Towards Context-Aware Internet Services with Unselfish Clients

Yu Lu
Department of Electrical and Computer Engineering
National University of Singapore
Email: lulu@nus.edu.sg

Wai-Choong Wong
Department of Electrical and Computer Engineering
National University of Singapore
Email:wong_lawrence@nus.edu.sg

***Abstract*—The rapid advancement in context-aware computing techniques greatly facilitates capturing the context information of the Internet clients, which can be utilized by the Internet services and applications to manage different network resources. Based on the built context-aware system and the deduced highly abstract context information, we propose a resource distribution framework that incentivizes context sharing and moderate competition among the selfish but rational Internet clients. Under the proposed framework, the Internet client, which provides its both negative and positive context, can be assigned to the prioritized class and accordingly receive more resources from the resource owner. Meanwhile, all the clients are motivated by the framework to compete moderately and the aggressive ones are penalized by receiving fewer resources. The Web system exemplar is used to aid understanding of our motivation. We further model the resource distribution process as a non-cooperative game and accordingly provide the theoretical insight of the proposed framework.**

I. INTRODUCTION AND MOTIVATION

Context-aware computing leverages on various sensors to collect a system's environmental and contextual information, which helps the system to increase its usability and effectiveness. The rapid progress in context-aware computing techniques greatly facilitate collecting and ascertaining context information of Internet end-users. Proper utilization of the highly abstract and substantive Internet client's context information presents major opportunities to enhance the Internet as a user-centric, context-aware and intelligent communication system. There has been some research dedicated to building such a systems for managing different network resources and provide adaptable Internet services [1], [2]. The Internet client refers to both the Internet end-user and the client side software, such as the Web end-user and the Web browser.

However, each Internet client is rational and selfish in nature, and therefore the Internet client may not be willing to provide its context information, especially the negative context information that may lead to lower priority in receiving network resources or services. Moreover, the selfish nature also means that the Internet clients compete aggressively for the limited resources on the Internet. Based on the captured context information, we propose a resource distribution framework with the following explicit design objectives: (1) *encourage the Internet clients to provide and share their context information, especially the negative ones*; (2) *encourage moderate competition among the Internet clients*.

We take the contemporary World Wide Web system as an illustrative example. The Web system adopts the client-server architecture and leverages on the HTTP protocol for transferring Web pages between the Web server and the Web clients. On the Web server side, the child process usually creates multiple worker threads to handle the incoming HTTP connection requests: normally, one worker thread serves one HTTP connection at a time. Too many worker threads can easily cause thrashing in virtual memory system and considerably degrade the server performance. In practice, a fixed limit is always imposed on the maximum number of worker threads. For example, the default maximum number in an Apache HTTP Server 2.2 is set to 256. Therefore, the worker threads held by the Web server always become the scarce resource in the Web system. On the Web client side, HTTP/1.1 specifies that "*Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy*". However, today's commercial Web browsers frequently violate this specification: the default maximum value of Firefox 3.6 is set to 6 parallel persistent connections per server, and 8 persistent connections per proxy. Recently, the latest Internet Explorer and Google Chrome also aggressively adopt at least 6 parallel persistent connections per server as their default settings. However, the existing Web system simply handles all incoming HTTP requests and clients equally, and maintains a FIFO queue with the drop-tail queue management. The context information of the Web clients can be directly used by the Web servers to differentiate between the worker threads that are being used by the real end-users and the worker threads that are just grabbed by the aggressive Web browsers at the client side. Accordingly, the limited worker threads can be properly allocated to the starving Internet clients.

In short, the limited resource, e.g., the worker threads held by the Web server, often faces excessive competition from the selfish but rational Internet clients. The captured context information can directly help the Internet systems or services to distribute the limited resource in an optimal way. Therefore, it is necessary to explore an effective framework to encourage the actual context sharing and the moderate competition among the selfish but rational Internet clients.

