

ANALYSIS OF ADMISSION CONTROL IN P2P-BASED MEDIA DELIVERY NETWORK BASED ON POMDP

BAOQUN YIN, SHAN LU AND DONG GUO

Key Lab of Anhui Network Communication System and Control
University of Science and Technology of China
P. O. Box 4, Hefei 230027, P. R. China
bqyin@ustc.edu.cn; {lsdaisy; guodong7}@mail.ustc.edu.cn

Received May 2010; revised September 2010

ABSTRACT. *In this paper, we study admission control problems of P2P-based Media Delivery Network (P2P-based MDN). We propose a specific partially observable Markov Decision Process (POMDP) model for P2P-based MDN. Based on this model, the observation-based randomized policy for admission control is provided, which optimizes system's performance by applying policy-gradient algorithm. Observation-based policy can enhance the real-time performance of the system, and policy optimization can improve the accuracy of controller's judgments based on partial information. Novel source selection policy and bandwidth allocation policy are designed to reduce service delay and provide high-quality service. A numerical example is provided to illustrate the effectiveness of our methods.*

Keywords: P2P-based MDN, POMDP, Observation-based randomized policy, Admission control

1. **Introduction.** Streaming media services pose stricter demand on the Quality of Service (QoS) comparing with other types of Internet services. Before the birth of P2P-based architecture, much work has been done to provide QoS assurance of media delivery systems. Various techniques, such as admission control, congestion control and load balance, have been applied to improve QoS. Admission control is an important issue worthy of discussing. A good admission control policy can reduce the degree of the system's congestion and lighten the workload of the central server. The development of admission control just follows the evolution of network architectures.

Admission control policies on the traditional service framework with one single server can be simply divided into two types: deterministic algorithm and statistical algorithm. Deterministic algorithm guarantees the resource consumed by all the connections in the system less than or equal to the capacity of the system [1,2]; statistical algorithm uses the distributions of the system's parameters to compute the overflow probability of the system [3-6]. Admission control in distributed systems can be divided into two kinds: centralized control [7] and distributed control [8,9]. In this paper, considering the special architecture of P2P-based MDN, we adopt both statistical algorithm and distributed control among Peer Nodes (PNs).

Another function of proxy servers in distributed system is to store the heads of media contents to reduce the service delay by transmitting data to clients in advance. This technique is called prefix caching, which has been widely used in streaming media distribution services [10-12]. Prefix caching has a potential problem as mentioned in [8]. If a proxy server accepts a client's request and transmits data, the effort may be worthless when the media server is heavily loaded and has to finally deny the request. However, this problem is not addressed in [8].

P2P-based MDN, which hands out media contents to the edge of network through P2P way, was created in recent years. Kang and Yin discuss the deployment of clients with different service capabilities in hybrid CDN-P2P system in [13]. Zheng et al. study the hit ratios and throughputs of the system in [14]. To the best of our knowledge, there is no work concerning admission strategy based on this kind of architecture yet. For an unstructured P2P network, every PN can be treated as a server. Each PN decides whether to transmit data to another PN according to the “reputation” of that PN and the “recommendation” from other PNs. Reputation mechanism and recommendation mechanism are hot research fields for unstructured P2P systems [15-17]. However, admission control of P2P-based MDN is nothing about the reputation mechanism, because each PN in the system is set by the service providers and there is just mutual profitable relationship among the PNs. In addition, streaming media servers in P2P-Based MDN also have the prefix caching function as the proxy servers in distribution systems do. In this paper, we will study admission control of P2P-based MDN and try to solve the problem of resource waste in prefix caching. The major distinctions of our work from previous work are as follows.

Considering the independency and heterogeneity of PNs, it is not reasonable for each PN to know the specific states of all the other PNs when conducting independent admission control tasks. Due to this uncertainty of the system’s state, we model the process of P2P-based MDN admission control based on POMDP. In order to construct the model, we creatively define a new form of the system’s state to describe the service interactions among PNs in the system. An observation-based randomized policy is proposed for the admission control. Observation is the partial information of the system. Each node makes admission decisions according to the part of information it can get. Randomized policy takes into account the uncertainty of the system’s state and source peer selection, and admits a new request without determinacy.

There are three main contributions in this paper. Firstly, a model based on POMDP is developed to model admission control process of PNs in P2P-based MDN, and a novel method is proposed to define the state for P2P-based architecture especially in the process of modeling. Secondly, it emphasizes the importance of raising providers’ benefits for the long-term development of the system. It is proved that the optimal algorithm can provide both high-level quality media services and economic benefit. Thirdly, a new source selection policy and bandwidth allocation strategies for storage PNs are proposed to help reduce the service delay and stabilize service quality. All these points above can be applied in performance research on other P2P-based networks.

The rest of the paper is organized as follows: the detailed architecture of P2P-based MDN is presented in Section 2; a specific POMDP model for P2P-based MDN admission control is constructed in Section 3; in Section 4, admission steps, admission policy and algorithm to optimize the policy are proposed on the basis of the model we constructed; in Section 5, we discuss other strategies existing in P2P-based architecture, especially, source selection strategy and bandwidth allocation strategy for PNs; in Section 6, we give a numerical example; we draw conclusions and propose further work in Section 7.

