An Efficient Solution to a Retrial Queue for the Performability Evaluation of DHCP

Tien Van Do

Department of Telecommunications,
Budapest University of Technology and Economics,
H-1117, Magyar tudósok körútja 2., Budapest, Hungary.
Tel: +36 1 463 2070

Abstract

Dynamic Host Configuration Protocol (DHCP) is designed to provide an automatic mechanism for the allocation, configuration and management of IP addresses and TCP/IP protocol stack parameters of computers and devices in IP networks. The important feature of DHCP is a "dynamic allocation" mechanism, which assigns an IP address to a client for a limited period of time (called a lease time). Therefore, a previously allocated IP address can automatically be assigned to another host by a DHCP server upon the expiration of the lease time if a host does not renew the use of an allocated IP address.

This paper proposes a retrial queueing model to approximate the performability of the DHCP dynamic allocation mechanism. An efficient computational algorithm is developed to calculate the steady state probabilities of a continuous time discrete state Markov process. A comparison of our approximate model with a detailed simulation model of the DHCP dynamic allocation mechanism shows an excellent agreement between the analytical and simulation results. The impact of the lease time parameter on the performability of the DHCP dynamic allocation mechanism is also illustrated through a numerical study.

Keywords: retrial queue, computational algorithm, DHCP dynamic allocation, lease time, performance model

Email address: `do@hit.bme.hu` (Tien Van Do).

1 Introduction

Dynamic Host Configuration Protocol (DHCP) is designed by the dynamic host configuration working group within the framework of the Internet Engineering Task Force (IETF). At the present, DHCP is specified for Internet Protocol version 4 in IETF "draft standard" RFC 2131 [1] and for Internet Protocol version 6 in IETF RFC 4361 [2]. The main aim of DHCP is to provide an automatic mechanism for the allocation, configuration and management of IP addresses and IP networking parameters (netmask, router IP address, etc) for computers and devices in IP networks.

The important feature of DHCP is a "dynamic allocation" mechanism, which assigns an IP address to a client for a limited period of time (called a lease time). Therefore, a previously allocated IP address which is not used by one host can automatically be assigned to another host by a DHCP server implementing the dynamic allocation mechanism. It is recognized that the appropriate setting of a lease time in a DHCP server plays an important role in the efficient allocation of IP addresses. In [3], the authors investigated the impact of setting lease times using the data from the Georgia Tech campus network. However, due to the lack of a quantitative performability model and the lack of data at clients (whether they are forced to wait for an IP address), they only examined the utilization of the allocatable address space in a DHCP server.

This paper proposes a method to quantitatively evaluate the performance of a DHCP dynamic allocation mechanism and the impact of a lease time. To construct a retrial queue and a tractable solution, the following steps are performed. We show that interarrival times of DHCP requests from clients follow the exponential distribution. We make a relaxed assumption concerning the lease time sent by a DHCP server and the retrials of clients. We develop an efficient computational algorithm to calculate the steady state probabilities and the performance measures of a continuous time discrete state Markov (CTMC) process associated with the proposed retrial queue. It is shown via simulation of more detailed model than an analytical abstract model of DHCP that the proposed model is accurate to calculate the performance of the interaction between the behavior of clients and the DHCP mechanism. A numerical study is also performed, which provides an insight for the impact of trade-off parameters and factors on the operation of DHCP.

The rest of this paper is organized as follows. In Section 2, the overview of DHCP operation is presented. In Section 3, the proposed model and a computational algorithm is described. In Section 4 a numerical study is provided to reveal some interesting behaviors of the IP address allocation mechanism. Finally, the paper is concluded in Section 5.

2 Overview of DHCP operation

The operation of DHCP assumes two roles. A centralized DHCP server manages a range of IP addresses allocated by a network administrator for a specific IP subnet. The communications between a DHCP server and a client are delivered by the DHCP protocol. A DHCP client software running on computers or devices normally sends a broadcast query (DHCPDISCOVERY message) requesting information from a DHCP server. The DHCP server checks whether the message is sent from the client with a permissible Media Access Control (MAC) address. If the client is authorized, the server assigns the client an IP address, a lease time, the subnet mask and the default gateway address encapsulated in the DHCPOFFER message.

Note that the whole process is performed in the similar way, if a client knows the IP address of a DHCP server in advance of the request of an IP address. The only exception is that a client sends DHCPREQUEST message instead of DHCPDISCOVERY message.

Three main modes for IP address allocation are supported: manual, automatic and dynamic allocation. The purpose of the "manual allocation" mode is to allow the network administrator to centrally store information concerning client hosts. In this mode the IP address is assigned by the network operator to a client host. After the identification of a specific client (e.g. based on hardware MAC address) DHCP sends a fixed IP address and configuration parameters (e.g.: the subnet mask, the default gateway address) for the client. This kind of operation is typically applied in a campus or LAN environment. In the mode "automatic allocation", a DHCP server assigns a permanent IP address to a client host.