The structure of the paper is as follows: Section I discusses

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

the motivation and background of this work. Section II describes the two basic categories of the advanced context information deduced by the built context-aware system. Section III presents the proposed resource distribution framework with the theoretical analysis. Section IV concludes the paper.

II. SYSTEM DESCRIPTION

The existent context-aware systems always adopt several context information acquisition approaches, typically including the direct sensor access, the context server based and the middleware based approaches [3]. The middleware based approach uses a method of encapsulation to separate and hide low-level sensing details to ease rapid prototyping and implementing of a context-aware system. Such separation of detecting and using context is also necessary to improve the extensibility and the reusability of a context-aware system. The middleware based approach thus has been widely adopted, such as in SOCAM [4] system, to support acquiring, discovering, interpreting and disseminating of the context information.

A middleware based context-aware system architecture typically consists of three functional modules: Context Sensing Layer, Context Middleware Layer and Context Application Layer. Based on such layered architecture, we have designed and implemented a context-aware system to capture and utilize the context information of the Internet clients [2]. The Context Sensing Layer deploys a number of physical sensors and virtual sensors: physical sensor refers to the hardware sensor to capture the basic physical context, while virtual sensor collects basic context from software environment including the operating system. In the Context Middleware Layer, the context inference engine or reasoning model performs the context abstraction and reasoning task to translate basic context data into highly abstract and significant context information, which are termed as the ***Key Context Information*** in this paper. Utilization of the Key Context Information would facilitate building context-aware Internet services and disseminating context information over the Internet. Different Internet services and the protocol stack are located in the Context Application Layer, where they do not necessarily need to know details of the basic context data but directly make use of the Key Context Information. In addition, a set of control rules are required for the corresponding Internet services to trigger the actions when a certain Key Context Information is deduced and delivered.

In the context-aware system we built, two abstract and significant categories of the Key Context Information have been defined:

1) **Communicating State** (**CS**): The end-user interacts with the specific Internet service and the information exchange occurs between them.
2) **Inactive State** (**IS**): The end-user is detached from the specific Internet service and no information exchange occurs between them.

The above-defined Key Context Information covers a wide range of the captured basic context information. For example, the Internet end-user browsing the Web pages can be translated into the **Communicating State** with the Web service, and the end-user temporarily leaving the computer can be translated into the **Inactive State** with the Web service.

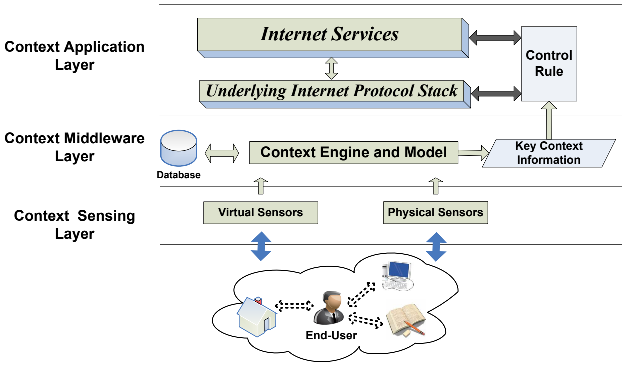

Fig. 1. Internet services operating at the Context Application Layer.

On the basis of the built context-aware system and the above defined Key Context Information, we propose a resource distribution framework that aims to encourage actual context sharing and moderate competition among the Internet clients.

III. A RESOURCE DISTRIBUTION FRAMEWORK

A. Framework Workflow

Assume that μ basic units of the limited resource are held by the server (or server cluster), which is termed *resource owner* in this framework. The limited resource can be of any type, such as worker thread, bandwidth, CPU time, memory, etc. A finite set of the Internet clients, denoted by $P_i,\ i \in I = \{1, 2, ..., N\}$, compete for the given limited resources. All Internet clients update and transfer their latest Key Context Information to the resource owner through interoperable communication protocols or approaches, such as the XML Protocol (XMLP) [5] or JAVA RMI (Remote Method Invocation) [6]. The resource owner maintains a database managing the delivered Key Context Information with the time stamp recording the update time. On the resource owner side, the time domain is divided into fixed-size time slots $T_j,\ j \in \{1, 2, ..., +\infty\}$. Each individual time slot can be further divided into two parts: an Initialization Period and a subsequent Hold Period. The resource distribution process only occurs in the Initialization Period, but its result effects the entire Hold Period and part of the next Initialization Period. Within each Initialization Period, the interaction steps between the resource owner and the Internet clients, i.e., the basic workflow of the resource distribution framework, can be described as follows:

