

# A COMMUNICATION ARCHITECTURE FOR REAL-TIME AUCTIONS

Hella Kaffel Ben Ayed, Safa Kaabi Chihi and Farouk Kamoun

*CRISTAL Lab., National School of Computer Science, University of Manouba, Tunis, Tunisia*

**Abstract:** This paper explores the possibility to use a communication protocol other than HTTP under real-time auction applications in order to provide best-suited communication services. We specify a distributed communication architecture named AHS (Auction Handling System) based on the IRC architecture to support real-time auctions. While using the suitable services provided by IRC, this architecture provides auction applications with what we define as required communication services. We also specify a communication protocol, called BSA-protocol, to support interactions between auction participants and the auctioneer in a real-time auction process. This protocol uses the services provided by the IRC-client protocol as well as the channel facilities provided by the IRC architecture for group communications. We report on the encapsulation of this protocol within the IRC-client protocol and on the implementation of a prototype. The originality of this architecture lies in the fact that it both frees auction applications from communication issues and is independent from the auction protocol.

**Key words:** Auctions, communication protocols, communication services, communication architecture, interactions, IRC, messages.

## 1. INTRODUCTION

Auctions are market mechanisms whereby one seller or one buyer – named here the initiator– is involved with many buyers or sellers called bidders. More complex forms have appeared and involve many initiators and many bidders (Beam et al., 1996; Ben Ameer,2001; Ben Youssef et al., 2001; EPRI, 2001). The way participants interact is controlled by a set of rules called the auction protocol. These interactions are managed by a mediator, called the auctioneer (e.g. the auction site), which provides the insti-

tutional setting for the auction (Beam and Segev, 1997; Klein, 1997; Kumar and Feldman, 1998a; Strobel, 2000). They result in the exchange of various messages: bid, bid withdrawal (BW), bid admittance, bid rejection, price quote (PQ) and transaction notification (TN) (Wurman et al., 1998).

Well known auctions (English, Dutch, Vickrey and Combinatorial Double Auction or CDA) have been variously classified (Ben Ameer, 2001; Wurman et al., 1998) (ascending/descending, sealed/outcry, single/double, etc.). Our study concerns real-time auctions. We talk about real-time when data from auction participants must be continuously monitored and processed in a timely manner to allow for a real-time decision over the Internet (Pen et al., 1998). Real-time auctions' activities are organized in rounds governed by a clock. The duration of a round depends on the auction type and the evaluation by the auctioneer of submitted bids during a round can be scheduled, or occurs, at random times or through bidders' activity/inactivity, depending on the auction protocol (Panzini and Shrivastava, 1999; Wurman et al., 1998).

Communication issues have an important impact on the process and results of real-time auctions (Pen et al., 1998). Hence, we focus on the communication protocol that supports the transfer of messages related to real-time auctions.

Current Internet sites running real-time auctions use the HTTP protocol. This is a Request/Response protocol that does not provide any additional communication service. Hence, required services for the auction process are implemented within the auction application (Wurman et al., 1998; Kumar and Feldman, 1998a; Kumar and Feldman, 1998b; Panzini and Shrivastava, 1999). This results in overloading auction applications with communication issues. This may influence the auctioneer's quality of service and cause frustrated customers (bidders or initiators) to leave the auction site (Amza et al., 2002; Cardellini et al., 2001; Pen et al., 1998).

Our objective is to explore the possibility to use another communication protocol in the application layer of the TCP/IP stack in order to provide auction applications with adequate communication services.

First, we defined in Section 2 real-time auctions' requirements in terms of communication services. We show that the IRC protocols are best suited to the defined auction requirements (Kaabi et al., 2001; Kaabi et al., 2003). As a result, we propose in Section 3 a novel communication architecture, based on IRC and named AHS, to support real-time auctions. In Section 4, we depict the specification of the BSA-protocol. Section 5 describes the encapsulation of this protocol within the IRC-client protocol and the implementation of a prototype. Finally section 6, concludes the paper and opens up prospects for future work.

## **2. WHICH COMMUNICATION SERVICES FOR REAL-TIME AUCTIONS?**

In this section we define first the requirements of real-time auctions in terms of communication services. Then, we report on a functional comparison of: HTTP, IRC, NNTP and SMTP with regard to the provided services and underline the suitable features of IRC.