The most important feature of DHCP is the "dynamic allocation" mechanism, which assigns an IP address to a client for a limited period of time. A lease time is defined as a period of time for which the server gives a permission for a client to use the address. Note that a lease time is also sent to a client. Upon the expiration of the lease time, the allocated address becomes free and can be assigned to another client unless a client extends the right to use a specific IP address before the expiration of the lease time. This feature is often applied in the environment of Internet Service Providers because the reuse of scarce IP addresses is possible.

The decision that a DHCP client "leaves" the system or renews the use of the allocate IP address depends on the relation between the lease time and the holding time (e.g.: the working time) of clients. In order to extend the use of the allocated IP address the client sends a DHCPREQUEST message which includes the client's allocated IP address in the "requested IP address" option of a DHCPREQUEST message.

3 A Proposed Model

3.1 A Retrial Queue

The size of the pool (i.e.: the number of allocatable IP addresses) is c. The fix lease time value sent by the DHCP server is denoted by T_l.

We assume the interarrival times of DHCP DISCOVERY messages are exponentially [1] distributed with a mean interarrival time $1/\lambda$.

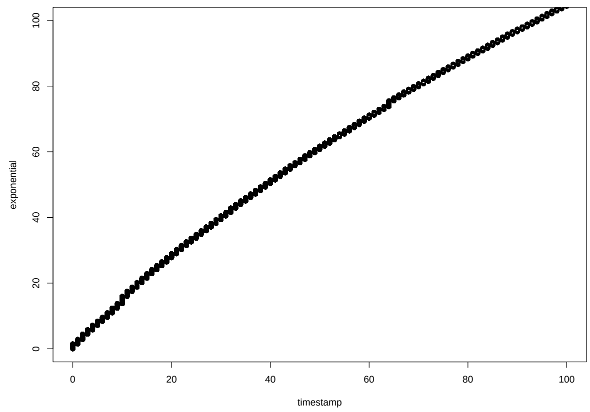


Fig. 1. Q-Q plot for the interarrival times (measured in seconds) of DHCPDISCOVERY messages

Assume that the holding times (i.e.: how long does a client need an IP address) of clients are represented by random variable H with a cumulative distribution function $\Pr(H < x) = F(x)$. Upon the expiration of the lease time, the previously allocated address at the DHCP server becomes free and can be allocated to another client unless the client extends the use of a specific IP address before the expiration of the lease time. Let a denote the probability that DHCP clients leave (i.e.: switch off the computer) the system or do not

[1] We process the log file of the DHCP server of our department between the period of January 2 and May 28, 2008. In Figure 1, the straight line of the Q-Q plot, where the interarrival times of DHCP requests between 8h and 18h during the investigation period to the DHCP server are plotted against the theoretical exponential distribution, confirms our assumption.

renew the allocated IP address after the expiration of its lease time. We can write

$$a = \Pr(H < T_l) = F(T_l).$$

It is worth emphasizing that there is no specific assumption concerning about the relation of the average holding time and the lease time in our model.

$I(t)$ denotes the number of allocated IP addresses at time t. Note that $0 \le I(t) \le c$ holds. A client who does not receive the allocation of an IP address because the shortage (when $I(t) = c$) of IP addresses sets a timer to wait for a limited time and will retry the request for an IP address upon the expiration of backoff time. We model this phenomenon as the client joins the "virtual orbit". $J(t)$ represents the number of DHCP clients in the "orbit" at time t and takes values from 0 to ∞.

In order to have a mathematically tractable model, we make the following assumptions.

- Lease times are exponentially distributed with a mean lease time $1/\mu = T_l$.
- Clients waiting in the orbit repeat the request for the DHCP server with rate ν (i.e.: the inter-repetition times are exponentially distributed with parameter ν), which is independent from the number of waiting clients in the orbit.

Therefore, the presented approach below is the application of an approximate model for the DHCP mechanism presented in Section 2. It will be shown in Section 4 (through the comparison with the simulation of the DHCP mechanism) that the approximate model provides a quite good prediction for the performance measures of the DHCP dynamic allocation mechanism.

As a consequence, the system is modeled by a CTMC, $Y = \{I(t), J(t)\}$, with a state space $\{0, 1, \ldots, c\} \times \{0, 1, \ldots\}$.

Remarks: The stationary distributions of the main M/M/c retrial queue with $c > 2$ can be computed using approximation techniques [4–6]. Falin and Templeton proposed a truncation model and a numerical tractable solution with a threshold in their book [6], which is followed by the work [7]. The retrial queue presented in this paper is indeed a numerically tractable model [6] with 0 threshold value. However, only matrix-geometric solution is suggested in [6]. We show in the later section that we develop an efficient computational algorithm for the considered retrial queue and the evaluation of the DHCP dynamic allocation mechanism based on the considered retrial queue is accurate.

We denote the steady state probabilities by $\pi_{i,j} = \lim_{t\to\infty} Prob(I(t) = i, J(t) = j)$, and introduce $\mathbf{v}_j = (\pi_{0,j}, \ldots, \pi_{c,j})$.

The evolution of Y is driven by the following transitions.