1) On the basis of the delivered Key Context Information, the resource owner first performs the ***Willingness Value Update Algorithm (WV-UA)*** to calculate its willingness value for each Internet client. The willingness value, say $w_i(T_j)$, reflects the amount of the resource that the resource owner is willing to offer to the client P_i during the current time slot T_j. After performing the WV-UA,

the resource owner instantly informs each client the assigned willingness value.

2) After receiving the assigned willingness value, each client, say P_i, takes proper strategy to select a bidding value $b_i(T_j)$ and sends it back to the resource owner. The bidding value $b_i(T_j)$ reflects the amount of the resource that the client P_i expects to obtain from the resource owner during the current time slot T_j. Meanwhile, based on its bidding value $b_i(T_j)$, the client P_i takes actions according to the pre-defined Control Rules.
3) With all the received bidding values as well as the original willingness values, the resource owner executes the ***Resource Distribution Algorithm (RDA)*** to obtain the final resource distribution result. The result $x_i(T_j)$, $\forall i \in I$ is the amount of the resource finally assigned to the client P_i for the current time slot T_j. Based on the final resource distribution result, the resource owner side takes actions according to the pre-designed Control Rules.

Remark 1: If any individual client cannot timely provide its bidding value before the STEP 3 starts, the resource owner can simply assume that such client uses the given willingness value as its bidding value.

Remark 2: Since the clients only need to update and transfer their Key Context Information to the resource owner when it changes, synchronization between the Internet clients and the resource owner is not required.

The given three-step procedure defines the basic workflow of the resource distribution framework. The STEP 1 and the STEP 3 of the framework workflow require the ***Willingness Value Update Algorithm*** and the ***Resource Distribution Algorithm***, which will be discussed in the following subsections, respectively. The STEP 2 requires the proper bidding strategy, which is discussed in the subsection of theoretical analysis.

B. Willingness Value Update Algorithm

In the STEP 1 of the framework workflow, the willingness value $w_i(T_j)$ reflects the amount of the resource that the resource owner is willing to offer to the client P_i during the time slot T_j. The main objective of introducing the willingness value concept and implementing the WV-UA is to make a preliminary resource distribution purely based on the current and historical Key Context Information of the Internet clients.

Based on the defined two categories of the Key Context Information, i.e., the Communicating State (CS) and the Inactive State (IS), we present a practical WV-UA to further demonstrate the above design principles. At the beginning of each time slot, i.e., the STEP 1 of the Initialization Period, the resource owner first categorizes all clients into four classes according to their Key Context Information during the previous and the current time slot, which are denoted by $s_i(T_{j-1})$ and $s_i(T_j)$, $\forall i \in I$:

$$\begin{aligned}
C_1 &= \{P_i : \ s_i(T_j) = CS \,\&\, s_i(T_{j-1}) = IS \ , \ i \in I\} \\
C_2 &= \{P_i : \ s_i(T_j) = CS \,\&\, s_i(T_{j-1}) = CS \ , \ i \in I\} \\
C_3 &= \{P_i : \ s_i(T_j) = IS \,\&\, s_i(T_{j-1}) = CS \ , \ i \in I\} \\
C_4 &= \{P_i : \ s_i(T_j) = IS \,\&\, s_i(T_{j-1}) = IS \ , \ i \in I\}.
\end{aligned}$$

The classes C_1 and C_2 include all the clients in the CS at the beginning of the current time slot T_j, while during the previous time slot T_{j-1}, they were in the IS and the CS, respectively. The classes C_3 and C_4 involve all the clients in the IS at the beginning of the current time slot T_j, while during the previous time slot T_{j-1}, they were in the CS and the IS, respectively.