2. P2P-Based MDN. P2P-based MDN is designed to provide high-quality streaming media service. It adopts P2P architecture in content delivery network (CDN). Media service providers push contents and services to the edge of network and connect all the caching servers in P2P way. P2P architecture reduces the workload of central server and improves the effectiveness in content sharing. We will first introduce the architecture of P2P-based MDN.

P2P-based MDN is composed of a Virtual Content Server (VCS), a Manager Module (MM) and PNs located at the network edge, with a Media Content Provider (MCP) and

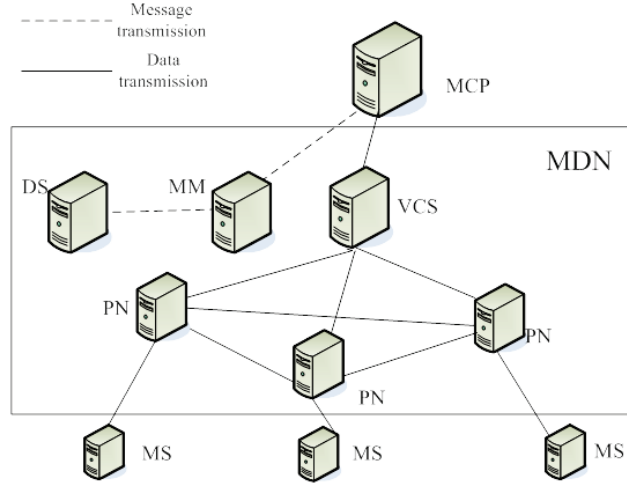


FIGURE 1. Architecture of P2P-based MDN

Media Servers (MSs) as parts of the network (as shown in Figure 1). The function of each part is as follows.

MCP releases media contents (e.g., movies) to MDN through VCS. VCS actually stores no content, and however, it transmits the contents that MCP has released to PNs. MM receives MCP's demands of releasing new media contents and deploys new movie resources among PNs according to certain deployment strategy. DS (Directory Server) returns source node lists to PNs when PNs send source query to DS. PNs are located at the edge of network. They store the tails of the movies released by MCP and provide media contents to MSs. MSs are also located at the edge of network. They request media contents from MDN as clients and provide streaming service to set-top boxes or PC users. The heads of the movies are stored in MSs in order to swiftly response to users.

PNs share their own resources with each others, and all the PNs integrate into a large storage space. For the relationship between the storage capability of PNs and the total size of movies released by MCP, Zheng et al. in [14] consider three cases. In this paper, we only discuss case 2, in which the total storage capability of PNs is greater than the total size of the tails of all movies, but the storage capacity of each PN is less than the total size of the tails of all movies. This case is much more common. For the resources that do not exist in local PN memory, PN submits the resource query to DS and sends data request to other PNs according to source selection policy when the request arrives. After the connection between PNs is established, PN will transmit data obtained from the source PN to MS at a constant rate. At the same time, MS will transmit data to users at a variable rate conforming to actual video playback.

Considering PNs' limit capabilities, not all the requests from MSs can be satisfied in time. So, PNs have to admit requests from MS according to certain rules and MS will react based on the admission decisions of PN. In this paper, we seek an adaptive method for PN to admit requests, which can effectively enhance the clients' confidence in the QoS of the system.

3. POMDP Model. In this section, we will briefly review the concept of POMDP at first. By analyzing the characteristics of P2P-based MDN admission control process, we demonstrate the fitness of POMDP model in solving this problem. Then, a mathematical model based on POMDP is developed under some reasonable assumptions.

3.1. POMDP. POMDP is the extension of Markov Decision Process (MDP), in which the state of MDP is not directly observable. POMDP allows for various types of uncertainty in the process. POMDP model is basically composed of the following elements.

S is a finite space of states. A is a finite space of actions. O is a finite space of observations. $p(j | i, a)$ is the state transition function, with $i, j \in S, a \in A$. $q(o | i)$ is the observation function, with $i \in S, o \in O$. $r(i, a, j)$ is the immediate reward function, with $i, j \in S$ and $a \in A$.

S represents all the possible underlying states in the process; however, these states are not directly observable. A contains all the available actions. O contains all the possible observations generated from the states in S . When i, j, a, o are fixed, $p(j | i, a)$ denotes the transition probability from state i to state j when taking the action a ; $q(o | i)$ denotes the probability of observed o under the state i ; $r(i, a, j)$ denotes the immediate reward that the controller can get as soon as the transition from state i to state j occurs when taking the action a .

3.2. Parameters in P2P-based MDN. P2P-based MDN is a network constructed on WAN. All the PNs are set by MDN providers and located at the edge of the network. This distributed architecture makes it better for all the PNs to conduct admission tasks independently. PNs make decisions only based on the information derived from the system's "current" state, so the decision process has Markov's characteristics. Meanwhile, independent control means that PN admits requests without knowledge of other PNs' serving conditions, and then the overall state of the system cannot be observed. All these characteristics fit the specific requirements of POMDP. Independent control may lack precision, but can save time in the communication among PNs. Before modeling, we list parameters of the system as follows.