(a) $A_j(i,k)$ denotes a transition rate from state (i,j) to state (k,j) $(0 \leq i,k \leq c; j = 0,1,\ldots)$, which is caused by either the arrival of DHCPDISCOVERY requests or by the expiration of the lease time without the renewal of an allocated IP address. Matrix A_j is defined as the matrix with elements $A_j(i,k)$. Since A_j is j-independent, it can be written as

$$A_j = A = \begin{bmatrix} 0 & \lambda & 0 & \ldots & 0 & 0 & 0 \\ a\mu & 0 & \lambda & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \ldots & a(c-1)\mu & 0 & \lambda \\ 0 & 0 & & \ldots & 0 & ac\mu & 0 \end{bmatrix} \quad \forall j \geq 0;$$

(b) $B_j(i,k)$ represents one step upward transition from state (i,j) to state $(k,j+1)$ $(0 \leq i,k \leq c; j = 0,1,\ldots)$, which is due to the arrival of DHCPDISCOVERY requests when no free IP address is available in the IP address pool. In the similar way, matrix B_j (B) with elements $B_j(i,k)$ is defined as

$$B_j = B = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \ldots & 0 & 0 & 0 \\ 0 & 0 & & \ldots & 0 & 0 & \lambda \end{bmatrix} \quad \forall j \geq 0;$$

(c) $C_j(i,k)$ is the transition rate from state (i,j) to state $(k,j-1)$ $(0 \leq i,k \leq c; j = 1,\ldots)$, which is due to the successful retrial of a request from the orbit. Matrix C_j $(\forall j \geq 1)$ with elements $C_j(i,k)$ is written as

$$
C_j = C = \begin{bmatrix} 0 \ \nu \ 0 \ \dots \ 0 \ 0 \ 0 \\ 0 \ 0 \ \nu \ \dots \ 0 \ 0 \ 0 \\ \vdots \ \vdots \ \vdots \ \vdots \quad \vdots \ \vdots \ \vdots \\ 0 \ 0 \quad \dots \ 0 \ 0 \ \nu \\ 0 \ 0 \quad \dots \ 0 \ 0 \ 0 \end{bmatrix} \quad \forall j \geq 1.
$$

D^A and D^C are diagonal matrices whose diagonal elements are the sum of the elements in the corresponding row of A and C, respectively. The infinitesimal generator matrix of Y can be written as follows

$$
\begin{bmatrix} A_{00} & \mathrm{B} & 0 & \dots & \dots & \dots & \dots \\ \mathrm{C} & Q_1 & \mathrm{B} & 0 & \dots & \dots & \dots \\ 0 & \mathrm{C} & Q_1 & \mathrm{B} & 0 & \dots & \dots \\ 0 & 0 & \mathrm{C} & Q_1 & \mathrm{B} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}, \tag{1}
$$

where $A_{00} = A - D^A - B$ and $Q_1 = A - D^A - B - D^C$.

Because of the special structure of the QBD, the steady state probabilities can be obtained with the existing methods like the matrix-geometric and its variants [8–10], and the spectral expansion [11]. However, the existing methods have the "state-space explosion" problem when c is large. The problem starts when c reaches a value of several hundreds (no results or a very long-running time of computer programs implementing these methods). Therefore, in what follows we present an efficient computational procedure to find the steady state probabilities.

3.3 An Efficient Computational Procedure

For $j \geq 1$, the balance equations are written as follows

$$
\mathbf{v}_{j-1}B + \mathbf{v}_j Q_1 + \mathbf{v}_{j+1}C = 0 \ \ (j \geq 1). \tag{2}
$$

$Q(x) = B + Q_1 x + Cx^2$ is defined as the characteristic matrix polynomial associated with equations (2). In the present paper, $Q(x)$ is a tridiagonal matrix

$$Q(x) = \begin{bmatrix} q_{11}(x) & \lambda x + \nu x^2 & 0 & \dots & 0 & 0 & 0 \\ a\mu x & q_{2,2}(x) & \lambda x + \nu x^2 & \dots & 0 & 0 & 0 \\ 0 & a2\mu x & q_{3,3}(x) & \lambda x + \nu x^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \dots & a(c-1)\mu x & q_{c,c}(x) & \lambda x + \nu x^2 \\ 0 & 0 & & \dots & 0 & ac\mu x & q_{c+1,c+1}(x) \end{bmatrix}$$

where

$$\begin{aligned} q_{1,1}(x) &= -(\lambda + \nu)x, \\ q_{i,i}(x) &= -(\lambda + \nu + (i-1)a\mu)x \quad (i = 2, \dots, c), \\ q_{c+1,c+1}(x) &= \lambda - (\lambda + c\mu a)x. \end{aligned}$$

The steady state probabilities are closely related to the eigenvalue-eigenvector pairs $(x, \boldsymbol{\psi})$ of $Q(x)$, which satisfy $\boldsymbol{\psi}Q(x) = 0$ and $det[Q(x)] = 0$ (c.f. [11]). It is easy to see that $Q(x)$ has c zero-eigenvalues. The corresponding independent eigenvectors for c zero-eigenvalues are $\boldsymbol{\psi}_1 = \{1, 0, \dots, 0\}$, $\boldsymbol{\psi}_2 = \{0, 1, 0, \dots, 0\}$,$\dots$,$\boldsymbol{\psi}_c = \{0, 0, \dots, 1, 0\}$.