As indicated in the WV-UA design principles, the resource owner can directly use the value μ, i.e., the total basic units of the limited resource, as the sum of the assigned willingness values. Since the classes C_1 and C_2 include all the clients that are currently in the CS, most of the willingness value should be assigned to these two classes. Hence, a splitting parameter θ $(0 < \theta < 1)$ can be specified by the resource owner to divide μ into two parts: the first part for the classes C_1 and C_2 and the second part for the classes C_3 and C_4. Given N_1 and N_2 are the number of clients in the class C_1 and the class C_2, the resource owner performs a WV-UA to calculate its willingness value for these two classes in the current time slot T_j, which is described by the following pseudo-code.

Algorithm 1 Willingness Value Update Algorithm (WV-UA)

```
Input: $\mu$, $\theta$, $C_1$, $C_2$, $N_1$, $N_2$.
Output: willingness values $w_i(T_j)$ for all clients in the classes $C_1$ and $C_2$.
$\eta = \mu * \theta$;
$\lambda = \left\lfloor \frac{\eta}{N_1+N_2} \right\rfloor$;
for $i = 1 \to N_1 + N_2$ do
    $w_i(T_j) = \lambda$;
end for
$\eta' = \eta - \lambda * (N_1 + N_2)$;
if $\eta' \leq N_1$ then
    Randomly choose $\eta'$ clients in $C_1$, denoted by $C_1'$;
    $w_i(T_j) = w_i(T_j) + 1$ for $P_i \in C_1'$;
else $\{\eta' \geq N_1\}$
    Randomly choose $\eta' - N_1$ clients in $C_2$, denoted by $C_2'$;
    $w_i(T_j) = w_i(T_j) + 1$ for $P_i \in C_1 \cup C_2'$;
end if
```

The given WV-UA first determines η, i.e., the sum of the assigned willingness values to all the clients in the classes C_1 and C_2. Given the client numbers N_1 and N_2, the WV-UA calculates the average willingness value $\frac{\eta}{N_1+N_2}$, and obtains its integer part λ by using the floor function $\lfloor \cdot \rfloor$. λ is essentially the maximum value that can be equally set for each client in both classes. The WV-UA then sets λ as each client's basic willingness value. An extreme example is $N_1 + N_2 > \eta$, then the basic willingness value $\lambda = 0$ would be assigned to each client. After assigning the basic λ, the WV-UA calculates the residual value in η, which is denoted by η'. If η' is smaller than the total number of clients in C_1, the WV-UA would randomly pick up η' clients from C_1 and increase their corresponding willingness values by 1. Otherwise, the WV-UA would randomly pick up $\eta' - N_1$ clients from the class C_2, and increase their corresponding willingness values as well as the willingness value of each client in the class C_1 by 1.

In the given WV-UA, the clients in the class C_1 receive higher willingness values than the clients in the class C_2 as long as the residual value $\eta' > 0$. Note that the prioritized class C_1 requires the condition $s_i(T_{j-1}) = IS$. Hence, the WV-UA encourages the Internet clients to actively and timely update their actual Key Context Information, especially the IS, and incentivizes such honest behavior by assigning the clients higher willingness value when they transit back to the CS. Similarly, the given WV-UA applies to the classes C_3 and C_4 with the total willingness values $\mu * (1 - \theta)$.

Remark 1: The splitting parameter θ can be either a constant (e.g., $\theta = 0.9$) or configured dynamically at the beginning of each time slot by the resource owner.

Remark 2: Selecting the basic unit of the limited resource is an Internet service-specific task. For example, in the Web system, the worker threads held by the Web servers would be the limited resource, where one worker thread can be simply regarded as the basic unit. In the streaming media system, the uplink bandwidth on the streaming server side would be the limited resource, where the basic unit of the uplink bandwidth can be set to 512 Kbps or 1 Mbps.