N is the total number of PNs in the system. M is the total number of movies released by MCP. p_{im} denotes the storage status of the m -th movie at PN i . $p_{im} = 1$ when the tail of the m -th movie is stored at PN i ; otherwise, $p_{im} = 0$. We use an $N \times M$ matrix $P = [p_{im}]$ to describe global storage status. $\lambda_i^{(m)}$ denotes the arrival rate of requests for the m -th movie at PN i .

L_i is the upper limit of connection number of PN i with other PNs, including requests being served and waiting to be served. PN sets this limitation according to the service history. All PNs are located at the edge of network and the bottleneck bandwidth between each PN is much smaller comparing to the bandwidth between PN and local MS. For the movie requests which are locally hit, PN can transmit data to MS at high speed and save the time in querying and requiring. On the other hand, PN has to admit data requests from other PNs under the upper limit to shorten the service delay and guarantee the quality of service. Therefore, we only discuss admission control of PNs for data requests that are not locally hit.

At PN i , we use l_{ij} to denote the number of connections that are initiated by PN j ($j \neq i$) and let l_i be the total number of connections. Then $l_i = \sum_{j \neq i} l_{ij}$ with the restriction of $l_i \leq L_i$. Meanwhile, we set $l'_i = \sum_{j \neq i} l_{ji}$ to denote the total number of connections that PN i initiates.

λ_{ij} ($j \neq i$) denotes the arrival rate of requests for data in PN j at PN i . λ_{ij} can be estimated when the following parameters are known: (1) arrival rates of movie's requests at PN j ; (2) global storage status; (3) source selection policy of PN. The detailed method to compute λ_{ij} under certain source selection policy will be discussed in Section 5.

μ_{ij} ($j \neq i$) denotes the service rate that PN i provides services to PN j . μ_{ij} is determined by the following two elements: (1) the bandwidth that PN i can contribute to serve other

PNs; (2) the bandwidth allocation strategy that it adopts. In Section 5, we will especially present different bandwidth allocation schemes and discuss their impacts on service rates.

3.3. Modeling admission control based on POMDP. Before modeling, we give some assumptions to simplify the mathematical model.

- (1) Every PN selects one source peer from the source peer list for one request task.
- (2) A connection can be ended only if the service is rejected by the remote server or when the service is completed.
- (3) If the remote server rejects a request, local server will terminate the service directly or submit it to the VCS. Admission control of VCS can be treated as admission control of single server and will not be discussed in this paper.

Based on the above concepts and assumptions, the POMDP model is developed for P2P-based MDN admission control process.

State space S . We use an $N \times N$ integer matrix $s = [l_{ij}]$ to denote the system's state, with the following restrictions:

$$l_{ij} = 0, 1, \dots, L_i, \quad l_{ii} = 0, \quad l_i \leq L_i. \quad (1)$$

Elements which are called inter-state in our state matrix reflect the interactions between PNs. In addition, because each PN sets the upper connection limit individually, the state space is finite and has relatively fewer elements in it. Assuming that the number of $N \times N$ matrixes satisfying the restrictions in (1) is K , then the state space can be denoted as $S = \{s_1, s_2, \dots, s_K\}$, with

$$K = \prod_{i=1}^N \sum_{l_i=0}^{L_i} C_{l_i+N-2}^{N-2}. \quad (2)$$

We define observation space and action space based on the concept "event" [18]. An event happens at a particular time, which causes either action being taken or state transition. During the period of time between two successive events, the state of the system remains unchanged. In our model, events happening at PN i can be "a request arriving at PN i " or "a service for PN j ($j \neq i$) completed at PN i ".

Observation space O . When the event "a request arriving at PN i " takes place, PN i observes its own serving condition, such as checking up the total number of connections initiated by PN i ; when the event "a service for PN j completed at PN i " takes place, PN i observes that "a service is completed". We use $o = (i, n)$ to denote an observation that "PN i receives a data request when $l'_i = n$ " and o_0 to denote the observation that "a service is completed". Then, the observation space is given by

$$O = \left\{ o_0, o = (i, n) \mid i = 1, 2, \dots, N, n = 0, 1, \dots, \sum_{j \neq i} L_j \right\}, \quad (3)$$

with $\sum_i \sum_{j \neq i} L_j + N + 1$ elements in it.

Action space A . $A = \{a_0, a_1, a_2\}$, where a_0 denotes "taking no action"; a_1 denotes "request admitted"; a_2 denotes "request refused". a_0 is taken only when o_0 is observed.

Immediate reward function should be defined under specific example. Without consideration of the time between two successive events, the process of P2P-based MDN admission control can be abstracted as a Markov chain $\{x_l, l = 1, 2, \dots\}$ with a finite state space S . l is the serial number of the time point when a event happens. We name these time points "event points". x_l is the system state that we sample at event point l during the process.