Note that if the system is ergodic, then the number of eigenvalues of $Q(x)$ of a QBD process, which are inside the unit disk, is $c+1$ (c.f. [11]). Therefore, $Q(x)$ should have a single eigenvalue x_0 inside the unit disk because $Q(x)$ has c zero-eigenvalues. Let $\boldsymbol{\psi}_0$ the corresponding left-hand-side eigenvector of $Q(\lambda)$ for the eigenvalue x_0.

As a consequence, the steady state probabilities can be expressed as follows

$$\begin{aligned} \mathbf{v}_j &= b_0 \boldsymbol{\psi}_0 x_0^j \ (j \geq 1), \\ \mathbf{v}_0 &= \sum_{k=0}^{c} b_k \boldsymbol{\psi}_k, \end{aligned} \tag{3}$$

where b_i are the coefficients to be determined. Since the probabilities are greater or equal 0, $0 < x_0 < 1$ holds.

The straightforward way to obtain the steady state probabilities is to find the eigenvalues of $Q(x)$ (see [12] for the methodology to find the eigensystem of the matrix polynomial). Then, one could use the balance equation for level 0

$$\mathbf{v}_0 A_{00} + \mathbf{v}_1 C = 0 \tag{4}$$

and the normalisation equation

$$\sum_{i=0}^{c}\sum_{j=0}^{\infty}\pi_{i,j}=\sum_{k=1}^{c}b_k\boldsymbol{\psi}_k\mathbf{e}+b_0\boldsymbol{\psi}_0\mathbf{e}/(1-x_0)=1 \tag{5}$$

to determine the coefficients b_i. Note that $\mathbf{e}$ is a $1\times(c+1)$ vector with all elements equal 1.

The key step towards the steady state probabilities is to determine x_0 and the corresponding eigenvector $\boldsymbol{\psi}_0$.

Theorem 1 $0<x_0<1$ *is the root of* $l_{c+1}(x)$*, the last diagonal element of* $L(x)$ *when we make the LU decomposition of* $Q(x)=L(x)U(x)$.

Proof. Since $Q(x_0)$ is a tridiagonal matrix and $q_{i,i}(x_0)\neq 0$, the component matrices of the LU decomposition of $Q(x_0)$ are written as

$$L(x_0)=\begin{bmatrix} l_1(x_0) & 0 & 0\dots 0 & 0 & 0 \\ a\mu x_0 & l_2(x_0) & 0\dots 0 & 0 & 0 \\ \vdots & \vdots & \vdots\ \vdots\ \vdots & \vdots & \vdots \\ 0 & 0 & \dots a(c-1)\mu x_0 & l_c(x_0) & 0 \\ 0 & 0 & \dots 0 & ac\mu x_0 & l_{c+1}(x_0) \end{bmatrix}, \tag{6}$$

$$U(x_0)=\begin{bmatrix} 1 & u_1(x_0) & \dots & 0 & 0\ 0\ 0 \\ 0 & 1 & u_2(x_0) & \dots & 0\ 0\ 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots\ \vdots\ \vdots \\ 0 & 0 & & \dots & 0\ 1\ u_c(x_0) \\ 0 & 0 & & \dots & 0\ 0\ 1 \end{bmatrix} \tag{7}$$

where $l_i(x_0)$ $(i=1,\dots,c+1)$ and $u_i(x_0)$ $(i=1,\dots,c)$ are the elements of $L(x_0)$ and $U(x_0)$, respectively. After a simple algebra, it can be written as

$$\begin{aligned} l_1(x_0) &= q_{1,1}(x_0)=-(\lambda+\nu)x_0 \\ l_i(x_0)+a(i-1)\mu x_0 u_{i-1}(x_0) &= q_{i,i}(x_0),\quad (i=2,\dots,c+1) \\ l_i(x_0)u_i(x_0) &= \lambda x_0+\nu x_0^2,\quad (i=1,\dots,c). \end{aligned} \tag{8}$$

Therefore, the determinant of $Q(x_0)$ is expressed as

$$Det[Q(x_0)]=Det[L(x_0)]Det[U(x_0)]=\prod_{i=1}^{c+1}l_i(x_0) \tag{9}$$

As the consequence of equation (8), $l_i(x_0) \neq 0$ $(1 < i \leq c)$. Hence, $Det[Q(x_0)] = 0$ follows $l_{c+1}(x_0) = 0$. $\square$

It is also easy to prove that $l_{c+1}(0)$ is positive and $l_{c+1}(1)$ is negative. Therefore, a bisection algorithm can be proposed to determine x_0 as illustrated in Algorithm 1. Note that the recursive relations (see Algorithm 1) between $\psi_{0,i}$ and $\psi_{0,i+1}$ $(i = c, \ldots, 1)$ are easily derived from equation

$$\boldsymbol{\psi}_0 Q(x_0) = \boldsymbol{\psi}_0 L(x_0) U(x_0) = 0.$$

Based on the property that multiplying an eigenvector with a scalar number results in an eigenvector, we can determine $\boldsymbol{\psi}_0 = \{\psi_{0,1}, \psi_{0,2}, \ldots, \psi_{0,c+1}\}$ by setting $\psi_{0,c+1} = 1$ and using the recursive relations.