C. Resource Distribution Algorithm

In the STEP 3 of the framework workflow, the resource owner executes the Resource Distribution Algorithm (RDA) to calculate the final resource distribution result. The following objectives should be considered when designing the RDA:

- **Moderate Competition Support**: Any Internet client, whose bidding value is equal to, or lower than, the assigned willingness value, i.e., $b_i(T_j) \leq w_i(T_j)$, would receive its bidding amount of resource. Any Internet client, whose bidding value is much higher than the assigned willingness value, i.e., $b_i(T_j) \gg w_i(T_j)$, would receive less or even no resource.
- **Fairness**: Any two Internet clients, who hold the same willingness value and the same bidding value, would receive the same amount of resource.

Based on the deduced Communicating State (CS) and the Inactive State (IS), we present a practical RDA based on water filling algorithm [7], [8]. Each Internet client, say P_i, is treated as a bucket with the area $b_i(T_j)$ and the width $w_i(T_j)$. Each bucket has a bottom thickness $\frac{b_i(T_j)}{w_i(T_j)}$, and accordingly its total height amounts to $\frac{2b_i(T_j)}{w_i(T_j)}$. The height of the bucket essentially reflects the aggressiveness level of the corresponding client: higher bucket indicates more aggressive in principle.

According to the bucket height, the RDA aims to divide all buckets (clients) into three groups: $G_1 = \{P_1, ..., P_L\}$, $G_2 = \{P_{L+1}, ..., P_M\}$ and $G_3 = \{P_{M+1}, ..., P_N\}$, where $1 \leq L \leq M \leq N$. The RDA regards G_1 as the "moderate" group, G_2 as the "normal" group and G_3 as the "aggressive" group. For the clients in the group G_1, the RDA fulfils all their demands, i.e., offering their bidding amount of resource. For the clients in the group G_2, the RDA partly satisfies their demands by offering a certain amount of resource, which ensures that all buckets in the group G_2 reach the same **final height**, denoted by h. For the clients in the group G_3, the RDA does not offer any resource to them. The RDA can be expressed by the pseudo-code in Algorithm 2.

Algorithm 2 Resource Distribution Algorithm (RDA)

```
Input: $\mu$, $b_i$ and $w_i$, $\forall i \in I$ for the current time slot .
Output: Three groups $G_1$, $G_2$, $G_3$, leftover resource $\mu'$.
Init: Sort all clients in ascending $\frac{b_i}{w_i}$ order, denoted
by $\left\{\frac{b_1}{w_1}, \frac{b_2}{w_2}, \dots, \frac{b_N}{w_N}\right\}$,
      $low = \frac{2b_1}{w_1}$, $high = \frac{b_N}{w_N}$, $b_0 = 0$.
Routine 1:      /* pick out all members in $G_1$ */
for $k = 1 \rightarrow N$ do
  $\alpha = \sum_{i=1}^{i=k} b_i$;
  $j = k + 1$;
  while $low \succ \frac{b_j}{w_j}$ do
    $\alpha += (low * w_j - b_j)$;
    $j++$;
  end while
  if $\alpha \leq \mu$ then
    $L = k$;      /*$P_k$ is assigned to $G_1$ */
    $low = \frac{2b_{k+1}}{w_{k+1}}$, $\alpha = 0$;
  else $\{\alpha \succ \mu\}$
    $\mu -= \sum_{i=0}^{i=k-1} b_i$; /*$P_k$ is the first member in $G_2$ */
    $\mu' = \mu$, exit for;
  end if
end for
Routine 2:      /* pick out all members in $G_3$ */
for $k = N \rightarrow L + 1$ do
  if $high \geq \frac{2b_{L+1}}{w_{L+1}}$ then
    $high = \frac{b_{k-1}}{w_{k-1}}$;      /*$P_k$ is assigned to $G_3$ */
  else if $high < \frac{2b_{L+1}}{w_{L+1}}$ then
    $\beta = \sum_{j=L+1}^{k} (high * w_j - b_j)$;
    if $\beta \geq \mu$ then
      $high = \frac{b_{k-1}}{w_{k-1}}$; /*$P_k$ is assigned to $G_3$ */
    else $\{\beta \prec \mu\}$
      $M = k$; /* $P_k$ is the last member in $G_2$ */
      exit for;
    end if
  end if
end for
```

