

Oligopoly Competition in Time-Dependent Pricing for Improving Revenue of Network Service Providers with Complete and Incomplete Information

Cheng ZHANG\(^{(a)}\), Bo GU, Members, Kyoko YAMORI\(^{(1,††)}\), Senior Member, Sugang XU\(^{(†)}\), Member, and Yoshiaki TANAKA\(^{(†,†††)}\), Fellow

SUMMARY Network traffic load usually differs significantly at different times of a day due to users’ different time-preference. Network congestion may happen in traffic peak times. In order to prevent this from happening, network service providers (NSPs) can either over-provision capacity for demand at peak times of the day, or use dynamic time-dependent pricing (TDP) scheme to reduce the demand at traffic peak times. Since over-provisioning network capacity is costly, many researchers have proposed TDP schemes to control congestion as well as to improve the revenue of NSPs. To the best of our knowledge, all the studies on TDP schemes consider only the monopoly or duopoly NSP case. In our previous work, the duopoly NSP case has been studied with the assumption that each NSP has complete information of quality of service (QoS) of the other NSP. In this paper, an oligopoly NSP case is studied. NSPs try to maximize their overall revenue by setting time-dependent price, while users choose NSPs by considering their own time preference, congestion status in the networks and the price set by the NSPs. The interactions among NSPs are modeled as an oligopoly Bertrand game. Firstly, assuming that each NSP has complete information of QoS of all NSPs, a unique Nash equilibrium of the game is established under the assumption that users’ valuation of QoS was uniformly distributed. Secondly, the assumption of complete information of QoS of all NSPs is relaxed, and a learning algorithm is proposed for NSPs to achieve the Nash equilibrium of the game. Analytical and experimental results show that NSPs can benefit from TDP scheme, however, not only the competition effect but also the incomplete information among NSPs causes revenue loss for NSPs under the TDP scheme.

key words: time-dependent pricing, revenue maximization, oligopoly competition, incomplete information game

1. Introduction

Driven by the popularity of smart devices (iPhone, iPad, etc.), bandwidth-hungry applications, cloud-based services, and media-rich web content [1], demand for broadband data is increasing rapidly every year [2], which is forcing Network Service Providers (NSPs) to use pricing as a congestion management tool. This trend is evidenced by the adoption of usage-based data pricing instead of the traditional flat-rate data plan by major wired and wireless NSPs in US, Europe and so on [3]–[6]. Unfortunately, depending on just usage-based pricing fails to prevent congestion at peak times; users must also be given time-dependent incentives to distribute traffic and ease congestion [6].

Previous works have shown that time-dependent pricing (TDP) scheme can give network users the right incentive to shift their traffic demands when the network gets congested [7]–[9]. Ha et al. proposed a TDP scheme for mobile data communication, which gives users monetary reward to delay traffic during traffic peak times [8]. Unlike the real-time feature of the “Smart Market” in [7], time is slotted in [8], such as 48 time slots for one day, 30 minutes per slot. They conducted surveys which reveal that users are indeed willing to wait 5 minutes (for YouTube videos) to 48 hours (for software updates). They concluded that the TDP scheme flattens the temporal fluctuation of traffic usage and benefits both users and NSP. Jiang et al. [9] studied hourly TDP offered by a monopoly selfish NSP, comparing the profit-maximizing time-dependent prices to the socially optimal ones in the case of complete information and incomplete information with users’ utilities. Although the congestion effects were taken into account in [9], the competition between NSPs were not studied.

As written in [6] by Andrew Odlyzko et al., not only congestion, but also competition plays an important part in NSPs’ pricing strategies. Different from the above papers [7]–[9], our previous work [10] studied the duopoly NSP case with the assumption that each NSP has complete information of quality of service (QoS) of both NSPs. The interactions between NSPs were modeled as a duopoly Bertrand game [11]–[13]. Unique Nash equilibrium was established under the assumption that users’ valuation of QoS was uniformly distributed. The simulation results reflected that the revenue from a TDP scheme was higher than that from the time-independent pricing (TIP) scheme in the duopoly case. NSPs could not extract all the surplus from users due to the competition effect. In our previous work [10], only the duopoly NSP case was considered. However, there are usually more than two NSPs in the real world. For example, there are four major wireless NSPs in USA (AT&T, T-Mobile, Verizon and Sprint), and there are three major wireless NSPs in Japan (Docomo, Au and Softbank).

In this paper, we consider the oligopoly NSP case, namely, the case when there are three or more than three NSPs competing with each other for network users to maximize their own revenue. The key idea is to establish an oligopoly Nash equilibrium of the game under the assumption that users’ valuation of QoS is incomplete information game.
Another major assumption made by our previous work [10] is that each NSP has complete information of QoS of both NSPs, on which the establishment of Nash equilibrium was based. However, it is actually very difficult for one NSP to acquire the complete information of QoS of the other NSPs. In this paper, we relax this assumption. We also consider how NSPs adapt their price strategies with only incomplete information of QoS of all other NSPs.

The main contributions of this paper are as follows:

Firstly, the oligopoly NSP case is studied. In each time slot, we model the NSP oligopoly competition as a Bertrand competition (price competition) game, in which three or more than three NSPs set prices to compete for market share (number of users) to maximize their revenue. We assume that each NSP can complete information of QoS of all NSPs, unique Nash equilibrium of the game is established under the assumption that users’ valuation of QoS is uniformly distributed.

Secondly, the assumption that each NSP has complete information with QoS of all NSPs is relaxed, and a learning algorithm is proposed for NSPs to achieve the Nash equilibrium of the game.

The rest of this paper is organized as follows. NSP model and user model are presented in Sect. 2 and Sect. 3, respectively. In Sect. 4, Bertrand competition game with complete information of QoS of all NSPs is formulated, then Nash equilibrium of the game is established for NSPs to choose the time-dependent pricing strategy in each time slot to maximize their overall revenue. In Sect. 5, the case that NSPs only have incomplete information of QoS with each other is investigated, and the learning algorithm is proposed for NSPs to achieve the Nash equilibrium of the game. Numerical results are presented in Sect. 6. Section 7 concludes this paper.

2. NSP Model

Consider a communication market with \( n \) NSPs, denoted by \( S_1, S_2, \ldots, S_n \) respectively. NSP \( S_1, S_2, \ldots, S_n \) provide the substitute network services to network users. There exists a sequence of time, \( t = 1, 2, \ldots, T \), at which NSP \( S_i \), \( i \in \{1, 2, \ldots, n\} \) sets time-dependent price \( p_{it} \) (where \( i \in \{1, 2, \ldots, n\} \)). We denote that \( p^t = (p_{1t}, \ldots, p_{nt})^t \), where \( p^t \) is a \( n \times 1 \) column vector, and “\(^t\)” means transpose operator. It is assumed that the population of users denoted by \( M \) is fixed, with \( M_j \) as the number of users choosing NSP \( S_i \) for \( i \in \{1, 2, \ldots, n\} \) at time slot \( t \). The proportion of users who choose NSP \( S_i \) at time \( t \) is denoted by Eq. (1) as presented in [14].

\[
x_i^t = \frac{M_j}{M}, \quad \text{where} \quad i \in \{1, 2, \ldots, n\}
\]

(1)

It is assumed that the value of \( M \) is very large. We also denote that \( x^t = (x_1^t, x_2^t, \ldots, x_n^t)^t \).

The following set \( D^t \) defined in Eq. (2) is the domain for \( x_1^t, x_2^t, \ldots, x_n^t \).

\[
D^t = \{(x_1^t, x_2^t, \ldots, x_n^t) | \sum_{i=1}^n x_i^t \leq 1, x_i^t \in [0, 1]\}
\]

(2)

The level of QoS provided by NSP \( S_i \) for \( i \in \{1, 2, \ldots, n\} \), denoted as \( q_i \), is assumed to decrease with the number of its subscribers due to congestion. We employ a function \( h_i(\cdot) \) defined on \([0, 1]\) to express the QoS provided by NSP \( S_i \) at time slot \( t \) as \( q_i^t = h_i(x_i^t) \). The following assumption is for the QoS function \( h_i(\cdot) \).