Algorithm 1 Bisection algorithm to determine x_0 and the calculation of $\boldsymbol{\psi}_0$

```
Initialize the required accuracy ε
x_{0,u} = 1.0, x_{0,d} = 0
repeat
   x_0 = (x_{0,u} + x_{0,d}) / 2
   calculate l_{c+1}(x_0) based on equation (8)
   if l_{c+1}(x_0) > 0 then
      x_{0,d} = x_0
   else
      x_{0,u} = x_0
   end if
until |l_{c+1}(x_0)| < ε
ψ_{0,c+1} = 1
for i = c to 1 do
   ψ_{0,i} = (−aiμx_0 / l_i(x_0)) ψ_{0,i+1}
end for
return x_0, ψ_0
```

Let us introduce $\mathbf{b}_{cf} = \sum_{i=1}^{c} b_i \boldsymbol{\psi}_i = \{b_1, b_2, \ldots, b_c, 0\}$. From equations (3) and (4), we can write

$$\sum_{i=0}^{c} b_i \boldsymbol{\psi}_i \left[D^A + B - A\right] = b_0 \boldsymbol{\psi}_0 x_0 C$$

$$\mathbf{b}_{cf} \left[D^A + B - A\right] = b_0 \boldsymbol{\psi}_0 x_0 C - b_0 \boldsymbol{\psi}_0 \left[D^A + B - A\right] \tag{10}$$

The computation of the coefficients is based on the following observations

- both the left and right hand side of equation (10) are vectors.
- the last element (i.e.: element $c + 1$) of the left hand side of equation (10) contains only b_c.

- only b_c and b_{c-1} are in element c of the left hand side of equation (10). only b_{c-1}, b_{c-2} and b_{c-3} are in element $c-1$ of the left hand side of equation (10), etc.

As a consequence, b_c can be expressed in b_0. Then, b_i $(i = c-1, \ldots, 1)$ can be calculated recursively. That is, all b_i $(i = c, \ldots, 1)$ can be given in terms of b_0. Thus, b_0 can be determined from the normalization equation. Then other coefficients $b_1, \ldots, b_c$ are calculated.

3.4 Performance measures

Note that performance parameters related to the DHCP dynamic allocation mechanism are obtained as follows:

- average number of occupied IP addresses

$$N_{occ} = \sum_{i=1}^{c} i \sum_{j=0}^{\infty} \pi_{i,j} = \sum_{i=1}^{c} i(\pi_{i,0} + \sum_{j=1}^{\infty} b_0 \boldsymbol{\psi}_{0,i+1} x_0^j) = \sum_{i=1}^{c} i(\pi_{i,0} + \frac{b_0 \boldsymbol{\psi}_{0,i+1} x_0}{1 - x_0}), \tag{11}$$

- average number of clients waiting in the orbit

$$N_{orbit} = \sum_{j=1}^{\infty} j \sum_{i=0}^{c} \pi_{i,j} = \sum_{j=1}^{\infty} j b_0 \sum_{i=0}^{c} \boldsymbol{\psi}_{0,i+1} x_0^j = \frac{b_0 x_0}{(1-x_0)^2} \sum_{i=0}^{c} \boldsymbol{\psi}_{0,i+1}. \tag{12}$$

4 Case study

Three scenarios are investigated in this section. The first scenario represents a case which may happen in a private company or in a small campus. In this case, a small number ($c = 250$) of IP addresses can be allocated to clients. The second and third scenarios correspond to a case where a large number ($c = 1000$ and $c = 3000$) of IP addresses are available to clients. For three cases, we choose $1/\nu = 30$ seconds and the exponential distribution of holding times (i.e.: $F(x) = 1 - e^{-x/t_h}$), where t_h is the mean holding time. The request rate, λ, is of 1 request/minute for the first scenario, and 6 requests/minute for the second and third scenario. The computational time of the proposed algorithm largely depends on c. On a machine with Intel® Xeon® E5410 2.33GHz processor, the computation took **0.28** seconds of CPU time for $c = 250$, **1.06** seconds for $c = 1000$ and **3.04** seconds for $c = 3000$.

For the first case, we present the comparison of our model with simulation.