State transition function. For any current state $s = [l_{ij}]$, we use $s^{\alpha\beta} = [l_{ij}^{\alpha\beta}]$ to denote the state of system when a data request from PN β has been accepted at PN α ,

in which

$$l_{ij}^{\alpha\beta} = \begin{cases} l_{ij} + 1, & i = \alpha, \quad j = \beta, \quad l_i = \sum_{j \neq i} l_{ij} < L_i, \\ l_{ij}, & \text{otherwise.} \end{cases} \quad (4)$$

If $\bar{s}^{\alpha\beta} = [\bar{l}_{ij}^{\alpha\beta}]$ denotes the state after a service for PN β is completed at PN α , then

$$\bar{l}_{ij}^{\alpha\beta} = \begin{cases} l_{ij} - 1, & i = \alpha, \quad j = \beta, \\ l_{ij}, & \text{otherwise.} \end{cases} \quad (5)$$

The transition probability function from state s to state s' is illustrated as follows.

$$p(s' | s, a_0) = \begin{cases} \frac{\epsilon(l_{\alpha\beta})\mu_{\alpha\beta}}{\mu(s)}, & s' = \bar{s}^{\alpha\beta}, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

$$p(s' | s, a_1) = \begin{cases} \frac{\lambda_{\alpha\beta}}{\lambda}, & s' = s^{\alpha\beta}, \\ \frac{\sum_{\alpha} \sum_{\beta} I_s(s^{\alpha\beta}) \lambda_{\alpha\beta}}{\lambda}, & s' = s, \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

$$p(s' | s, a_2) = \begin{cases} 1, & s' = s, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

in which $\lambda = \sum_i \sum_{j \neq i} \lambda_{ij}$ and $\mu(s) = \sum_i \sum_{j \neq i} \epsilon(l_{ij}) \mu_{ij}$. $\epsilon(x) = 0$ when $x = 0$; otherwise $\epsilon(x) = 1$. $I_a(b) = 1$ when $a = b$; otherwise $I_a(b) = 0$.

Observation function. Under the current state s , the observation probability is as follows.

$$q(o_0 | s) = \frac{\mu(s)}{\mu(s) + \lambda}, \quad (9)$$

$$q(o = (i, n) | s) = \begin{cases} \frac{\sum_{j \neq i} \lambda_{ji}}{\mu(s) + \lambda}, & n = l'_i, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

4. Admission Strategy. PN admission control for data requests from MS is a key link in the whole process of admission control of MDN. In this section, we will discuss admission steps, admission strategies in detail based on the above model.

4.1. Admission steps. As mentioned in Section 3, PNs have to set upper connection limits for data requests from other PNs. So, when a PN receives a request from MS, the PN checks up its storage status first. If PN lacks the resource, we call this request not locally hit, and PN will admit the request conditionally. The detailed admission steps for requests that are not locally hit are as follows.

Step 1. PNs communicate to know other PNs' capacities.

Step 2. PNs wait for a new request coming.

Step 3. A PN receives a data request from local MS, say PN i as an example. PN i checks up its local serving condition. If $l'_i = \sum_{j \neq i} L_j$, go to Step 8; otherwise, go to Step 4.

Step 4. PN i decides whether to admit this data request according to the admission strategy that it adopts. If the request is admitted, go to Step 5; otherwise, go to Step 8.

Step 5. PN i initiates a data request to another PN according to source selection policy, say PN j as an example. If $l_j = L_j$, go to Step 7; otherwise, go to Step 6.

Step 6. Data request is admitted successfully, l_{ji} plus one. PN i waits in queue at PN j and transmits data to local MS once it is served. Go to Step 2.

Step 7. PN j refuses the data request from PN i , so PN i informs MS of data request rejection. We call this kind of rejection as "Indirect Rejection". Go to Step 2.

Step 8. Data request is refused. We call this kind of rejection as “Direct Rejection”. Go to Step 2.

4.2. Prefix serving strategy. MS stores heads of the movies to decrease the service delay. The rest parts of movies are obtained from PNs. However, not all the data requests can be served in time due to the limitation of upload bandwidth. Some services might be denied for the congestion of network. Therefore, MS needs a certain strategy to reduce unnecessary costs.

Based on the admission steps mentioned above, we set a simple prefix serving strategy for MS, in which MS begins to pre-transmit the heads of movies to users after local PN admits the requests. The method for PNs to admit requests based on local serving conditions can effectively reduce response delay, which is used to measure the time difference between submitting request and sending admission decision back by controller. On the other hand, this method ignores the possibility that other PNs might have heavy workload. It is still risky of wasting the system’s resources. However, comparing to the way to pre-transmit data to all users who submit requests, it can reduce the cost of the system in a profound way.

To sum up, our admission steps have two traits: (1) PN decides whether to admit a new request and doesn’t need to inquire other PNs’ serving conditions; (2) MS pre-transmits data to clients for all admitted requests without consideration of the possibility of service failure. Services that are turned down indirectly by other PNs waste both time and system resource. Our work is to optimize the admission strategy to help PNs respond to requests more precisely, that is to say, to reduce the proportion of indirect rejections. In the following subsection, we will present the admission policy of each PN and the algorithm to optimize the policy.