Assumption 1: \( h_i(\cdot) \) is a non-increasing and continuous differentiable positive function of the number of users in network \( S_i \) for \( i \in \{1, 2, \ldots, n\} \). All of the NSPs provide best effort service. Without loss of generality, we assume that \( h_1 > h_2 > \ldots > h_n \).

Remark 1: The assumption of function \( h_i(\cdot) \) captures the congestion effects that users experience when choosing NSP \( S_i \) with limited resources [15]–[17]. Ren et al. adopted the same QoS assumption when considering the QoS formulation of an entrant NSP in a Femtocell communication market [17].

3. User Model

A continuum model of users is employed in this paper. If there are a large number of users in the communication market and each individual user is negligible, the continuum model approximates well the real user population [9], [10]. The payoff of user \( k \) at time \( t \) is denoted as Eq. (3)

\[
u_{ki} = \theta_k q_i^t - p_i^t
\]

(3)

where \( \theta_k \in [0, \varphi] \) is the QoS valuation of user \( k \) at time slot \( t \). Please note that \( \theta_k \) is time-dependent, reflecting users’ different preferences for different time slots [9], [10]. The value of \( \theta_k \) is private information of users, but the distribution of \( \theta_k \) is public information to NSPs. Furthermore, different users may have different valuations on the same level of QoS. \( q_i^t \) denotes the QoS provided by NSP \( S_i \)’s network, and \( \theta_k q_i^t \) denotes the benefit that the user can get from NSP \( S_i \). The unit of users’ valuation of QoS (i.e., \( \theta_k \)) is chosen so that \( \theta_k q_i^t \) has the same unit with that of the payment \( p_i^t \), which is the price set by NSP \( S_i \) at each time slot. Each NSP knows the level of QoS of its own network all the time. Therefore, the NSP can broadcast the information of QoS and price to users in its communication range at each time slot as described in
At each decision-making time \( t \) and learning techniques. The details of information acquisition is set as zero to simplify the analysis. NSPs need to know the distribution of users’ valuations of QoS by conducting market surveys and using data mining techniques. The details of information acquisition are beyond the scope of this paper.

At each decision-making time \( t \), each user only chooses one NSP’s network. Each user is a rational decision maker, which means that each user tries to maximize his payoff by choosing a proper NSP. When a user switches from one NSP to another one, the user may face switching cost. Although we will make an assumption in Assumption 3 that the switching cost for each user is zero, we discuss the cost involved when users change their NSPs. The switching cost across different NSPs may be different. For example, the cost for users to switch from NSP \( S_1 \) to NSP \( S_2 \) may be different as it would to switch from \( S_1 \) to \( S_2 \). For simplicity, we assume that the switching costs across different NSPs are the same and denote this cost as \( c_s \). Therefore, when a user switches to another network, he has to consider new payoff as in Eq. (4).

\[
u^t_k = \theta^t_k q^t_k - p^t_k - c_\text{s} \tag{4}\]

If we denote \((p^t_k + c_\text{s})\) as \( p^t_k' \), the new payoff in Eq. (4) can be expressed as in Eq. (5)

\[
u^t_k = \theta^t_k q^t_k' - p^t_k' \tag{5}\]

The new payoff in Eq. (5) exactly has the same structure as the payoff in Eq. (3). Therefore, most of the analytical results without considering switching cost are also variable when switching cost is incorporated. We have rigorously analyzed switching cost under the duopoly case in our previous work [19]. While rigorous analysis of \( c_s \) under the oligopoly case is left as our future work, we show in Sect. 6 the impact of switching cost \( c_s \) on NSPs’ revenue.

Assumption 3: There is no switching cost when users change from one NSP to another. At each decision time \( t \), each user makes decision independently.

Remark 3: Although it is technically feasible for users to change their NSPs at anytime, users always face switching cost when they change their NSPs in real world.

The notations used throughout this paper are summarized as shown in Table 1.

### 4. Revenue Maximization with Complete Information

NSPs try to maximize their overall revenue by maximizing their revenue in each time slot. We model the NSP oligopoly competition as a Bertrand competition game for each time slot. In the Bertrand competition game, different firms strategically choose prices independently at the same time while supplying quantities demanded at the chosen prices [11]–[13]. Unique Nash equilibrium of the Bertrand competition game is established under the assumption that users’ valuation of QoS is uniformly distributed. The revenue of NSPs depends on the number of users in the network and the prices set by NSPs, which is denoted as \( R^t = p^t x^t(p^t) \). \( x^t(p^t) \) is the proportion of users who choose NSP \( S_i \) at time \( t \) as defined in Eq. (1). Please note that it is a function of the prices set by all the network \( S_1, S_2, \ldots, S_n \), which can be calculated by Lemma 1 established in this section. The overall revenue of each NSP can be expressed as \( \sum_{i=1}^{n} R^t_i \).

**Assumption 4:** The level of QoS at the beginning of time slot \( t \) is estimated from the number of users in the end of time slot \( t-1 \).

**Remark 4:** Since we have assumed all NSPs provide best effort service and the number of users changes with time, the level of QoS also changes with time. It is very complicated to consider the real-time QoS, and this assumption is based on “if there are a large number of users in the last time slot, and the level of QoS is low, then expectation of QoS in the current time slot is also low”. The level of QoS at the beginning of time slot \( t \) can be expressed as \( h_i(x^{t-1}) \). This kind of estimation is valid if there are a large number of users in

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>( t \in {1, \ldots, n} ), which is NSP set</td>
</tr>
<tr>
<td>( k )</td>
<td>subscript of a user</td>
</tr>
<tr>
<td>( S_i )</td>
<td>the ( n ) oligopoly NSPs in the market, for ( i \in {1, \ldots, n} )</td>
</tr>
<tr>
<td>( T )</td>
<td>the total time slots</td>
</tr>
<tr>
<td>( t )</td>
<td>( t \in {1, T} ), the ( t )-th time slot</td>
</tr>
<tr>
<td>( M )</td>
<td>the population of users</td>
</tr>
<tr>
<td>( M_i )</td>
<td>the number of users choose NSP ( S_i ), for ( i \in {1, \ldots, n} ) at time ( t )</td>
</tr>
<tr>
<td>( x^t_i )</td>
<td>( x^t_i = M_i^t/M ): the proportion of users who choose NSP ( S_i ) at time ( t )</td>
</tr>
<tr>
<td>( x^t )</td>
<td>( x^t = (x^t_1, x^t_2, \ldots, x^t_n) ) is a ( n \times 1 ) column vector.</td>
</tr>
<tr>
<td>( u^t_k )</td>
<td>the payoff of user ( k ) in network ( S_i ) at time slot ( t )</td>
</tr>
<tr>
<td>( h_i(x^{t-1}) )</td>
<td>the QoS function of NSP ( S_i ) at the beginning of time slot ( t )</td>
</tr>
<tr>
<td>( q^t )</td>
<td>( q^t = h_i(x^{t-1}) )</td>
</tr>
<tr>
<td>( p^t )</td>
<td>the price set by NSP ( S_i ), at time slot ( t )</td>
</tr>
<tr>
<td>( p_i^t )</td>
<td>( p_i^t = (p_1^t, \ldots, p_n^t) ) is a ( n \times 1 ) column vector.</td>
</tr>
<tr>
<td>( \theta^t_k )</td>
<td>user ( k )'s valuation of QoS at time slot ( t )</td>
</tr>
<tr>
<td>( f^t(\cdot) )</td>
<td>probability density function (PDF) of users’ valuation of QoS at time slot ( t )</td>
</tr>
<tr>
<td>( F^t(\cdot) )</td>
<td>cumulative density function (CDF) of users’ valuation of QoS at time slot ( t )</td>
</tr>
<tr>
<td>( \varphi' )</td>
<td>the upper bound of the domain of the function ( f^t(\cdot) )</td>
</tr>
<tr>
<td>( r_i^t )</td>
<td>the marginal point where users will switch</td>
</tr>
<tr>
<td>( R_i^t )</td>
<td>the revenue get by NSP ( S_i ) at time slot ( t )</td>
</tr>
<tr>
<td>( p_{ij} )</td>
<td>NSP ( S_j )'s ( j )-th price strategy, ( i \in {1, \ldots, n}, j \in {1, \ldots, m_i} )</td>
</tr>
<tr>
<td>( a_i(t) )</td>
<td>NSP ( S_i )'s action at time slot ( t )</td>
</tr>
<tr>
<td>( \delta_{ij}(t) )</td>
<td>the probability of NSP ( S_i )'s strategy ( p_{ij} ) at time ( t )</td>
</tr>
<tr>
<td>( m_0 )</td>
<td>the number of NSP ( S_i )'s price strategy</td>
</tr>
<tr>
<td>( m_i^t )</td>
<td>( [\delta_{ij}(t), \ldots, \delta_{ij}(t)] ), NSP ( S_i )'s probability vector at time ( t )</td>
</tr>
<tr>
<td>( q_i^t )</td>
<td>the expected payoff of NSP ( S_i ) at time ( t )</td>
</tr>
</tbody>
</table>
the network, the marginal effect of a single user’s impact on QoS of the whole network can be neglected. But the accumulation effect of many users’ impact on QoS cannot be neglected. This assumption is the same as that in our previous work [10].