Lease time: $T_l = 5$ minutes				
Average holding time t_h (minutes)	Analytical Model		Simulation (conf. level=99%)	
	N_{occ}	N_{orbit}	N_{occ}	N_{orbit}
10	12.7075	0	12.715630	0
30	32.5694	0	32.590180	0
60	62.5347	0	62.574540	0
90	92.5231	0	92.582617	0
120	122.5170	0	122.596091	0
150	152.5140	0	152.612031	0
180	182.5120	0.000004	182.628786	0.000002
Lease time: $T_l = 30$ minutes				
10	31.5719	0	31.591576	0
30	47.4593	0	47.490397	0
60	76.2448	0	76.293760	0
90	105.832	0	105.899821	0
120	135.624	0	135.709828	0
150	165.500	0	165.605376	0
180	195.416	0.000437	195.540905	0.000423
Lease time: $T_l = 60$ minutes				
10	60.1491	0	60.186734	0
30	69.3911	0	69.436389	0
60	94.9186	0	94.980781	0
90	123.309	0	123.385416	0
120	152.49	0	152.587474	0
150	181.995	0	182.111794	
180	211.664	0.03816	211.799544	0.033517
Lease time: $T_l = 90$ minutes				
10	90.0111	0	90.068141	0
30	94.7156	0	94.774729	0
60	115.850	0	115.921970	0
90	142.378	0	142.471148	0
120	170.573	0	170.679458	0
150	199.473	0.001519	199.600627	0.001356
180	228.734	1.299020	228.881151	1.016362
Lease time: $T_l = 120$ minutes				
10	120.001	0	120.076530	0
30	122.239	0	122.315397	0
60	138.782	0	138.872755	0
90	162.954	0	163.052606	0
120	189.837	0.000067	189.961514	0.000060
150	217.916	0.154400	218.052797	0.114559
180	246.618	67.667500	247.106000	66.585000

Table 1
Analytical and simulation results ($c = 250$, $\lambda = 1$ requests/minute)

We have developed an own simulation program [2] in language C based on the SimPack toolkit [3] and the statistical module [4] from Politecnico di Torino, which have been used for many simulation studies. Note that the simulation model follows the real interaction of clients and the DHCP mechanism as much as possible. Therefore, it is different from the analytical model presented in Section 3.1 in three aspects:

- the retrial rate from the orbit: in the simulation the retrial rate depends on $J(t)$ (i.e.: each waiting client retrial after $1/\nu$), while the retrial rate in the queueing model is of fixed value when $J(t) > 0$.
- the holding time: in the simulation we simulate the phenomenon of the holding time of a specific request [5], while in the queueing model we use parameter a to take into account the phenomenon of the holding time.
- the lease time: the allocated lease times are of fixed value in the real DHCP operation and our simulation model, while the lease times are exponentially distributed in the queueing model.

That means, the simulation model does not follow the assumption of the analytical one. Note that the simulation results are generated with the confident level of 99%. Simulation runs are stopped when the relative precision (i.e.: the ratio of the half-width of the confidence interval and the mean of collected observations) of N_{occ} reaches 0.099%. The collected measures for N_{orbit} show high variability and the relative precision of N_{orbit} is $\pm 49\%$. As observed from Table 1 the agreement between the simulation and analytical results is excellent concerning N_{occ}. The analytical values of N_{orbit} are within the confidence interval.

We plot the average number of occupied IP addresses versus the average holding time and the lease time in Figures 2, the average number of requests waiting in the orbit versus the average holding time and the lease time in Figure 3, and the probability that all IP addresses are being allocated in Figure 4. It can be observed that the system is overloaded when the average holding time is higher than 200 minutes.

The most important resource of the DHCP server is the pool of IP address, so the efficient allocation of IP address poses a crucial issue for the network

[2] `http://www.hit.bme.hu/~do/dhcpmodeling/dhcp.c`

[3] `http://www.cise.ufl.edu/~fishwick/simpack.html`

[4] We use the statistical module (`http://www.telematica.polito.it/class/statistics.ps.gz`) to collect simulation data and to perform the analysis of simulation runs.

[5] The lease time sent to each a client is of a fixed value in a specific simulation and each client independently retries an IP requests after 30 seconds (it is the normal value observed in a DHCP client software implemented in the present operating systems).

administrator. As one observes that the allocation of IP addresses can be controlled with the appropriate setting of the lease length. If the DHCP is not overloaded, then the smaller the lease time is, the more efficient the allocation of IP address (Figure 2) and the smaller the number of requests waiting in the orbit is (Figure 3). For example in the second scenario when the average holding time is 90 minutes and a lease time has a value of 30 minutes, the average number of occupied IP addresses is 635 (365 free IP addresses are available in average). If we change the setting of a lease time to 120 minutes, only 186 free IP addresses are available in a DHCP server. It is worth emphasizing that the small value setting of the lease time has the impact of increased number (load) of renewal messages (DHCPREQUEST). Similar observations can be obtained in the third scenario (Figure 6) as well (the only difference between the second and third scenario that we increase the size of the IP address pool to 3000).

In Figure 5, we show the rate of renewal messages versus the lease time and the average holding time. We observe that the smaller the lease time is, the larger the rate of renewal messages is, which contrasts with the behavior of the average number of occupied IP addresses versus the lease time and the average holding time (Figure 2). Therefore, the trade-off parameter of the DHCP dynamic allocation mechanism is the rate of renewal messages. That is, the choice of an appropriate lease time depends on the processing capacity (how many messages can be handled during one minute or one second) of a DHCP server.

5 Conclusions

We have provided a methodology to evaluate the performability of the DHCP dynamic allocation mechanism. It can be used to determine the appropriate size of the IP address pool in a DHCP server and to set an appropriate lease time.

We have observed that the setting of a small lease time in a DHCP server has the advantage of the more efficient usage (i.e.: more clients can be allocated) of the IP address pool and the smaller number of clients waiting in the orbit than a large lease time. It is also worth emphasizing that we also have to take into account the load of renewal messages when we want to set a small lease time (i.e: a DHCP server is powerful enough to handle renewal messages).