### **2.1 Real-time auctions' requirements**

In the literature, various studies have described the typical real-time process and identified the following steps: Initiator's registration, Setting up the auction event, Bidders' registrations, Request for participation, Bidding and Settlement (Beam and Segev, 1997; Kaabi et al., 2001; Klein, 1997; Kumar and Feldman, 1998b; Turban, 1997). Some of the above mentioned studies have defined a number of requirements (Panzini and Shrivastava, 1999)—such as scalability, responsiveness and consistency—to be considered. We lay emphasis on the requirements that must be met by the communication protocol underlying the auction applications. We identify two sets of services: basic and optional.

#### **Basic services:**

These are always required and are considered as mandatory.

- i. Synchronous and push modes: The synchronous mode enables real-time interactions between bidders/initiators and the auctioneer. The Push mode is recommended to minimize the bidders' reaction time and increase the system's reactivity (Kumar and Feldman, 1998b; Panzini and Shrivastava, 1999).
- ii. Group communications: These are required to support the broadcasting of the intermediary result –PQ– and final result –TN (Maxemchuk and Shur, 2001, Panzini and Shrivastava, 1999). Group communications minimize the number of messages sent by the auctioneer and optimize the way the bandwidth is used.

#### **Optional services:**

These services are required only in some cases such as particular auction events, auction protocols or auction messages.

- iii. Duration of message validity: Usually bids and price quotes are time-sensitive in that they are valid for a certain time only and then become obsolete (Wurman et al., 1998; Kumar and Feldman, 1998b). This service allows the communication layer to reject obsolete bids or other messages transparently to the auction application.
- iv. Fairness: In our context, fairness means that all auction participants have the same privilege with regard to communications. Fairness may

be required for both bidders and initiator or for only one side (Asokan, 1998).

- v. Message-tracking: This service would allow the history of an auction event to be saved and different stages of the auction process to be kept track as proposed by Wurman (Wurman et al., 1998).
- vi. Notification: A notification is an acknowledgment that confirms the reception of a message by its recipient. This allows auction participants to be informed about intermediary stages of the bidding process and contribute to provide the non repudiation of the recipient service (Asokan, 1998). We defined three notifications:
  - \* Submission Notification (SN). It acknowledges the reception of a bid by the auctioneer.
  - \* Acceptance Notification (AN): It acknowledges the fact that a submitted bid conforms to the auction rules and has been accepted by the auctioneer for evaluation.
  - \* Rejection Notification (RN): It is sent by the auctioneer and indicates that the auctioneer, before the evaluation, has rejected a submitted bid. This rejection may be caused by the expiry of the bid or the violation of auction rules.
- vii. Security: We focus here on security services, common to all real-time auctions, which must be met by the communication protocol underlying auction applications. These services may be required for both sides: bidders/initiator and the auctioneer.
  - On the auctioneer's side:
    - Identification and authentication*: This would permit unauthorized posting of bids and violation of auction rules to be prevented.
    - Non-repudiation of bidders*: This service permits to prevent a given bidder denying having sent a bid or received a TN (Omote, 2002).
  - On the bidder's side:
    - Anonymity*: This is required when bidders should not know one another's identity from the exchange of information.
    - Non-repudiation of the auctioneer*: This service permits to prevent an auctioneer denying having received a bid (Omote, 2002).

Data integrity would allow a recipient to be sure that a received message had not been modified during transmission. Data confidentiality may also be required when submitted bids should not be accessible to unauthorized recipients (Harkavy et al., 1998).
- viii. Time stamping. It might be required that exchanged messages be time stamped when sent or when received by the auctioneer. This service provides a tool for the auctioneer to manage the temporal accuracy of arrival or sending of messages (Maxemchuk and Shur, 2001, Wurman et al., 1998).

## **2.2 Which communication protocol for real-time auctions?**

In Kaabi (Kaabi et al., 2003) we have compared the functionalities of HTTP, Internet mail, IRC and NNTP according to the defined requirements. This comparison shows that IRC is the best suited protocol with regard to the defined requirements.