At time slot \( t \), user \( k \) chooses NSP that gives him/her highest payoff. The index of NSP that user \( k \) chooses can be expressed as shown in Eq. (6)

\[
l = \arg \max_{i=1,...,n} u'_{ik}
\]

(6)

Now we characterize the marginal points that identify users’ valuation of QoS associated with changes in their decision to choose NSP. We have the following Definition 1.

**Definition 1:** Define \( n \) marginal points \( r_k^1, r_k^2, ..., r_k^n \), as

\[
r_k^i = \frac{p_i - p_{i+1}}{q_i - q_{i+1}} \quad \text{if } i = 1, 2, ..., n - 1.
\]

(7)

\[
r_k^n = \frac{p_i - p_1}{q_i - q_1}
\]

(8)

**Remark 5:** We can derive these marginal points by letting \( u'_{ik} = u'_{ik+1} \) for \( i = 1, 2, ..., n - 1 \) and \( u_{ik,n} = 0 \). Geometrically, for \( i=1,2,...,n-1 \), \( r_k^i \) is the horizontal axis value of the intersection point between line \( u'_{ik} \) and \( u'_{ik+1} \). \( r_k^n \) is the horizontal axis value of the intersection point between horizontal axis and line \( u_{kn} \).

In order to illustrate the marginal points in Definition 1, we assume that there are three NSPs, which means that \( n = 3 \). User’s payoff function \( u'_{ik} = \theta_i' q_i - p_i' \) is a linear function of user’s QoS valuation \( \theta_i' \), and the slope of the payoff function is \( q_i' \). By Assumption 1, we have \( q_1' > q_2' > q_3' \). The user’s payoff function lines can be drawn in the same coordinate system with \( \theta_i' \) as the horizontal axis and user’s payoff as the vertical axis (see Fig. 2). It is shown that the prices set by NSPs are the \( y \)-intercepts in the coordinate system. The marginal point \( r_k^1 \) can be derived by letting \( u'_{ik,1} = u'_{ik,2} \), which is \( r_k^1 = \frac{p_i - p_1}{q_i - q_1} \), and the payoff lines \( u'_{ik,1} \) and \( u'_{ik,2} \) intersect at point \( E \). Similarly, the marginal point \( r_k^2 \) can be derived by letting \( u'_{ik,2} = u'_{ik,3} \), which is \( r_k^2 = \frac{p_i - p_2}{q_i - q_2} \), and payoff lines \( u'_{ik,2} \) and \( u'_{ik,3} \) intersect at point \( C \). The marginal point \( r_k^3 \) can be derived by letting \( u'_{ik,3} = 0 \), which is \( r_k^3 = \frac{p_i}{q_i} \).

The payoff line \( r_k^3 \) and horizontal axis intersect at point \( A \).

**Lemma 1:** The necessary and sufficient condition for positive number of users in each NSP’s network is shown in Eq. (9).

\[
\varphi' > r_k^1 > r_k^2 > r_k^3 > ... > r_k^n
\]

(9)

The number of users in each NSP’s network can be expressed as shown in Eq. (10).

\[
x_i^k = \begin{cases} F((r_k^1) - F(r_k^1)) & \text{if } i = 2, ..., n \\ 1 - F(r_k^1) & \text{otherwise} \end{cases}
\]

(10)

**Proof.** (1) **Sufficiency:** If \( \varphi' > r_k^1 > r_k^2 > ... > r_k^n \), then the users with valuation of QoS belonging to \( [r_k^1, r_k^n] \) will choose NSP \( S_i \)'s network at price \( p_i' \) \((\text{This is obvious by Definition 1, we will further illustrate this point by example after the proof of this lemma.})\). The number of users in NSP \( S_i \)'s network is given by

\[
x_i^k = \begin{cases} \int_{r_k^1}^{r_k^1} f(y)dy = F((r_k^1) - F(r_k^1) & \text{if } i = 2, ..., n \\ \int_{r_k^1}^{\varphi'} f(y)dy = 1 - F(r_k^1) & \text{otherwise} \end{cases}
\]

(11)

According to Assumption 2, it is obvious that \( x_i^k > 0 \), which means that the number of users in NSP \( S_i \)'s network is positive.

(2) **Necessity:** If the number of users in each network is positive, or \( x_i^k > 0 \). By Eq. (7), if \( i = 2, ..., n \), we have \( F((r_k^i - F(r_k^i)) > 0, \text{ or } F((r_k^i) > F(r_k^i)) \). Since \( F(\cdot) \) is an increasing function, we have the following Eq. (12)

\[
r_k^i - F(r_k^i) > r_k^i
\]

(12)

If \( i = 1 \), we have \( 1 - F(r_k^1) > 0 \). Since \( 1 = F(\varphi') \), we have \( F(\varphi') - F(r_k^1) > 0 \). Similarly, we have the following Eq. (13).

\[
\varphi' > r_k^1
\]

(13)

By combining Eq. (12) and Eq. (13), we can check that the condition in Eq. (9) is right.

**Q.E.D**

Although users’ QoS valuation is a random variable with CDF \( F(\cdot) \), the proportion of users \( x_i^k \) calculated from \( F(\cdot) \) is not a random variable. Once the \( F(\cdot) \) and the marginal points determined, the value of \( x_i^k \) is determined from Eq. (10). In order to illustrate that a user with valuation of QoS between \( [r_k^1, r_k^1] \) will choose NSP \( S_i \)'s network at price \( p_i' \) in the proof process of Lemma 1, we take \( n = 3 \) as an example. In Fig. 2, if the user’s QoS valuation is between \( [r_k^1, r_k^2] \), the line \( u_{ik,3} \) is above lines \( u_{ik,1} \) and \( u_{ik,2} \), this means that the user can obtain maximum payoff with NSP \( S_3 \) in
this case, then the user will choose NSP S3. Similarly, if the user’s QoS valuation is between \([t^2_k, t^3_k]\), the line \(u^2_k\) is above lines \(u^1_k\) and \(u^3_k\), then the user will choose NSP S2; if the user’s QoS valuation is between \([t^1_k, \varphi'']\), the line \(u^1_k\) is above lines \(u^2_k\) and \(u^3_k\), then the user will choose NSP S1.

Each NSP tries to maximize its overall revenue by considering the following subproblem shown in Eq. (14) in each time slot \(t\):

\[
\max_{p_i} R^i_t
\]

subject to \(x_i' \in D^i\)

The above problem can be solved by considering the game played by NSP S1, S2, ..., Sn. The Nash Equilibrium point is the solution of the problems. Now we consider that n NSPs play a Bertrand competition (or price competition) game in each time slot \(t\). The Bertrand competition game \(\Gamma(\text{Player, Strategy, Payoff})\), is described as follows:

- **Player**: NSP S1, ..., and Sn are the \(n\) players in the game.
- **Strategy**: The strategy is the price set by NSP \(S_i\) for \(i \in \{1, 2, ..., n\}\).
- **Payoff**: The payoff is the revenue gotten by NSP \(S_i\) for \(i \in \{1, 2, ..., n\}\).