In Routine 1, the RDA successively selects a bucket from the shortest one and assumes it to be the last member of the group G_1. Then the RDA calculates the corresponding amount of the required resource α: if α is less than the available amount of resource μ, the selected bucket would be assigned to the group G_1 and the same procedure is applied to the next bucket; otherwise the RDA calculates the leftover resource and jumps to Routine 2. In Routine 2, the RDA successively selects a bucket from the tallest one and assumes it to be the first member of the group G_3. Then it calculates the corresponding amount of the required resource β: if β is larger than the leftover resource, the selected bucket would be assigned to the group G_3 and the same procedure is applied to the next bucket; otherwise the RDA ends. Since Routine 1 and Routine 2 have picked out all the buckets in the groups G_1 and G_3, the rest would be automatically assigned to the group G_2.

The final height h in the group G_2 can be calculated by

$$h = \frac{b_M}{w_M} + \frac{\mu' - \sum\limits_{i=L+1}^{M-1} (\frac{b_M}{w_M} * w_i - b_i)}{\sum\limits_{i=L+1}^{M} w_i}.$$

Finally, the resource owner can simply distribute the resource in terms of the three groups derived by the RDA:

$$x_k(T_j) = \begin{cases} b_k(T_j), & \forall k \in [1,\ L]; \\ w_k(T_j) * h - b_k(T_j), & \forall k \in [L+1,\ M]; \\ 0, & \forall k \in [M+1,\ N]. \end{cases}$$

D. Theoretical Analysis

From the theoretical perspective, we discuss the important properties of the proposed resource distribution framework with the given WV-UA and the RDA. The basic workflow of the framework determine the three-step interaction process between the resource owner and the Internet clients. Such interaction process can be modeled and analyzed as a non-cooperative game: all Internet clients can be regarded as the game players; each game player needs to adopt a bidding strategy to decide its bidding value; the given WV-UA and the RDA jointly work as the utility function and the final resource distribution results give the payoff to each game player. Therefore, we can adopt the non-cooperative game theory tool to further analyze the resource distribution framework.

Lemma: With the given WV-UA and the RDA, the bidding strategy profile $B^*(T_j) = \{b^*_c(T_j) :\ b^*_c(T_j) = w_c(T_j),\ \forall c \in I\}$ is a pure-strategy Nash equilibrium in the time slot T_j.

Proof: Consider P_L and P_M are the last member of the group G_1 and the group G_2 in the RDA, we have

$$\begin{cases} \frac{2b_L}{w_L} \le h < \frac{2b_{L+1}}{w_{L+1}} \\ \frac{2b_M}{w_M} \le 2h < \frac{2b_{M+1}}{w_{M+1}}. \end{cases}$$

where the time expression T_j can be omitted. Thus, the amount of the resource assigned to the group G_1, denoted by μ_1, satisfies

$$\mu_1 = \sum_{i=1}^{L} x_i = \sum_{i=1}^{L} b_i \le \sum_{i=1}^{L} \frac{h}{2} * w_i.$$

The resource assigned to G_2, denoted by μ_2, satisfies

$$\mu_2 = \sum_{i=L+1}^{M} x_i = \sum_{i=L+1}^{M} (h * w_i - b_i) < \sum_{i=L+1}^{M} \frac{h}{2} * w_i.$$