IRC communications are synchronous with a push mechanism. IRC channels are defined to support group communications. With regard to fairness, IRC provides a fair distribution of messages to clients: IRC servers maintain for each client the same short period of time (2 seconds) so that all clients are served fairly by the IRC server they are connected to (IETF, 2000c). This may be considered a factor of fairness in accessing the server resources.

With regard to security, provided services are authentication and anonymity (IETF, 2000c; IETF, 2000d). The other required security services are not provided by IRC. Moreover, IRC does not provide duration of message validity nor time stamping and message-tracking.

As a result, all these missing services have to be added to the IRC functionalities in order to satisfy real-time auction requirements. In the following section, we propose the specification of a communication architecture, named AHS and based on the IRC architecture.

## **3. A PROPOSAL FOR COMMUNICATION ARCHITECTURE**

The objective of this architecture, named AHS (Auction Handling System), is to provide the required services for real-time auction applications. It takes advantage of the previously emphasized features of the IRC. Auction applications are the end-users of the AHS architecture. This work is inspired with the ITU X435 architecture that provides suitable communication services to EDI applications by using the store and forward service provided by the X400 messaging (CCITT, 1991a; CCITT, 1991b). A similar study was done to provide communication services for EDI applications while using the Internet mail architecture (Ben Azzouz,1999; Ben Azzouz et al.,2000; Shih et al., 1997).

### **3.1 Description of the AHS architecture:**

The AHS architecture is composed of the following functional components (see Figure 1):

**The Auction Server Agent (ASA):** An ASA is associated to an auction site and supports the auctioneer's activities. It may simultaneously support several auction events that have different auction protocols. An auction event

may involve more than one ASA. The set of ASAs is called Auction Server System (ASS). An ASS is a distributed auctioneer.

**The Buyer/Seller Agent (BSA):** A BSA is a user agent that can be associated with an initiator or a bidder. A BSA is attached to one ASA. Several BSAs may be attached to the same ASA.

**The Bids Store (BS):** A BS provides, when required, the capacity of storing messages such as bids and PQs or TNs for further use (message-tracking requirement). The physical situation of BS is not specified. It can be situated within an ASA, a BSA or a separate entity.

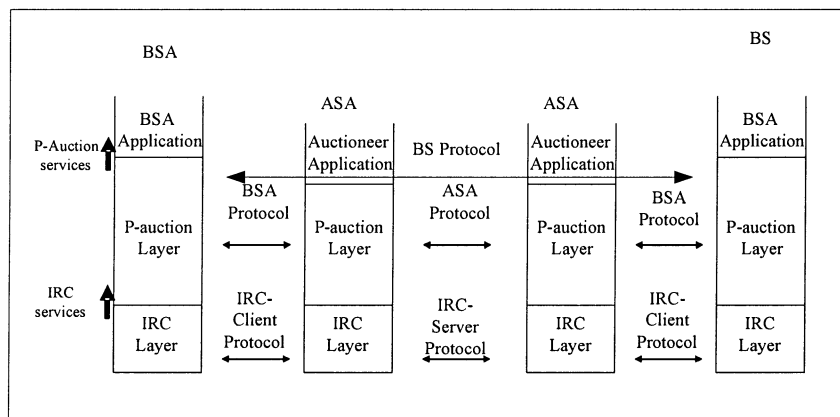
### 3.2 The Layered Model

The AHS architecture is based on the IRC architecture (IETF, 2000a). A BSA and a BS will be implemented over an IRC client and an ASA over an IRC server.

We define a new layer, called the P-auction Layer, over the IRC layer and under the auction application layer (see Figure 1). This layer is responsible for providing the auction application with the required communication services. This layered structure has the advantage of separating between the processing (in auction applications) and communication (P-auction layer). This gives the P-auction layer flexibility and independence from the auction protocol and the technology used by the auction application

### 3.3 The Protocols

The following protocols are defined to specify the interactions between functional components of AHS and how the required services are provided to end-users (see Figure 1):



F

Figure 1. The AHS layered model and protocols

**The BSA-protocol:** This protocol specifies the interactions between a BSA and an ASA. It is encapsulated within the IRC-client protocol.