In this game, NSP S1, S2, ..., Sn set their prices \(p^i_1, p^i_2, ..., p^i_n\) respectively to maximize their revenue, which is the multiplication of price and the market share (or the number of users). The number of users in each NSP’s network can be derived by Lemma 1.

In order to derive the number of users in each time slot, NSPs should know the QoS function of each other. The QoS function is predetermined by the technology and capacity investment. For example, the QoS functions of 3G network and 4G network are different. The QoS function of each NSP \((h_i(\cdot))\) is the same for all the time slot. We assumed that each NSP observes the QoS function of all other NSPs by investigating the technology and capacity before the game of the first time slot. Before playing the game in each time slot, we use the number of users in the last time slot \((x_i^{t-1})\) to estimate the level of QoS in the current time slot as \(h(x_i^{t-1})\).

**Proposition 1**: If users’ QoS valuation is distributed uniformly, the necessary and sufficient condition for unique Nash Equilibrium of the game \(\Gamma(\text{Player, Strategy, Payoff})\) is as that in Eq. (15)

\[
|I - A| \neq 0
\]

where

\[
A = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & ... & 0 \\
0 & 0 & \beta'_3 & 0 & 0 & ... & 0 \\
0 & \alpha'_2 & 0 & \beta'_4 & 0 & ... & 0 \\
0 & 0 & \alpha'_3 & 0 & \beta'_5 & ... & 0 \\
... & ... & ... & ... & ... & ... & ... \\
0 & 0 & 0 & 0 & \alpha'_n & 0 & \beta'_n \\
0 & 0 & 0 & 0 & 0 & \alpha'_n & 0
\end{pmatrix}
\]

\[
\alpha'_i = \frac{\varphi'(q^i_t - q^i)}{2} \quad \text{if} \quad i = 1
\]

\[
\beta'_i = \frac{q^i_{t-1} - q^i_t}{2(q^i_{t-1} - q^i_{t+1})} \quad \text{else if} \quad i = 2, ..., n - 1
\]

\[
\beta'_n = \frac{p^i_t}{2(q^i_{t-1} - q^i_{t+1})} \quad \text{for} \quad i = 2, ..., n - 1
\]

and I is an \(n \times n\) unit matrix with ones on the main diagonal and zeros elsewhere.

And the Nash Equilibrium price is

\[
p^*_i = \frac{|(I-A_i)|}{|I-A|}
\]

where \((I-A_i)\) is the matrix formed by replacing the \(i\)th column of \((I-A)\) by the column vector \(\mu'\). \(\mu'\) is defined as \(\mu' = (\alpha'_0, 0, ..., 0)'\). The operator \(|\cdot|\) on a matrix denotes the determinant [20] of the matrix. Thus, \((I-A)\), \((I-A_i)\) is determinant of the matrix \((I-A)\).

**Proof**: Please refer to Appendix A for the details of proof.

The Proposition 1 has generalized the Nash equilibrium result in [10] to the oligopoly NSP case. It can be easily verified that the Nash equilibrium result of [10] is just the special case of Proposition 1 with \(n = 2\).

**Remark 6**: Please note that in Definition 1 and Proposition 1, one implicit assumption is that \(q^1_t > q^2_t > ... > q^n_t\). When we make assumption that \(h_1 > h_2 > ... > h_n\) in **Assumption 1**, it is not necessarily that \(q^1_t > q^2_t > ... > q^n_t\) where \(q^i_t = h_i(x_i^{t-1})\) is the level of QoS at the beginning of time slot \(t\), since \(q^i_t\) is also dependent on the number of users in the network. However, as the game between NSPs is played in each time slot, only the information of \(q^1_t, q^2_t, ..., q^n_t\) is necessary at the beginning of each time slot. Even if \(q^1_t, q^2_t, ..., q^n_t\) may not be in the order of \(q^1_t > q^2_t > ... > q^n_t\), for a time slot, NSPs could be re-indexed so that \(q^1_t' > q^2_t' > ... > q^n_t'\), where \(1', 2', ..., n'\) are the new indexes of NSPs. Even though the indexes of NSPs may be changed in each time slot, the NSP related information, such as price, revenue, etc., are kept for each NSP over all the time slots.

### 5. Revenue Maximization with Incomplete Information

The unique Nash equilibrium has been established in Sect. 4. NSPs can maximize their revenue in each time slot by setting the Nash equilibrium price. One important assumption for deriving the Nash equilibrium in Sect. 4 is that each NSP knows the QoS function of all other NSPs before the Bertrand game. However, the information of QoS function cannot be easily gotten by each NSP beforehand. Therefore, one key question may be asked is as follows: How do NSPs set their price strategies to maximize their revenue when they do not know the complete information of all other NSPs?

A team of learning automata is proposed to evolve to Nash equilibrium, when NSPs do not have information of
QoS function of other NSPs. A learning automaton is a simple adaptive decision making device that is capable of learning the optimal price strategies of NSPs through interactions with an unknown environment [21]. The automaton keeps a probability distribution over the set of action (price strategies) and at each time instant it selects one action at random based on this distribution. The revenue gotten by NSPs is viewed as reaction for automaton. Then the automaton uses this reaction to update its action probability distribution through a learning algorithm and the cycle repeats until convergence to the Nash equilibrium. Actually, the learning automaton has been successfully used in wireless networks for power control [22], [23]. It is shown that the learning automaton is simple, efficient, and never converges to a point which is not Nash equilibrium (Theorem 1 in [23]).

We assume that each NSP has a finite set of price levels. We define $p_{ij}$, $i \in \{1, 2, ..., n\}$ and $j \in \{1, 2, ..., m_i\}$, to be the $m_i$ candidate price levels for NSP $S_i$. The NSP’s pricing strategy is defined over a probability vector $\delta(t) = [\delta_{i1}(t), ..., \delta_{im}(t)]$, where NSP $S_i$ chooses price level $p_{ij}$ with probability $\delta_{ij}(t)$. The price strategy chosen by NSP $S_i$ each time is called action, which is denoted as $a_i(t)$.

Then we can define the expected payoff for NSP $S_i$ as $g'_i$ given by

$$g'_i(\delta(t), ..., \delta_n(t)) = \sum_{j_1, ..., j_n} d'_i(j_1, ..., j_n) \prod_{l=1}^N \delta_{ij_l}$$

Where

$$d'_i(j_1, ..., j_n) = E[R_i|\text{player } i \text{ choose action } j_l, 1 \leq l \leq n]$$

**Definition 2:** The N-tuple of strategies $(\delta_1^*, ..., \delta_n^*)$ is said to be a Nash equilibrium, if for each $i$, $1 \leq i \leq n$, we have

$$g'_i(\delta_1^*, ..., \delta_{i-1}^*, \delta_{i+1}^*, ..., \delta_n^*) \geq g'_i(\delta_1, ..., \delta_{i-1}, \delta_i, \delta_{i+1}, ..., \delta_n)$$

$\forall$ probability vector $\delta_i \in [0, 1]^{m_i}$.

**Proposition 2:** There exists a mixed strategy Nash equilibrium for the game $\Gamma$(Player, Strategy, Payoff) with incomplete information.

**Proof.** It is obvious that the game $\Gamma$(Player, Strategy, Payoff) defined is a finite strategic-form game. According to Theorem 1.1 in [12], it is well known that every finite strategic-form game has a mixed strategy equilibrium.

Q.E.D.

The proposed learning algorithm for NSPs to choose price strategy is shown in Fig. 3. The update rule defined in Eq. (21) is known as linear reward-inaction (LR) rule [21]. The basic idea behind the update rule is rather simple [24]. If NSP $S_i$ selects price $p_{ij}$, the probability for price $p_{ij}$ is increased, while probabilities for other price strategies are decreased. For the probability vector of NSP $S_i$, $\delta_i(t)$ converges, stop. Otherwise, go to step 2.