4.3. Observation-based randomized policy. Observation-based randomized policy especially fits the condition of admission control with partially observable state. Although we may have the same observation when the underlying states are different, it is not wise to try to obtain the whole system’s state. Taking our model as an example, in order to record the system’s state, PNs have to submit every successfully accepted request or finished service to the central server and the server will have to check the authenticity of the information to avoid fraud. Making decision based on local observation can save the time and effort spending in saving and refreshing information. Thus, we only consider the observation-based policy, which is an active policy. Although an active policy may not be optimal, it can be implemented easily in practical systems and has higher efficiency [19].

We use $\mu(a | o, \theta)$, $\theta \in \Theta$ (use $\mu(\theta)$ for short) to denote a randomized and parameterized policy, which is a mapping from the observation space O to the probability measure set on the action space A . An action a is taken from the action space A according to probability distribution of $\mu(a | o, \theta)$ when the observation is o and the adjustable parameter θ is fixed.

We simulate the evolvement of the system offline based on the above model and the parameters we set. From each round of simulation, we learn the performance of a fixed admission policy and adjust it to a better-performing policy using policy-gradient algorithm. Normally, each round of simulation contains a fairly large number of event points to get more accurate performance of the policy $\mu(\theta)$. The long-term average payoff criterion is used to measure the performance of the policy, which is defined as:

$$\eta(\theta) = \lim_{L \rightarrow \infty} E^\theta \left[\frac{\sum_{l=1}^L r(x_l, a_l, x_{l+1})}{L} \right], \quad x_l \in S, \quad a_l \in A, \quad (11)$$

where E^θ denotes the expectation with respect to the policy $\mu(\theta)$, a_l is the action taken at event point l .

Based on the assumption that $\mu(a | o, \theta)$ is differentiable with respect to θ , the performance derivative formula is given by [19].

$$\frac{\partial \eta(\theta)}{\partial \theta} = \sum_{o \in O} \pi^\theta(o) \sum_{a \in A} \frac{\partial \mu(a | o, \theta)}{\partial \theta} Q^\theta(o, a), \quad (12)$$

where $\pi^\theta(o)$ is the stationary probability of observation o under policy $\mu(\theta)$; $Q^\theta(o, a)$ is the Q -function of observation o and action a under policy $\mu(\theta)$, it is given by

$$Q^\theta(o, a) = \lim_{L \rightarrow \infty} E^\theta \left\{ \sum_{l=1}^L [r(x_l, a_l, x_{l+1}) - \eta(\theta)] \mid o_1 = o, a_1 = a \right\}. \quad (13)$$

Assuming that in the k -th round, the performance gradient of the k -th policy θ_k is $\nabla \eta_k$, then using policy gradient algorithm, the parameter of the policy in the $k+1$ -th round will be changed into

$$\theta_{k+1} = \theta_k + \omega \nabla \eta_k, \quad (14)$$

where ω can be set by experience.

4.4. Estimation of parameters. Both $\pi^\theta(o)$ and $Q^\theta(o, a)$ in (12) can be solved theoretically, but the formula has to deal with matrix inversion and can hardly be realized with the expansion of the system. So, we will estimate them offline based on Monte Carlo method [19].

Assuming that there are L event points during one round of simulation and $o_1, o_2, \dots, o_L$ is the observation under policy $\mu(\theta)$. Then, $\pi^\theta(o)$ and $Q^\theta(o, a)$ can be estimated by following formulas:

$$\hat{\pi}^\theta(o) = \frac{\sum_{l=1}^L I_o(o_l)}{L}, \quad (15)$$

$$\hat{Q}^\theta(o, a) = \frac{\sum_{l=1}^{L-T} \{I_{o,a}(o_l, a_l) \sum_{t=1}^T [r(x_{l+t}, a_{l+t}, x_{l+t+1}) - \hat{\eta}(\theta)]\}}{\sum_{l=1}^{L-T} I_{o,a}(o_l, a_l)}. \quad (16)$$

where

$$\hat{\eta}(\theta) = \frac{\sum_{l=1}^L r(x_l, a_l, x_{l+1})}{L}, \quad (17)$$

$I_{o,a}(o_l, a_l) = 1$ when $o_l = o, a_l = a$; otherwise, $I_{o,a}(o_l, a_l) = 0$.

5. Other Strategies in P2P-Based MDN. There are some important strategies in P2P systems, such as source selection policy and bandwidth allocation policy. Source selection policy helps PNs select server nodes from the source node list. Server nodes divide their upload bandwidth among user nodes according to bandwidth allocation policy. These two policies are also applied in the process of file sharing among PNs in P2P-based MDN. In order to illustrate our POMDP model's applicability, we analyze the parameters in the model under some specific policies and propose modified policies based on the model.

