.

[3] T. Salonidis, M. Garetto, A. Saha, E. Knightly. Identifying High Throughput Paths in 802.11 Mesh Networks: a Model-based Approach. *In Proceedings of IEEE ICNP*, Beijing, China, Oct 2007.

[4] R. Chandra, Lili Qiu, Kamal Jain, and Mohammad Mahdian. Optimizing the placement of integration points in multi-hop wireless networks. *In Proceedings of ICNP*, Berlin, Germany, October 2004.

[5] Y. Li, L. Qiu, Y. Zhang, R. Mahajan. Predictable performance optimization for wireless networks. *In Proceedings of the ACM SIGCOMM*, Seattle, U.S, 2008.

[6] Lili Qiu, Yin Zhang, Feng Wang, Mi Kyung Han, Ratul Mahajan. A General Model of Wireless Interference. *In Proceedings of the ACM Mobicom*, Montreal. CA. 2007.

[7] W. Wang, Y. Wang, X-Y. Li, W-Z. Song, O. Frider. Efficient Interference-Aware TDMA Link Scheduling for Static Wireless Networks. *In Proceedings of the ACM Mobicom*, Los Angeles, CA, USA, September 23-29, 2006.

[8] Fan Li, Yu Wang, Xiang-Yang Li and Ashraf Nusairat. Gateway placement for throughput optimization in wireless mesh networks. ACM MONET. Special Issue on Advances in Wireless Mesh Networks. December, 2007.

[9] G.S Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, F. Cuomo. Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks. *In Proceedings of the ACM Sensys*,Boulder, Colorado, USA. October 2006.

[10] M. A. Kaaffar, L. Mathy, C. Barakat, etc. Securing Internet Coordinate Embedding Systems. *In Proceedings of the ACM SIGCOMM*, Kyoto, Japan, August 2007.

[11] M. Kodialam, T. Nandagopal. Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks. *In Proceedings of ACM/Mobicom*, Cologne, Germany. Aug, 2005.

[12] M. Andrews, L. Zhang. Scheduling Algorithms for Multi-Carrier Wireless Data Systems, *In Proceedings of the ACM Mobicom*, Montreal. CA. 2007.

[13] M. Alicherry, R. Bhatia, L. E. Li, Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks. *In Proceedings of ACM/Mobicom*, Cologne, Germany. Aug, 2005.

[14] K. N. Ramachandra, E. M. Belding, K. C. Almeroth, M. M. Buddhikot. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. *In Proceedings of IEEE INFOCOM*, Barcelona, Spain, April, 2006.

[15] S. Baek and G. de Veciana. Spatial Energy Balancing Largescale Wireless Multihop Networks. *In Proceedings of IEEE INFOCOM*. Miami, FL, USA, March 2005.

[16] L. Popa, A. Rostamizadeh, R. M. Karp, C. Papadimitriou, Ion Stoica. Balancing Traffic Load in Wireless Networks with Curveball Routing. *In Proc. Of ACM Mobihoc 2007*. Montreal, Quebec, Canada, September, 2007.

[17] P. Kyasanur, N. H. Vaidya. Capacity of multi-channel wireless networks: Impact of number of channels and interfaces. *In Proceedings of ACM/Mobicom*, Cologne, Germany. Aug, 2005.

[18] Y. Xi, E. M. Yeh. Distributed Algorithms for Spectrum Allocation, Power Control, Routing, and Congestion Control in Wireless Networks. *In Proceedings of the ACM Mobicom*, Montreal. CA. September, 2007.

[19] N. B. Chang, M. Liu. Optimal Channel Probing and Transmission Scheduling for Opportunistic Spectrum Access. *In Proceedings of the ACM Mobicom*, Montreal. CA. September, 2007.

[20] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework, *IEEE Journal on Selected Areas in Communications*, VOL. 25, NO. 3, April 2007.

[21] S. Biswas and R. Morris. ExOR: Opportunistic Multi-Hop Routing for Wireless Networks. *In proceedings of ACM SIGCOMM*, Philadelphia, PA, August, 2005.

[22] H. K. Le, D. Henriksson, and T. F. Abdelzaher. A Control Theory Approach to Throughput Optimization in Multi-Channel Collection Sensor Networks. *In Proceedings of the ACM/IPSN*, Cambridge, Massachusetts, April, 2007.

[23] R. Draves, J. Padhye, , and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. *In Proceedings of ACM MobiCom*, Philadelphia, PA, USA. Sept, 2004.

[24] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. *In Proceedings of ACM MobiCom*, Philadelphia, PA, USA, Sept. 2004.

[25] D. Zheng, W. Ge, J. Zhang. Distributed Opportunistic Scheduling For Ad-Hoc Communications: An Optimal Stopping Approach. *In Proc. Of ACM Mobihoc 2007*. Montreal, Quebec, Canada, September, 2007.

[26] G. Wu, E. P. Chong, R. Givan. Congestion Control via Online Sampling. *In the Proceedings of IEEE INFOCOM*, Anchorage, Alaska, April, 2001.

[27] E. K. P. Chong, R. L. Givan, and H. S. Chang, A framework for simulationbased network control via Hindsight Optimization. *In the 39th IEEE Conf. on Decision and Control*, Dec. 2000.

[28] E. A. Feinberg, A. Shwarz. Handbook of Markov Decision Process, Methods and Applications. *Kluwer's International Series*. 2002.

[29] D. P. Bertsekas, Dynamic Programming and Optimal Control. Volumes 1 and 2, *Athena Scientific*, 1995.

[30] Z. Ghahramani. An Introduction to Hidden Markov Models and Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*. 15(1):9-42, 2001.

[31] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*. Vol 77, No. 2, Feburary, 1989.

[32] B. Givan, H. Chang, etc. http://www.cs.rice.edu/ vardi/dag01/givan3.ppt.

[33] T. Ferguson. Optimal Stopping and Applications. http://www.math.ucla.edu/ tom/Stopping/Contents.html, 2006.

[34] Peter Kall, Janos Mayer. Stochastic Linear Programming. Models, theory, and Computation. Springer. 2005.

[35] X. Zeng, R. Bagrodia, and M. Gerla, GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In the 12th Workshop on Parallel and Distributed Simulations, May 1998.