**Step 1:** Set the initial probability vector $\delta_i(0)$.

**Step 2:** At every time instant $t$, each NSP chooses a price according to its probability vector $\delta_i(t)$. Thus, NSP $S_i$ chooses action $a_i(t)$ at time $t$, based on the probability distribution $\delta_i(t)$.

**Step 3:** Each NSP obtains payoff based on the set of all actions. The payoff to NSP $S_i$ is $R_i(t)$, which is normalized.

**Step 4:** Each NSP $S_i$ updates its action probability according to the rule:

$$\begin{align*}
\delta_{ij}(t+1) &= \delta_{ij}(t) - \gamma R_i(t) \delta_{ij}(t) \quad a_i(t) = p_{ij}, \\
\delta_{ij}(t+1) &= \delta_{ij}(t) + \gamma R_i(t) \sum_{s \neq i} \delta_{js}(t) \quad a_i(t) \neq p_{ij}, \\
i &= 1, ..., n, j = 1, ..., m_i.
\end{align*}$$

(21) where $0 < \gamma < 1$ is the step size.

**Step 5:** If $\delta_i(t)$ converges, stop. Otherwise, go to step 2.

Fig. 3 Learning algorithm for NSP to choose price strategy.

$\rho_{s, i} \neq j$, are decreased. For the probability vector of each NSP, $\delta_i(t) = [\delta_{i1}(t), ..., \delta_{im}(t)]$, two properties are satisfied: (1) $\sum_{j=1}^{m_i} \delta_{ij}(t) = 1 \forall t$; (2) $\delta_{ij}(t) \geq 0 \forall t$. Please refer to Appendix B for the proof of these two properties. Let $\phi(t) = (\delta_{i1}(t), ..., \delta_{in}(t))$ denote the state of the price strategies of all NSPs at time slot $t$. Under the learning algorithm in Fig. 3, $\phi(t), t \geq 1$ is a Markov process. The $L_{R-1}$ update rule is known to be $\epsilon$-optimal ($\epsilon > 0$), i.e., upon convergence, the price strategies generated by this scheme will produce a value for the revenue of NSPs that is within $\epsilon$ of the optimal value.

6. Simulation

This section makes simulation analysis to validate our analytical results. We assume that $n = 3$, namely, there are three NSPs, $S_1$, $S_2$ and $S_3$, compete with each other for network users to maximize their revenue. The simulation analysis includes the following aspects:

- Compare the revenue from TDP scheme with the revenue from Time-Independent Pricing (TIP) scheme for the oligopoly NSP case when each NSP has complete information (CP) of QoS function of all NSPs.
- Compare the revenue from TDP scheme with the revenue from TIP scheme for the oligopoly NSP case when each NSP has incomplete information (ICP) of QoS function of all NSPs.
- Compare the total revenue gotten by NSP $S_1$, $S_2$ and $S_3$ with that from duopoly NSP case when each NSP has CP of QoS function of all NSPs.
- Compare the revenue from TDP scheme when each NSP has CP of QoS function of all NSPs with the revenue from TIP scheme when NSPs have ICP.

Please refer to [10] for the analysis of duopoly NSP case.

The parameters for simulations are summarized in Table 2. We assume that the distribution of users’ valuation of QoS $\theta_i$ follows an uniform distribution $[0, \varphi']$. Time of a day is divided into 24 slots. When $t \in [1, 8]$ or $t \in [17, 24]$, $\varphi' = 2$, representing off-peak slots, and when $t \in [9, 16]$, $\varphi' = 3$, representing peak slots.
\( \varphi' = 4 \), representing peak slots. Therefore, users in average have much higher valuation of QoS during time slots \([9, 16]\) than that during other time slots. It can be expected that the peak traffic will occur during time slots \([9, 16]\). The QoS function is defined as simple affine function satisfying Assumption 1 aforementioned. This kind of affine QoS function has been also adopted in \([16, 17]\). The three NSPs set the prices in each time slot according to the NE established in Proposition 1 when each NSP has complete information of QoS function of all NSPs. When NSPs have incomplete information with other NSPs’ QoS function, the proposed learning algorithm in Fig. 3 of Sect. 5 will be used to determine the price of each NSP. In our simulation, when NSPs apply TIP scheme, they set their TIP prices strategically by considering other NSPs’ pricing strategies in the initial time slot (the optimal prices in the initial time slot, please see TIPs in Table 2) and keep the prices unchanged over all the time slots. We also investigated other two TIP prices, which are denoted as TIPb and TIPc in Table 2.

**Observation 1:** The revenue from TDP scheme is higher than that from TIP scheme in the oligopoly competition environment when each NSP has complete information of QoS function of all NSPs.

We investigated three TIP prices, which are denoted as TIPa, TIPb, TIPc (Please see parameters in Table 2). It is shown in Fig. 4 that the revenue from TIPa, TIPb and TIPc is lower than that from TDP scheme. In TDP scheme, NSPs can set new prices for each time slot, which is the Nash Equilibrium of the game \( \Gamma \), thus the revenue of NSPs gets maximized at each time slot. However, in TIP scheme, the price can only be set initially without considering the competition from the rival NSPs and the congestion effect. The percentages of revenue improvement are also shown in Fig. 4(a), Fig. 4(b) and Fig. 4(c). The percentages of revenue improvement from TIPs are also shown in the figures. It is shown in Fig. 7(a) that the average percentage of total revenue improvement is 21.62%.

**Observation 2:** The revenue from TDP scheme is higher than that from TIP scheme in the oligopoly competition environment when each NSP has incomplete information of QoS function of all NSPs.

Similar to Observation 1, the revenue from TDP scheme is much higher than that of TIP scheme. Even under incomplete information case, NSPs use TDP scheme dynamically to change their price strategy to achieve much higher revenue. Please see Fig. 5(a), Fig. 5(b), Fig. 5(c) and Fig. 9. The percentages of revenue improvement are also shown on the figures. It is shown in Fig. 9 that the average percentage of total revenue improvement is 16.14%.

**Observation 3:** Under TDP scheme, the revenue of NSP is lower when NSP has incomplete information than that when NSP has complete information of QoS function of other NSPs.

We can see from Fig. 6(a) that the revenue of NSP \( S_1 \) is much lower when NSP has incomplete information than that when NSP has complete information of QoS function of other NSPs. And with times going by, the revenue difference between complete information and incomplete information becomes smaller and smaller. The reason is that the

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Parameters in the simulation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ( k )'s valuation of QoS at time slot ( t ): ( \theta_k^t )</td>
<td>( (1) ) when ( r \in {1, 8} ) or ( r \in {17, 24} ), ( \theta_k^t ) is from uniform distribution defined on ( [0, 2] ); ( (2) ) when ( r \in {9, 16} ), ( \theta_k^t ) is from uniform distribution defined on ( [0, 4] ), ( f(\theta_k^t) = \frac{1}{4}, F(\theta_k^t) = \frac{\theta_k^t}{2}, \theta_k^t \in [0, 2] ); ( f(\theta_k^t) = \frac{1}{4}, F(\theta_k^t) = \frac{\theta_k^t}{2}, \theta_k^t \in [0, 4] )</td>
</tr>
<tr>
<td>QoS function of NSP ( S_1, S_2 ) and ( S_3 )</td>
<td>( h_1(x') := 0.9 - 0.1 \times x'; h_2(x') := 0.8 - 0.15 \times x'; h_3(x') := 0.65 - 0.2 \times x'; )</td>
</tr>
<tr>
<td>The NSP ( S_k )'s price at time slot ( t ): ( p_k^t )</td>
<td>In the case of CP, set the price as the NE point as Eq. (18) in Proposition 1; in the case of ICP, set the price according to algorithm in Fig. 3.</td>
</tr>
<tr>
<td>Three kinds of TIP prices for NSP ( S_1, S_2 ) and ( S_3 )</td>
<td>(a) TIPbs1 = 0.017, TIPbs2 = 0.588, TIPbs3 = 0.432; (b) TIPbs1 = 0.650, TIPbs2 = 0.620, TIPbs3 = 0.460; (c) TIPcs1 = 0.6, TIPcs2 = 0.530, TIPcs3 = 0.400;</td>
</tr>
<tr>
<td>Number of time slots</td>
<td>24 time slots per day</td>
</tr>
</tbody>
</table>

(a) Revenue of NSP \( S_1 \) under the TIPa, TIPb, TIPc and TDP schemes.  
(b) Revenue of NSP \( S_2 \) under the TIPa, TIPb, TIPc and TDP schemes.  
(c) Revenue of NSP \( S_3 \) under the TIPa, TIPb, TIPc and TDP schemes.