5.1. Source selection policy. The value of λ_{ij} depends on $P, \lambda_j^{(m)}$ and source selection policy that PN j adopts. We take the Benchmark policy in [20] as an example, in which PNs always select source nodes by using random or rotating strategy. We use λ_{ijm} to denote the arrival rate of the requests for the m -th movie in PN j at PN i . Then, we have

$$\lambda_{ijm} = \begin{cases} \frac{\lambda_j^{(m)} p_{im}}{\sum_{k=1}^N p_{km}}, & p_{jm} = 0, \\ 0, & p_{jm} = 1, \end{cases} \quad (18)$$

$$\lambda_{ij} = \sum_{m=1}^M \lambda_{ijm}. \quad (19)$$

Statistic-based selection (SBS). According to the statistic results during simulation, we adjust Benchmark selection policy as follows:

$$\lambda_{ijm} = \begin{cases} \frac{\lambda_j^{(m)} p_{im} \rho_i}{\sum_{k=1}^N p_{km} \rho_k}, & p_{jm} = 0, \\ 0, & p_{jm} = 1, \end{cases} \quad (20)$$

with $\rho_i = \frac{1}{P_i}$. P_i is the probability that PN i is overload under certain policy and it can be estimated based on the stationary distribution of states during the simulation. PN i is said to be overload when the number of links at PN i equals to its capacity. If $P_i = 0$, ρ_i can be set large enough to ensure that PN j choose PN i with probability close to 1.

5.2. Bandwidth allocation policy. In some P2P applications, taking Emule as an example, PNs in the network only consider one service at a time by using the strategy “First Come First Served (FCFS)”. This strategy takes into account equity, but lacks efficiency. Although some services have small workload, they may have to wait for a long time if a large-workload service is occupying the bandwidth. Another method is to evenly allocate bandwidth for every ongoing service; however, this method leads to unstable service quality.

In our work, to consider both efficiency and stableness, PN reserves bandwidth for other PNs (but not all services) to reduce service delay. For the requests from the same PN, PN serves by using the strategy FCFS. In this way, each PN can be seen as $N - 1$ independent sub-servers. Let B_i denote the total bandwidth that PN i reserves for other PNs and b_{ij} denote the bandwidth that PN i allocates to PN j . Then $B_i = \sum_{j \neq i} b_{ij}$. In order to seek the most efficient strategy, we discuss three different bandwidth allocation strategies in this paper.

Even allocation policy. Each PN evenly allocates its bandwidth, so we have

$$b_{ij} = \frac{B_i}{N - 1}, \quad j \neq i. \quad (21)$$

Impartial service policy (ISP). PN i serves other $N - 1$ PNs at the same service rate μ_i , so service rates at PN i satisfy

$$\mu_{i1} = \cdots = \mu_{ij} = \cdots = \mu_{iN} = \mu_i, \quad j \neq i. \quad (22)$$

Partial service policy (PSP). Each PN allocates its bandwidth according to the arrival rates of other PNs’ requests. The simplest way is to make the service rates directly proportional to the requests’ arrival rates. Then, service rates satisfy

$$\frac{\mu_{ij}}{\lambda_{ij}} = C_i, \quad j \neq i, \quad (23)$$

where C_i is a constant set by PN i according to its total reserved upload bandwidth.

For ISP and PSP, we need to compute the expected value of bandwidth that PN should allocate for other PNs to get required service rates. Assuming that the length of the m -th movie is H_m , average bit rate is β_m and the percentage of the tail stored at PN is δ_m .

Let τ_{ij} be the number of services that PN i provides for PN j per unit time and τ_{ijm} be the number of services for the data of m -th movie. Then we have $E(\tau_{ij}) = \mu_{ij}$ and $\tau_{ij} = \sum_m \tau_{ijm}$. Let v_{ij} be the total amount of data that PN i should serve for PN j per unit time and v_{ijm} be the amount of data for the m -th movie. Assuming that, for each movie request, PN transmits all the data it stored to MS, then we have $v_{ijm} = \delta_m \beta_m H_m \tau_{ijm}$ and $v_{ij} = \sum_m v_{ijm}$.

Let p_{ijm} denote the probability that a service PN i providing for PN j is for data of the m -th movie, then we have $p_{ijm} = \frac{\lambda_{ijm}}{\lambda_{ij}}$. The expected value of τ_{ijm} under $\tau_{ij} = n$ is given by

$$E(\tau_{ijm}|\tau_{ij} = n) = \sum_{k=0}^n k C_n^k p_{ijm}^k (1 - p_{ijm})^{n-k} = np_{ijm}. \quad (24)$$

So, we get

$$E(\tau_{ijm}|\tau_{ij}) = \tau_{ij} p_{ijm}. \quad (25)$$

If let $\gamma_{ij} = \sum_m \delta_m \beta_m H_m p_{ijm}$, then we have

$$\begin{aligned} E(v_{ij}) &= \sum_m E(v_{ijm}) \\ &= \sum_m \delta_m \beta_m H_m E(\tau_{ijm}) \\ &= \sum_m \delta_m \beta_m H_m E[E(\tau_{ijm}|\tau_{ij})] \\ &= \sum_m \delta_m \beta_m H_m p_{ijm} E(\tau_{ij}) \\ &= \mu_{ij} \gamma_{ij}. \end{aligned} \quad (26)$$

PNs allocate bandwidth in advance according to the workload they estimate. That is to say, we set $b_{ij} = E(v_{ij})$.

In impartial service, from (22), we have

$$b_{ij} = B_i \frac{\gamma_{ij}}{\gamma_i}, \quad (27)$$

where $\gamma_i = \sum_{j \neq i} \gamma_{ij}$.