References

[1] R. Droms, Dynamic Host Configuration Protocol, RFC 2131 (Draft Standard), updated by RFCs 3396, 4361 (Mar. 1997).
URL `http://www.ietf.org/rfc/rfc2131.txt`

[2] T. Lemon, B. Sommerfeld, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4), RFC 4361 (Proposed Standard) (Feb. 2006).
URL `http://www.ietf.org/rfc/rfc4361.txt`

[3] M. Khadilkar, N. Feamster, M. Sanders, R. Clark, Usage-based dhcp lease time optimization, in: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, ACM, New York, NY, USA, 2007, pp. 71–76. doi:http://doi.acm.org/10.1145/1298306.1298315.

[4] J. R. Artalejo, A. Economou, A. Gómez-Corral, Applications of maximum queue lengths to call center management, Computers & OR 34 (4) (2007) 983–996.

[5] J. R. Artalejo, A. Gómez-Corral, Retrial Queueing Systems, Springer, 2008.

[6] G. I. Falin, J. G. C. Templeton, Retrial Queues, Chapman & Hall, London, 1997.

[7] J. R. Artalejo, A. Economou, A. Gómez-Corral, Algorithmic analysis of the Geo/Geo/c, European Journal of Operational Research 189 (3) (2008) 1042–1056.

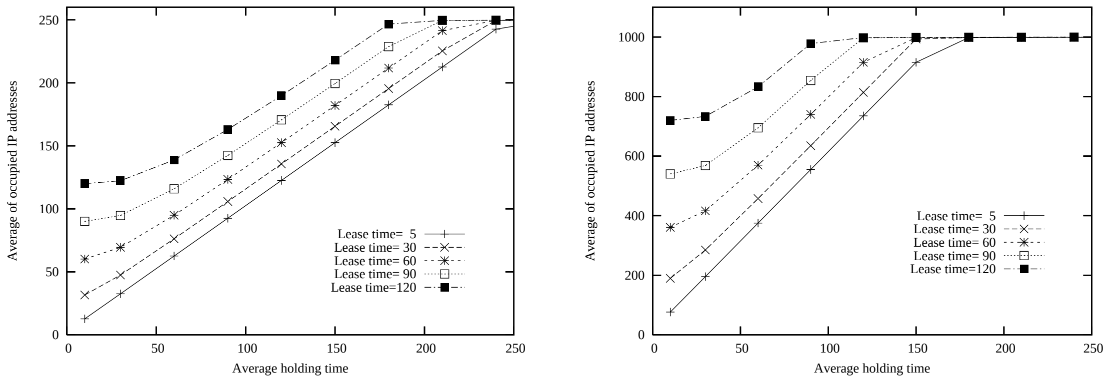
[8] D. Bini, B. Meini, On the solution of a nonlinear matrix equation arising in queueing problems, SIAM Journal on Matrix Analysis and Applications 17 (4) (1996) 906–926.

[9] G. Latouch, V. Ramaswami, A logarithmic reduction algorithm for quasi-birth-death processes, Applied Probability (1993) 650–674.

[10] V. Naoumov, U. Krieger, D. Wagner, Analysis of a Multi-server Delay-loss System with a General Markovian Arrival Process, in: S. Chakravarthy, A. Alfa (Eds.), Matrix-analytical methods in Stochastic models, Marcel Dekker, 1997, pp. 43–66.

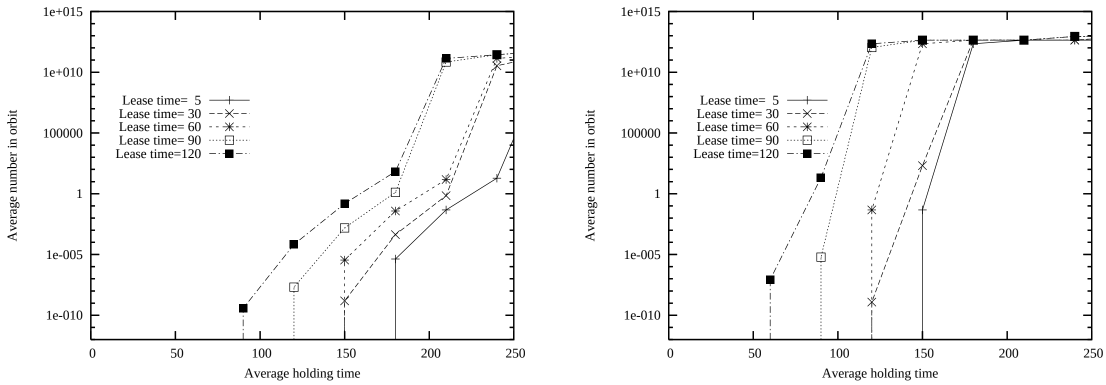
[11] I. Mitrani, R. Chakka, Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method, Performance Evaluation 23 (1995) 241–260.

[12] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (Eds.), Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia.