Fig. 4 Revenue of NSP \( S_1, S_2 \) and \( S_3 \) under TIPa, TIPb, TIPc and TDP schemes with each NSP has CP of QoS function of all NSPs.
ability of price strategy learning gives NSPs the power to improve their revenue when the information is incomplete. However, the incomplete information do cause revenue loss for NSPs. The trends of NSP $S_2$ and $S_3$ is similar to NSP $S_1$ as shown in Fig. 6(b) and Fig. 6(c), so it is the total revenue of all NSPs as shown in Fig. 7(b). The percentages of revenue loss due to incomplete information are also shown in the figures. It is shown in Fig. 7(b) that the average percentage of total revenue loss is $28.80\%$.

**Observation 4:** The revenue from TDP scheme in the oligopoly NSP case is lower than that in a duopoly NSP case when each NSP has complete information of QoS functions of all NSPs.

In oligopoly NSP case, the competitions between NSPs are much fiercer than that of duopoly NSP case, and the social surplus goes much more to users in oligopoly case. Thus the revenue from TDP scheme in the oligopoly NSP case
is lower than that in a duopoly NSP case. Please refer to Fig. 7(c). The percentages of revenue loss due to oligopoly competition are also shown in Fig. 7(c). It is shown that the average percentage of revenue loss is 10.10%. This observation is reminiscent of an observation of [10], in which the revenue from TDP scheme in the duopoly NSP case is lower than that in a monopoly NSP case. It can be concluded that users benefit from competitions between NSPs, and the more competition, the more benefit for network users.

**Observation 5:** The price and revenue of NSPs have a positively correlation with NSPs’ QoS level.

We have shown the price of NSP S1, S2 and S3 under TDP scheme in Fig. 8. We can see that $p_1 > p_2 > p_3$ for all time slots. Since the level of QoS of NSP S1 is greater than that of NSP S2, and level of QoS of NSP S2 is greater than that of NSP S1 by Assumption 1, this result is consistent with the result in duopoly case that the prices of NSPs are upper-bounded by their QoS functions [10]. Similarly, we can also see from Fig. 5 that the higher the level of QoS provided by NSP, the higher the revenue of NSP. Our previous work [10] have analytically shown similar result in a duopoly case.

**Observation 6:** TDP scheme has congestion control effect in the oligopoly case.

According to Fig. 10, the total number of users under TIP scheme is much more than that under TDP scheme in “peak hours”. The reason is that when price is time-dependent, NSPs can adjust their price accordingly to make less users use their network. However, in “non-peak hours”, the total number of users under TIP scheme is much less than that under TDP scheme. During all time slots, the total number of users under TDP scheme is much more than that of TIP scheme. This result is the same as that in [8], where Ha et al. observed a 130% increase in usage from TIP to TDP.

We do not show numerical result of TDP scheme’s effect on the level of NSPs’ QoS during “peak hours”, but we can infer that the level of NSPs’ QoS increases during “peak hours” since the QoS function decreases with the number of users, and the number of users is much less under TDP scheme than that of TIP scheme during “peak hours”.

**Observation 7:** Total revenue of NSPs under TDP scheme decreases with switching cost.

As is shown in Fig. 11, the total revenue of NSP S1, S2 and S3 under TDP scheme over 24 hours decreases with switching cost. As shown in Eq. (4) before Assumption 3, switching cost decreases users’ payoff, which makes users keep staying in their current network. The revenue improvement by TDP scheme is limited since users can not change their network freely.

7. Conclusion

This paper analyzed the time-dependent pricing scheme in an oligopoly competition environment. We model the NSP oligopoly competition as a Betrand competition (price competition) game, in which each NSP sets time-dependent price to compete for market share (number of users) to maximize its revenue. When each NSP has complete information of QoS function of all NSPs, unique Nash equilibrium is established under the assumption that users’ valuation of QoS is uniformly distributed. When each NSP has incomplete information of QoS function of all NSPs, a learning
algorithm is proposed for NSPs to achieve the Nash equilibrium of the game. The simulation results show that the revenue from TDP scheme is higher than that from TIP scheme when NSPs have complete information of QoS function of all NSPs in the oligopoly case. The revenue of NSP is lower when each NSP only has incomplete information of QoS function than that of when each NSP has complete information. And the total revenue of all NSPs in oligopoly case is lower than that of duopoly case. It can be concluded that not only the competition effect but also the incomplete information among NSPs causes revenue loss for NSPs under TDP scheme.

Another interesting alternative is the asymmetrical case wherein some NSPs use TDP scheme, and other NSPs use TIP scheme. The Bertrand game is no longer suitable for modeling this asymmetrical case. However, some hints can be gained from the Bertrand game. For example, NSPs with high level of QoS and relative low price has competitive advantage. For rigorous analysis of the asymmetrical case, a multiple leader multiple follower Stackelberg game can be used to model the asymmetrical case. The multiple leaders of Stackelberg game are the NSPs with TIP scheme, while the NSPs with TDP scheme are the followers. In the first stage of the Stackelberg game, the NSPs with TIP scheme set a fixed price, then in the second stage, the NSPs with TDP scheme can observe the leaders’ price strategies and set time-dependent price to maximize their own revenue. Backward induction can be used to solve the NE of the Stackelberg game in which the second stage problem of the game is solved first, given the fixed price of leaders. Then the first stage problem of the Stackelberg game can be solved. The specific analysis of the Stackelberg game model is left as future work. This work assumes that users’ switching cost is zero, but non-zero switching cost can cause a “lock-in” effect for users. While rigorous analysis of switching cost under the oligopoly case is left as our future work, we have shown in Sect. 6 that the revenue improvement from the TDP scheme may also be impacted by switching cost.

Acknowledgements

The authors express sincere gratitude to associate Professor Noriaki Kamiyama at Graduate School of Information Science and Technology, Osaka University for constructive suggestions, and associate Professor Min Ouyang at School of Automation, Huazhong University of Science and Technology and Miss Lijuan Liu at Sumitomo Chemical Company, Limited for proofreading.

References

Appendix A: Proof of Proposition 1

We denote the price of $n$ NSPs as a vector $p' = (p'_1, ..., p'_n)'$, and the price vector $p''_i = (p''_1, ..., p''_{i-1}, p''_{i+1}, p''_{n})'$. Since users’ QoS valuation is distributed uniformly, we have $F^*(p') = \frac{\partial R'}{\partial p_1} \cdot \frac{\partial p_1}{\partial p_1} - 0$.

(1) The revenue of NSP $S_1$ can be expressed as the follows (The number of users in NSP $S_1$’s network can be gotten by Lemma 1).

$$R'_1 = x'_1 \ast p'_1 = \left[1 - F'(\tau'_1)\right] \ast p'_1 = \left[1 - \frac{\tau'_1}{\varphi'}\right] \ast p'_1$$

$$= \left[1 - \frac{p'_1 - p'_2}{q'_1 - q'_2} \cdot \frac{1}{\varphi'}\right] p'_1 = \left[\varphi' - \frac{p'_1 - p'_2}{q'_1 - q'_2} \cdot \frac{p'_1}{\varphi'}\right]$$

(A-1)

To maximize $R'_1$, we have the following optimal condition,

$$\frac{dR'_1}{dp'_1} = 0$$

(A-2)

Therefore, we can get the optimal price by solving Eq. (A-2),

$$p'_1 = BR_1(p'_{-1}) = \frac{1}{2} p'_2 + \frac{1}{2} \varphi'(q'_1 - q'_2)$$

(A-3)

The optimal price of NSP $S_1$ is a function of $p'_{-1}$, which is defined as function $BR_1(p'_{-1})$. In game theory [12], we call the function $BR_1(p'_{-1})$ as best response function of player NSP $S_1$.