In partial service, from (23), we have

$$b_{ij} = B_i \frac{\kappa_{ij}}{\kappa_i}, \quad (28)$$

where $\kappa_{ij} = \sum_m \delta_m \beta_m H_m \lambda_{ijm}$, $\kappa_i = \sum_{j \neq i} \kappa_{ij}$.

6. An Numerical Example. In this section, we provide a numerical example to testify the effectiveness of our methods with four service scenarios. In the first three scenarios, we use Benchmark policy as source selection policy and choose even allocation policy, ISP and PSP separately as bandwidth allocation strategy. In the fourth scenario, both SBS and PSP are adopted.

The main parameters of the system we set are as follows: $N = 3$, $L_1 = L_2 = L_3 = 3$, $B_1 = 90$ (Mbit/s), $B_2 = 140$ (Mbit/s), $B_3 = 72$ (Mbit/s); $M = 10$, $H_1 = 84$ (min), $H_2 = 60$ (min), $H_3 = 120$ (min), $H_4 = 45$ (min), $H_5 = 90$ (min), $H_6 = 100$ (min), $H_7 = 65$ (min), $H_8 = 70$ (min), $H_9 = 125$ (min), $H_{10} = 90$ (min), $\beta_1 = \beta_2 = \dots = \beta_{10} = 2$ (Mbit/s), $\delta_1 = \delta_2 = \dots = \delta_{10} = 75\%$;

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

The admission probability under the observation $o = (i, n)$ is denoted as $b(i, n)$. The policy optimization begins with the Best Effort Service policy, which is realized by setting $b(i, n) = 1$ ($i = 1, 2, 3, n = 0, 1, \dots, 5$). The policy in the last round of optimization is defined as the “optimal policy”. Taking the third scenario as an example, the average

performance of admission policy gradually increases with optimization (as seen in Figure 2(a)). The rising of policy performance means that system providers can get more payoffs using the optimal policy. The proportion of indirect rejections in all rejections is named as “indirect rejection ratio”. As seen in Figure 2(b), indirect rejection ratio sharply reduces, which is about 55% of that under Best Effort Service policy. The optimal strategy ensures that most of the admitted requests can be served successfully. The success rate of accepted service under optimal policy increases about 17% (as indicated in Figure 2(c)).

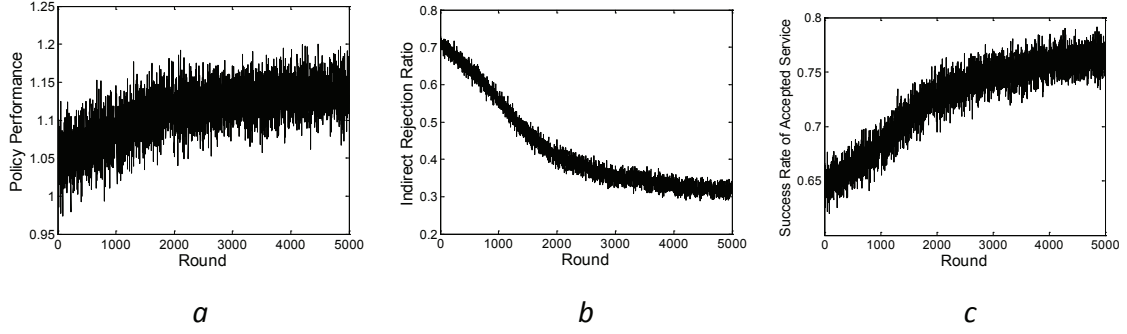


FIGURE 2. Variation of the system's performance *a*. Average policy performance *b*. Indirect rejection ratio *c*. Success rate in accepted services

Meanwhile, simulation results also show that, when the admission policy is fixed, PSP and SBS can bring more benefit, raise the bandwidth utilization rate and keep higher success rate (as shown in Figure 3).

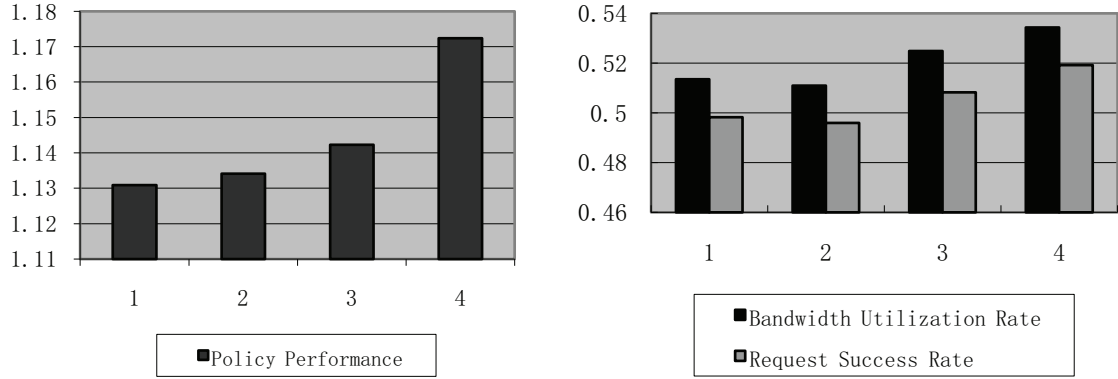


FIGURE 3. Comparisons of Policy and Service Performances under Four Scenarios

7. Conclusions. In this paper, a POMDP model is developed for P2P-based MDN systems. Specifically, the concept of inter-state is created to describe the overall state of the system, which can also be used in other P2P applications. On the basis of the model, an observation-based randomized policy is proposed and optimized by applying policy-gradient approach to implement the task of admission control. Simulation results show that requiring data based on simulation statistics and partially allocating the system's resource can increase the system's effectiveness in a better way.