**The ASA-protocol:** This protocol specifies interactions between the ASAs involved in a distributed auction event. This protocol is encapsulated within the IRC-server protocol.

**The BS-protocol:** This protocol specifies interactions between a BSA or an ASA and a BS. In the next section we specify the BSA-protocol.

## 4. SPECIFICATION OF THE BSA-PROTOCOL

We first define the interactions and then, the exchanged messages.

### 4.1 The interactions

Interactions between a BSA and an ASA are performed according to the auction process. They are the following (Kaabi et al., 2003):

1. Bidder/initiator Registration: This is initiated by a BSA. It allows a bidder or an initiator to be registered within an ASA or to have access to an auction event managed by the ASA.
2. Setting up the auction event: This is initiated by the initiator of an auction event..
3. Access: This corresponds to the Request for participation step in the auction process. This interaction is initiated by a BSA.
4. Exit: This is initiated by a BSA that asks for leaving an auction event. It may also be initiated by an ASA to force a BSA to leave the auction event.
5. Bidding: This implies the sending of messages transporting bids and BWs and notifications between a BSA to the ASA it is attached to.
6. Price quote (PQ) announcement: This consists in the intermediary result announcement and occurs after the evaluation of submitted bids. This interaction is initiated by the ASA.
7. Transaction Notification (TN) announcement: This corresponds to the broadcasting of the final result and the sending of the request notifications. This interaction is initiated by the ASA.
8. Auction Event Closure: This is initiated by the ASA attached to the initiator's BSA. It implies the sending of a set of messages to clear the auction event.

### 4.2 The BSA-Protocol messages

We define three types of messages:

- Requests (.Req): These messages require a reply from the recipient.
- Responses (.Resp): These are responses to Requests.

- Indications (.Ind): These are messages that do not require replies from the recipient.

A P-auction message is composed of two parts: the Header and the Data. The Data transports the message generated by this application. The Header is processed by the P-auction layer and is structured into fields that allow the implementation of the required services.

The first field of the Header is called *code* and has a numeric value to identify each message.

Since many ASAs may constitute an auction site and many auction events may be handled by an ASA, we define two fields: *ASA-name* and *Auction-name* to identify respectively an ASA and an auction event. A field called *BSA-name* identifies a BSA.

To achieve the goal of duration of message validity and time stamping, we introduce within the Header of concerned messages respectively two fields: *Expiry-time* and *ASA-time*. The *Expiry-time* is filled by the message's sender. The *ASA-time* is filled by the ASA just before the sending, or after the reception, of time stamped messages.

To perform a control at the P-auction layer, we define two header fields that carry semantic information related either to the auction event and bids or to the BSA. These fields are the *Parameters* and the *Type*. This makes it possible to carry out a rudimentary bid validity control within this layer independently of the structure of the Data part of the message. By so doing, we save processing time in this application since a number of invalid bids are rejected at the P-auction layer (for example, a bid submitted by the seller is rejected in case of English auction).

To provide the notification service, we define a field, called *Notif-req*, in the Header of messages that may require a notification and a specific message to carry these notifications, i.e. *Notif.Resp*.

The other fields that compose the header are described in the following section and are classified according to BSA-ASA interactions.

## 5. THE ENCAPSULATION OF THE BSA-PROTOCOL IN THE IRC CLIENT PROTOCOL

### 5.1 Technical considerations related to IRC

These aim to use properly the interesting features of the IRC standard without modifying the source code of the IRC-server or the IRC-client protocols.

- An IRC channel is created by the ASA to which the initiator is attached and is associated to an auction event. The channel-name is the same as the auction name (*Auction-name*).

- For security reasons, we have opted to use exclusively channels in the Invite mode. This mode disables any IRC client to join the channel unless he receives an Invite message from the channel operator (IETF, 2000b). This



way, we guarantee that the access to the channel used by an auction event is under the control of the auctioneer application.

- We define an IRC client, in addition to the IRC server within an ASA (see Figure 2). The nickname of this IRC client corresponds to the name of the ASA (*ASA-name*). This is to overcome the following limitations within the IRC specifications.