(2) The revenue of NSP $S_i$, for $i = 2, ..., n - 1$, can be expressed as

$$R'_i = x'_i \ast p'_i = p'_i \left[1 - F'(\tau'_i)\right]$$

$$= p'_i \left[1 - \frac{1}{\varphi'} \left(\tau'_i - \tau'_1\right)\right]$$

$$= \frac{p'_i \left(p''_{i-1} - p'_1\right)}{\varphi' \left(q''_{i-1} - q'_1\right)} - \frac{p'_i \left(p''_{i+1} - p'_1\right)}{\varphi' \left(q''_{i+1} - q'_1\right)}$$

(A-4)

To maximize $R'_i$ by letting $\frac{dR'_i}{dp'_i} = 0$, we can get the best response function of player NSP $S_i$ as shown in Eq. (A-5).

$$p'_i = BR_i(p'_{-1})$$

$$= \frac{q''_{i+1} - q'_1}{2(q''_{i+1} - q'_1)} p''_{i+1} + \frac{q''_{i-1} - q'_1}{2(q''_{i-1} - q'_1)} p''_{i-1},$$

for $i = 2, ..., n - 1$ (A-5)

(3) The revenue of NSP $S_n$ can be expressed as follows,

$$R'_n = x'_n \ast p'_n = p'_n \left[1 - F'(\tau'_n)\right]$$

$$= p'_n \left[1 - \frac{1}{\varphi'} \left(\tau'_n - \tau'_1\right)\right]$$

$$= \frac{p'_n \left(p''_{n-1} - p'_1\right)}{\varphi' \left(q''_{n-1} - q'_1\right)} - \frac{p'_n \left(p''_{n} - p'_1\right)}{\varphi' \left(q''_{n} - q'_1\right)}$$

(A-6)

To maximize $R'_n$ by letting $\frac{dR'_n}{dp'_n} = 0$, we can get the best response function of player NSP $S_n$ as that in Eq. (A-7).

$$p'_n = BR_n(p'_{n-1}) = \frac{q''_{n}}{2q''_{n-1}} p''_{n-1}$$

(A-7)

If we define

$$\alpha'_i = \left\{ \begin{array}{ll} \frac{1}{2} p'_i + \alpha'_0 & \text{if } i = 1 \\ \alpha'_i - p''_{i-1} + \beta'_i p''_{i+1} & \text{if } i = 2, ..., n - 1 \\ \alpha'_n p''_{n-1} & \text{else if } i = n \end{array} \right.$$ (A-8)

$$\beta'_i = \frac{q''_{i+1} - q'_1}{2(q''_{i+1} - q'_1)}$$ for $i = 2, ..., n - 1$ (A-9)

NSPs’ best response functions (see Eq. (A-3), Eq. (A-5), Eq. (A-7)) can be expressed as the follows

$$p'_i = BR_i(p'_{-1}) = \left\{ \begin{array}{ll} \frac{1}{2} p'_i + \alpha'_0 & \text{if } i = 1 \\ \alpha'_i - p''_{i-1} + \beta'_i p''_{i+1} & \text{if } i = 2, ..., n - 1 \\ \alpha'_n p''_{n-1} & \text{else if } i = n \end{array} \right.$$ (A-10)

It is obvious the the solution of linear equations in Eq. (A-10) is the Nash Equilibrium of the game $\Gamma$.

If we can proof that there is a unique solution for linear equations in Eq. (A-10), we can conclude that there is a unique Nash Equilibrium of the game $\Gamma$.

We can express the linear equations in Eq. (A-10) as the following way,

$$p' = \left( \begin{array}{c} p'_1 \\ p'_2 \\ p'_3 \\ \vdots \\ p'_{n-1} \\ p'_n \end{array} \right) = \left( \begin{array}{cccccccc} \frac{1}{2} p'_1 + \alpha'_0 & \alpha'_1 & \alpha'_2 & \ldots & \alpha'_{n-1} & 0 \\ \alpha'_1 & \frac{1}{2} p'_2 + \beta'_1 p'_3 & \alpha'_3 & \ldots & 0 & \alpha'_n \\ \alpha'_2 & \beta'_3 & \frac{1}{2} p'_4 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha'_{n-1} & 0 & 0 & \ldots & \frac{1}{2} p''_{n-1} + \beta''_{n-1} p''_{n-1} & 0 \\ \alpha'_n & 0 & 0 & \ldots & 0 & \frac{1}{2} p''_{n} + \beta''_{n} p''_{n} \end{array} \right)$$

$$= \left( \begin{array}{c} \alpha'_0 \\ \alpha'_1 \beta'_1 \alpha'_3 \ldots \alpha'_{n-1} \alpha'_n \\ \alpha'_1 \beta'_3 \alpha'_5 \ldots \beta'_{n-1} \alpha'_{n-1} \alpha'_n \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ \alpha'_1 \beta'_3 \alpha'_5 \ldots \beta'_{n-1} \alpha'_{n-1} \alpha'_n \end{array} \right)$$

If we define matrix $A$ and vector $p'$ as follows,

$$A = \left( \begin{array}{cccccccc} 0 & \frac{1}{2} & 0 & 0 & \ldots & 0 \\ \alpha'_1 & 0 & \beta'_1 & 0 & \ldots & 0 \\ 0 & \alpha'_2 & 0 & \beta'_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \beta'_{n-1} & 0 & \alpha'_{n-1} \\ 0 & 0 & 0 & \ldots & \alpha'_{n-1} & 0 \end{array} \right)$$

$$p' = \left( \begin{array}{c} p'_1 \\ p'_2 \\ p'_3 \\ \vdots \\ p'_{n-1} \\ p'_n \end{array} \right) \begin{array}{c} \alpha'_0 \\ \alpha'_1 \beta'_1 \alpha'_3 \ldots \alpha'_{n-1} \alpha'_n \\ \alpha'_1 \beta'_3 \alpha'_5 \ldots \beta'_{n-1} \alpha'_{n-1} \alpha'_n \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ \alpha'_1 \beta'_3 \alpha'_5 \ldots \beta'_{n-1} \alpha'_{n-1} \alpha'_n \\ \end{array}$$
\[ \mathbf{\mu}' = (a_{ij}', 0, \ldots, 0)' \]  
(A-11)

We can express the best response functions in Eq. (A-10) of \( n \) NSPs as the follows

\[ \mathbf{p}' = \mathbf{A} \mathbf{p}' + \mathbf{\mu}' \]  
(A-12)

The Eq. (A-12) can be written as the follows

\[ (\mathbf{I} - \mathbf{A}) \mathbf{p}' = \mathbf{\mu}' \]  
(A-13)

where \( \mathbf{I} \) is an \( n \times n \) unit matrix with ones on the main diagonal and zeros elsewhere.

By Cramer’s rule [20], the necessary and sufficient condition for unique solution of Eq. (A-13) is shown as that in Eq. (A-14).

\[ |\mathbf{I} - \mathbf{A}| \neq 0 \]  
(A-14)

And the Nash Equilibrium price is

\[ p_{\text{NE}}^i = \frac{|(\mathbf{I} - \mathbf{A})_{ij}|}{|\mathbf{I} - \mathbf{A}|} \]  
(A-15)

where \( (\mathbf{I} - \mathbf{A}) \) is the matrix formed by replacing the \( i \)th column of \( (\mathbf{I} - \mathbf{A}) \) by the column vector \( \mathbf{\mu}' \). The operator \(| \cdot |\) on a matrix denotes the determinant [20] of the matrix. Thus, \(|(\mathbf{I} - \mathbf{A})\)| is determinant of the matrix \( (\mathbf{I} - \mathbf{A}) \), \(|(\mathbf{I} - \mathbf{A})_{ij}\)| is determinant of the matrix \( (\mathbf{I} - \mathbf{A})_{ij} \).