The resource assigned to G_3, denoted by μ_3, satisfies

$$\mu_3 = \sum_{i=M+1}^{N} x_i = 0.$$

Hence, the resource assigned to all clients satisfies

$$\mu_1 + \mu_2 + \mu_3 < \sum_{i=1}^{M} \frac{h}{2} * w_i. \tag{1}$$

Now, we prove that any client, say P_c, whose bidding value $b_c = w_c$, must be assigned to the group G_1 by the given RDA. We consider the other two possible cases:

(1) When the client P_c is assigned to the group G_2, i.e., $h < \frac{2b_c}{w_c} \le 2h$: the given WV-UA satisfies $\sum\limits_{i=1}^{N} w_i(T_j) = \mu$ and consider $b_c = w_c$, we have $h < \frac{2\mu}{\sum\limits_{i=1}^{N} w_i}$. Together with (1), we get

$$\mu_1 + \mu_2 + \mu_3 < \sum_{i=1}^{M} \frac{h}{2} * w_i < \mu * \frac{\sum\limits_{i=1}^{M} w_i}{\sum\limits_{i=1}^{N} w_i} < \mu.$$

The above inequality shows that the total assigned resource is less than the total available resource, which conflicts with the given RDA. Hence, it is impossible that the client P_c is assigned to the group G_2 by the RDA.

(2) When the player P_c is assigned to the group G_3: similarly, we get

$$\mu_1 + \mu_2 + \mu_3 < \sum_{i=1}^{M} \frac{h}{2} * w_i < \frac{\mu}{2} * \frac{\sum\limits_{i=1}^{M} w_i}{\sum\limits_{i=1}^{N} w_i} < \mu.$$

The above inequality also conflicts with the given RDA, and thus the client P_c cannot be assigned to the group G_3 by the RDA. Hence, with the given WV-UA and the RDA, any Internet client, say P_c, who bids the assigned willingness value, i.e., $b_c(T_j) = w_c(T_j)$, can be guaranteed to receive its bidding amount of resource, i.e., $x_c(T_j) = b_c(T_j)$, regardless of other clients' bidding strategy.

With the given strategy profile B^*, we have

$$\sum_{c=1}^{N} b_c = \sum_{c=1}^{N} x_c = \mu, \tag{2}$$

The above equation shows that the resource is just used up and all clients are assigned to the group G_1 by the RDA. Because no individual client, say P_c, could gain more resource by a unilateral deviation from its initial bidding strategy $b_c = w_c$, given that all the other clients insist on their own bidding values. Therefore, the strategy profile B^* is a pure-strategy Nash equilibrium of the competition game.

Proposition: Under the proposed framework with the given WV-UA and the RDA, the best policy for any individual

Internet client is to actively provide its Key Context Information and meanwhile adopt the moderate bidding strategy to compete for the limited resource.

Proof: As mentioned earlier, in general, all the Internet clients are rational and selfish in nature, and thus they would not be willing to provide their context information to the resource owner, especially the negative IS. Moreover, they always behave aggressively to acquire more resource regardless of others. The proposed framework with the given WV-UA and the RDA addresses both issues:

1) In the given WV-UA, the highest prioritized class $C_1 = \{P_i : \ s_i(T_j) = CS \ \& \ s_i(T_{j-1}) = IS \ , \ i \in I\}$ requires the clients in the IS during the previous time slot. Hence, for any rational Internet client temporarily in the IS, the best policy is not to hide it but timely update the IS to the resource owner. As a reward, when such client transits back to the CS in the new time slot, the WV-UA will promptly classify it into the first class C_1 and accordingly offer it the highest willingness value. The first part of Lemma shows that the higher willingness value received, the more resource can be guaranteed to gain from the resource owner. In other words, for any Internet clients, timely providing the negative IS would be incentivized by allocating more resource when they transit back to the CS. Therefore, any rational Internet clients would be motivated to provide both of their positive and negative Key Context Information to the resource owner.