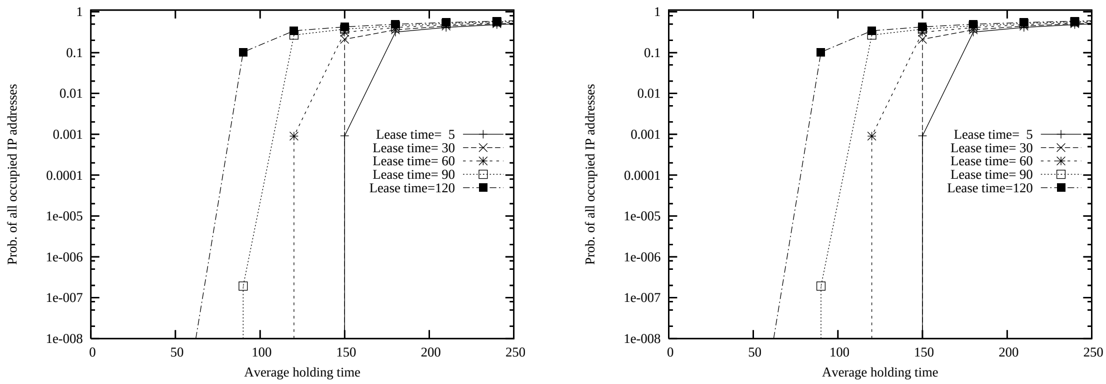
(a) $c = 250$, $\lambda = 1$ requests/minute (b) $c = 1000$, $\lambda = 6$ requests/minute

Fig. 2. Average number of occupied IP addresses



(a) $c = 250$, $\lambda = 1$ requests/minute (b) $c = 1000$, $\lambda = 6$ requests/minute

Fig. 3. Average number of requests waiting in the orbit



(a) $c = 250$, $\lambda = 1$ requests/minute (b) $c = 1000$, $\lambda = 6$ requests/minute

Fig. 4. Probability that all IP addresses are being allocated

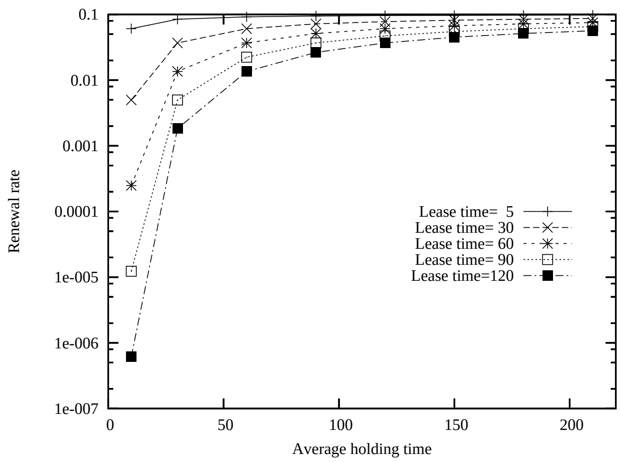
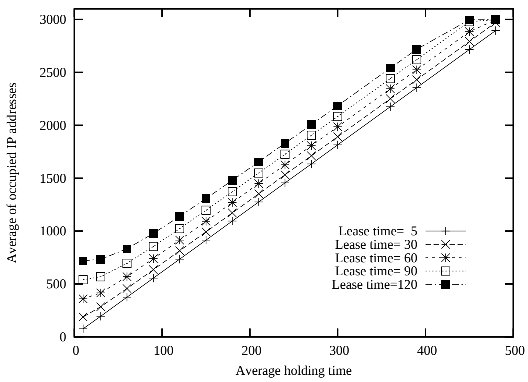
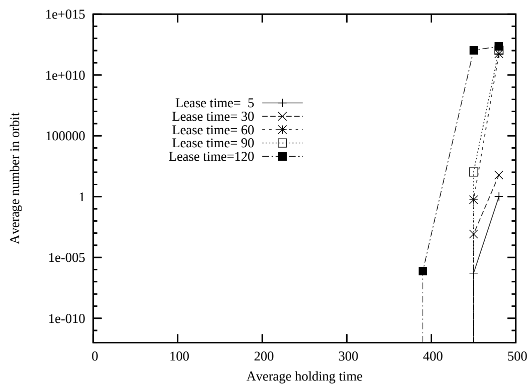


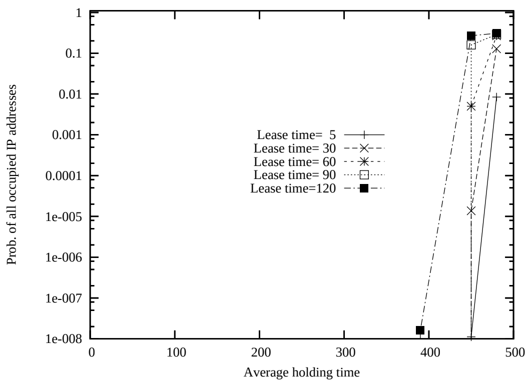
Fig. 5. Renewal rate ($\lambda = 6$ requests/minute)



(a) Occupied IP address



(b) Waiting in the orbit



(c) Probability

Fig. 6. The third scenario ($c = 3000$, $\lambda = 6$ requests/minute)