Please note that the solution from Cramer’s rule is not necessarily positive, therefore, the non positive solution should be omitted.

### Appendix B: Proof of Two Properties

For the probability vector of each NSP, \( \delta_i(t) = [\delta_{i1}(t), \ldots, \delta_{im}(t)] \), two properties are satisfied:

- (1) \( \sum_{j=1}^{m} \delta_{ij}(t) = 1 \) \( \forall t \);
- (2) \( \delta_{ij}(t) \geq 0 \) \( \forall t \).

Firstly, we prove property (1) by mathematical induction as follows:

- (i) It is obvious that when \( t = 0 \), \( \sum_{j=1}^{m} \delta_{ij}(0) = 1 \). Because in step 1 of the learning algorithm in Fig. 3, initial probability vector \( \delta_i(0) \) is set to ensure this property to be satisfied.
- (ii) It is assumed that when \( t = k \), property (1) is satisfied. This means that the following Eq. (A-16) is true,

\[ \sum_{j=1}^{m} \delta_{ij}(k) = 1 \]  
(A-16)

We have to proof that \( \sum_{j=1}^{m} \delta_{ij}(k + 1) = 1 \) is also correct. According the update rule in Eq. (21), if \( a_i(k) \neq p_{ij} \),

\[ \delta_{ij}(k + 1) = \delta_{ij}(k) - \gamma R_i(k) \delta_{ij}(k) \]  
(A-17)

if \( a_i(k) = p_{ij} \),

\[ \delta_{ij}(k + 1) = \delta_{ij}(k) + \gamma R_i(k) \sum_{s \neq j} \delta_{is}(k) \]  
(A-18)

By summing Eq. (A-17) for all \( s \) in \( \{1, \ldots, m\} \) except \( j \), we have the following Eq. (A-19)

\[ \sum_{s \neq j} \delta_{is}(k + 1) = \sum_{s \neq j} \delta_{is}(k) - \gamma R_i(k) \sum_{s \neq j} \delta_{is}(k) \]  
(A-19)

By combining Eq. (A-18) and Eq. (A-19), we have the following Eq. (A-20)

\[ \delta_{ij}(k + 1) + \sum_{s \neq j} \delta_{is}(k + 1) = \delta_{ij}(k) + \gamma R_i(k) \sum_{s \neq j} \delta_{is}(k) + \sum_{s \neq j} \delta_{is}(k) - \gamma R_i(k) \sum_{s \neq j} \delta_{is}(k) \]  
(A-20)

Eq. (A-20) can be written as shown in Eq. (A-21)

\[ \sum_{j=1}^{m} \delta_{ij}(k + 1) = \sum_{j=1}^{m} \delta_{is}(k) \]  
(A-21)

Since the right hand side of Eq. (A-21) equals to 1 by Eq. (A-16), we have \( \sum_{j=1}^{m} \delta_{is}(k + 1) = 1 \). This means that when \( t = k + 1 \), property (1) is also satisfied.

By mathematical induction, we can conclude that \( \sum_{j=1}^{m} \delta_{ij}(t) = 1 \) for all \( t \).

Similarly, property (2) can also be proofed by mathematical induction. The full proof is omitted. We just show key points for proof. If \( a_i(k) = p_{ij} \), Eq. (A-18) shows that the probability \( \delta_{ij}(k + 1) \) will not decrease, there is no probability for \( \delta_{ij}(k + 1) \) becomes less than 0. If \( a_i(k) = p_{ij} \), by Eq. (A-17), \( \delta_{ij}(k + 1) \) will decrease. We can rewrite Eq. (A-17) as shown in Eq. (A-22)

\[ \delta_{ij}(k + 1) = \delta_{ij}(k) - \gamma R_i(k) \delta_{ij}(k) \]  
(A-22)

The step size \( \gamma \) is in \( (0, 1) \), and \( R_i(k) \) is normalized in step 3 of the learning algorithm in Fig. 3, which means \( R_i(k) \) is in \( (0, 1) \). Then \( \gamma R_i(k) \) is in \( (0, 1) \). Therefore, \( (1 - \gamma R_i(k)) \) is in \( (0, 1) \). Obviously, \( \delta_{ij}(k) \) is a probability value in \( [0, 1] \). So \( \delta_{ij}(k)(1 - \gamma R_i(k)) \) is in \([0, 1]\). Finally, we can conclude that \( \delta_{ij}(k + 1) \geq 0 \).
Cheng Zhang received his B.S. degree in Automation from Wuhan University of Science and Technology, Wuhan, China, in 2005 and M.S. degree in Control Theory and Control Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008. From 2008 to 2013, he was a research engineer at Sony Digital Network Applications, Japan. He is currently an invited researcher at Global Information and Telecommunication Institute, Waseda University. His research interests include congestion-based pricing and game theory application in telecommunication network. He received the IEICE Young Researcher's Award in 2013. He is a member of IEICE.

Bo Gu received the B.E. degree from Tianjin University, Tianjin, China, in 2004, M.E. degree from Peking University, Beijing, China, in 2007, and Ph.D. degree from Waseda University, Tokyo, Japan, in 2013, respectively. From 2007 to 2011, he was a research engineer at Sony Digital Network Applications, Japan. He is currently an assistant professor at School of Fundamental Science and Engineering, Waseda University. His research interests include resource and mobility management of wireless networks, QoS provisioning, and next generation wireless networks. He received the IEICE Young Researcher’s Award in 2011, and the IEICE Communication Quality Conference Premium Award in 2012.

Kyoko Yamori received her B.A. degree in business administration and M.A. and Ph.D. degrees in information management science from Asahi University in 1995, 1997, 2000. She is presently an associate professor at Department of Management Information, Asahi University, and a visiting associate professor at Global Information and Telecommunication Institute, Waseda University. She received the IEICE Switching System Research Award in 2001, the IEICE Young Researcher’s Award in 2005, and the IEICE Best Paper Award in 2005. She is a senior member of IEICE.

Sugang Xu received his B.E. and M.E. degrees in computer engineering from Beijing Polytechnic University, Beijing, China, in 1994 and 1997, respectively, and Ph.D. degree in information and communication engineering at the University of Tokyo, Tokyo, Japan, in 2002. He joined Global Information and Telecommunication Institute, Waseda University in 2002, as a research associate there. Since 2005, he joined National Institute of Information and Communications Technology (NICT), Tokyo, Japan, as an expert researcher. He is also an invited researcher at Waseda University. His research interests include algorithms, network architectures, photonic network control, optical grid network systems, parallel and distributed processing. He is a member of IEEE and IEICE.

Yoshiaki Tanaka received the B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1974, 1976, and 1979, respectively. He became a staff at Department of Electrical Engineering, the University of Tokyo, in 1979, and has been engaged in teaching and researching in the fields of telecommunication networks, switching systems, and network security. He was a guest professor at Department of Communication Systems, Lund Institute of Technology, Sweden, from 1986 to 1987. He was also a visiting researcher at Institute for Posts and Telecommunications Policy, from 1988 to 1991, and at Institute for Monetary and Economic Studies, Bank of Japan, from 1994 to 1996. He is presently a professor at Global Information and Telecommunication Institute, Waseda University, and a visiting professor at National Institute of Informatics. He received the IEEE Outstanding Student Award in 1977, the Niwa Memorial Prize in 1980, the IEICE Achievement Award in 1980, the Okawa Publication Prize in 1994, the TAF Telecom System Technology Award in 1995 and in 2006, the IEICE Information Network Research Award in 1996, in 2001, in 2004, and in 2006, the IEICE Communications Society Activity Testimonial in 1997 and in 1998, the IEICE Switching System Research Award in 2001, the IEICE Best Paper Award in 2005, the IEICE Network System Research Award in 2006, in 2008, and in 2011, the IEICE Communications Society Activity Award in 2008, the Commendation by Minister for Internal Affairs and Communications in 2009, the APNOMS Best Paper Award in 2009 and in 2012, and the IEICE Distinguished Achievement and Contributions Award in 2013. He is a Fellow of IEEE.