2) When any selfish client, say P_c, attempts to acquire much more resource by adopting aggressive bidding strategies, i.e., $b_c(T_j) \gg w_c(T_j)$, such client would deviate itself far from the system Nash equilibrium $B^*(T_j)$. As a result, the client cannot gain more resource to improve its payoff, but receives less or even no resource from the resource owner. Since adopting aggressive bidding strategies suffers a significant reduction in the allocated resource, any rational Internet clients are motivated to adopt a moderate bidding strategy when competing for the resource with others. For example, the client could request the amount of the resource that equals to, or is slightly higher than, the given willingness value.

In short, with the given WV-UA and the RDA, the proposed resource distribution framework effectively encourages the Internet clients to provide their Key Context Information, and meanwhile motivates moderate competition among the Internet clients.

E. Related Work

The above analysis uses the non-cooperative game theory and the Nash equilibrium concept. Briefly Speaking, game theory [9] is a mathematical tool for modeling and analyzing the strategic interactions among rational decision makers (game players). Subsequently, it provides insight into the corresponding competitive environments and mechanisms. The non-cooperative game theory is one of the main branch of game theory. It essentially describes the situation where each selfish game player makes decisions independently and acts to maximize his own benefit [10].

The outcome of the non-cooperative game is termed as the Nash equilibrium, which indicates that no individual game player can unilaterally improve his payoff/utility given that the other players adopt the existing Nash equilibrium. One of the important applications of non-cooperative game theory is to help design the mechanism that leads independent and selfish players towards a system-wide desirable outcome [11]. A comprehensive analysis and representative examples of non-cooperative game are given in [12].

IV. CONCLUSION

In this paper, we present a resource distribution framework that can utilize the context information of the Internet clients to provide service differentiation. More importantly, the framework workflow with the given practical WV-UA and the RDA algorithms encourages the clients to actively provide their context information. Meanwhile, the framework also incentivizes moderate competition and penalizes excessive competition among the Internet clients. The resource distribution process has been further modeled as a non-cooperative game to provide theoretical insights of the framework. Our ongoing research work and project envision a new realm for introducing the highly abstract context information into different Internet resource distribution processes, which eventually leads to context-aware Internet services with unselfish Internet clients.

REFERENCES

[1] H.-L. Truong and S. Dustdar, "A Survey on Context-aware Web Service Systems," *International Journal of Web Information Systems*, vol. 5, no. 1, pp. 5–31, 2009.

[2] Y. Lu, M. Motani, and W. C. Wong, "When Ambient Intelligence Meets the Internet: User Module Framework and its Applications," *Computer Networks*, vol. 56, no. 6, pp. 1763–1781, 2012.

[3] H. Chen, "An intelligent broker architecture for pervasive context-aware systems," Ph.D. dissertation, University of Maryland, 2004.

[4] T. Gu, H. K. Pung, and D. Q. Zhang, "A Service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1–18, 2005.

[5] E. Box *et al.*, "Simple object access protocol (soap)," [online] Available: http://www.w3c.org/TR/soap.

[6] A. Wollrath, R. Riggs, and J. Waldo, "A Distributed Object Model for the Java System," Sun Microsystems, Tech. Rep., 2009.

[7] J. L. Boudec, "Rate adaptation, congestion control and fairness: A tutorial," [online] Available: http://icapeople.epfl.ch/leboudec, 2008.

[8] B. M. Richard, T, C. M. Lee, Sam, C. S. Lui, John, and K. Y. Yau, David, "Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 978–991, 2006.

[9] G. Owen, *Game Thoery, 3rd Edition*. San Diego, CA: Academic, 1995.

[10] B. Wang, Y. Wu, and K. J. R. Liu, "Game Theory for Cognitive Radio Networks: An overview," *Computer Networks*, vol. 54, pp. 2537–2561, 2010.

[11] K. Akkarajitsakul, E. Hossain, D. Niyato, and D. Kim, "Game Theoretic Approaches for Multiple Access in Wireless Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 372–395, 2011.

[12] K. Ritzberger, *Foundations of Noncooperative Game Theory*. New York, Oxford University Press, 2002.