Although the SBS policy that we propose can balance the workload of network to some extent; however, the effect still needs improvement. Further work includes modifying SBS policy, such as selecting source peers according to different statistics under different observations.

Acknowledgment. This work is supported in part by the National High Technology Research and Development Program of China (863 Program) under grant No. 2008AA01A317 and in part by the National Science Foundation of China under grant No. 60935001. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

- [1] E. Chang and A. Zakhor, Cost analyses for VBR video servers, *IEEE Multimedia*, vol.2, no.4, pp.56-71, 1996.
- [2] D. Makaroff, G. Neufeld and N. Hutchinson, An evaluation of VBR disk admission algorithms for continuous media file server, *Proc. of the 5th ACM International Conference on Multimedia*, pp.143-154, 1997.
- [3] H. M. Vin, P. Goyal, A. Goyal and A. Goyal, A statistic admission control algorithm for multimedia servers, *Proc. of ACM Multimedia*, pp.33-40, 1994.
- [4] R. Zimmermann and K. Fu, Comprehensive statistical admission control for streaming media servers, *ACM International Conference on Multimedia*, pp.75-85, 2003.
- [5] S. E. Kim and C. R. Das, A reliable statistical admission control strategy for interactive video-on-demand servers with interval caching, *Proc. of International Conference on Parallel Processing*, pp.135-142, 2000.
- [6] J. B. Kown and H. Y. Yeom, A statistical admission control scheme for continuous media servers using caching, *Multimedia Tools and Applications*, vol.19, no.3, pp.279-296, 2003.
- [7] K. Gopalan, L. Huang, G. Peng, T. C. Chiueh and Y. J. Lin, Statistical admission control using delay distribution measurements, *ACM Transactions on Multimedia Computing, Communication, and Applications*, vol.2, no.4, pp.258-281, 2006.
- [8] Z. H. Xia, W. Hao, I. L. Yen and P. Li, A distributed admission control model for QoS assurance in large-scale media delivery systems, *IEEE Transactions on Parallel and Distributed Systems*, vol.16, no.12, pp.1143-1153, 2005.
- [9] P. Mundur, A. K. Sood and R. Simon, Class-based access control for distributed video-on-demand systems, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.15, no.7, pp.844-853, 2005.
- [10] C. H. Lee, Y. Q. Gui, I. B. Jung, C. Y. Choi and H. K. Choi, A peer to peer prefix patching scheme for VOD servers, *Proc. of International Conf. on Information: New Generations*, pp.530-534, 2006.
- [11] C. L. Chan, S. Y. Huang and J. S. Wang, Performance analysis of proxy caching for VOD services with heterogeneous clients, *IEEE Transactions on Communications*, vol.55, no.11, pp.2142-2151, 2007.
- [12] K. H. Li, C. Q. Xu, Y. H. Zhang and Z. M. Wu, Optimal prefix caching and data sharing strategy, *IEEE Conference on Multimedia and Expo*, pp.465-468, 2008.
- [13] S. Kang and H. Yin, A hybrid CDN-P2P system for video-on-demand, *Proc. of the 2nd International Conf. on Future Networks*, Sanya, China, pp.309-313, 2010.
- [14] Q. Zheng, X. L. Zheng, J. Li, Q. C. Xiong and G. Wu, Performance analysis of MDN based on P2P architecture, *Proc. of the International Conf. on Advanced Computer Theory and Engineering*, pp.111-115, 2008.
- [15] X. Li and X. Gui, Tree-trust: A novel and scalable P2P reputation model based on human cognitive psychology, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3797-3808, 2009.
- [16] H. Zhang, C. Yin, J. Liu and F. Yuan, PNRC: Personalized news recommendation mechanism based on collaborative filtering in unstructured P2P networks, *ICIC Express Letters*, vol.3, no.3(B), pp.561-566, 2009.
- [17] D. K. Sepandar, T. S. Mario and G. M. Hector, The Eigentrust algorithm for reputation management in P2P networks, *Proc. of the 12th International Conf. on World Wide Web*, 2003.
- [18] X. R. Cao, Basic ideas for event-based optimization of Markov systems, *Discrete Event Dynamic Systems: Theory and Applications*, vol.15, no.2, pp.169-197, 2005.
- [19] Y. J. Li, B. Q. Yin and H. S. Xi, Partially observable Markov decision processes and performance sensitivity analysis, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol.38, no.6, pp.1645-1651, 2008.
- [20] H. Deshpande, M. Bawa and H. Garcia-Molina, Streaming live media over peers, *Technical Report*, Stanford University, <http://ilpubs.stanford.edu:8090/863/>, 2002.