- a. An IRC server has not the ability to create channels. Only clients (channel operators) have this ability (IETF, 2000b). The client *ASA-name* acts as a channel operator. This way, the creation of channels for auction events is controlled by the auctioneer application that asks the Client *ASA-name* for channel creation.
- b. An IRC client cannot send textual messages to an IRC server but only to IRC clients (IETF, 2000c). BSA-protocol messages sent by a given BSA will be sent as a regular point-to-point IRC exchange between the IRC client of the BSA and the Client *ASA-name*.
- c. An IRC server has not the ability to send point-to-point textual messages to a specific client member of a channel (IETF, 2000b). Messages sent by the auctioneer application within an ASA to specific BSAs will be sent as point-to-point exchanges between the Client *ASA-name* and IRC clients within these BSAs.
- d. A channel operator has not the ability to close a channel. A channel is closed when the last member leaves it (IETF, 2000b). The Client *ASA-name* removes (Kick) all active channel members to close the auction channel.

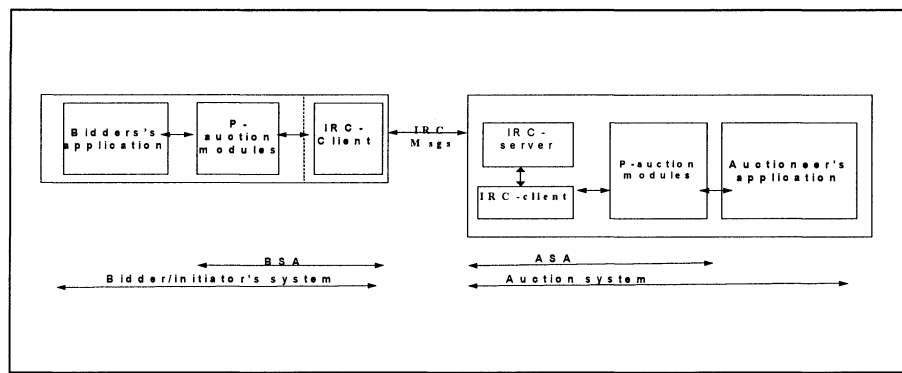


Figure 2: Design of the auctioneer's and bidders/initiators' systems

## 5.2 The Encapsulation

Each P-auction message will be encapsulated within an IRC message when sent and de-capsulated when received. An IRC message is composed of three fields: The Prefix, the Command and the Command parameters (IETF, 2000c). The Command parameters is composed of several fields containing information, such as <nickname>, <channel-name> and <text-to-be-

sent>. This latter transports the text sent by IRC clients. The Prefix, the Command and some Command parameters fields constitute the header of an IRC message. The encapsulation process consists of inserting a P-auction message within the Command parameters of an IRC message. This permits to leave the Prefix and the Command unchanged since they have a specific meaning for IRC protocols. The Data is carried within the <text-to-be-sent> field of the Command parameters. The Header's fields are in two categories:

- Fields that have a semantic equivalence with some fields within the Command parameters of the encapsulating IRC message. These are: BSA-name (eq. to <nickname> of the client within the BSA), BSA-pwd (eq. to <password>), Auction-name (eq. to <channel-name>) and ASA-name (eq. to <nickname> of the client ASA-name). These fields are carried by their corresponding within the Command parameters of the IRC message.

- Fields that do not have any semantic equivalence with the fields in the Command parameters of the encapsulating IRC message. These are conveyed within the <text-to-be-sent> part of the IRC message.

Table1 describes the structure of the BSA-Protocol messages as well as the syntax of their transporting IRC messages and their encapsulation within these messages.

**Encapsulation during the Bidder and initiator registration:** This interaction implies the sending of two messages: Register Req and Register Resp. A subset of the Header of the Register Req message will be encapsulated within IRC connection registration messages. This allows using the access control facility of IRC. The Parameters of the PASS and NICK messages will carry respectively, the *BSA-pwd* and the *BSA-name*. The remaining fields of the Register Req, as well as the Register Resp message, are carried within a separate IRC message: PRIVMSG. This message is generally used for communication between clients to carry textual information (IETF, 2000c). The <msg-target> parameter of this message contains the nickname of target client (i.e. *ASA-name* for this case).

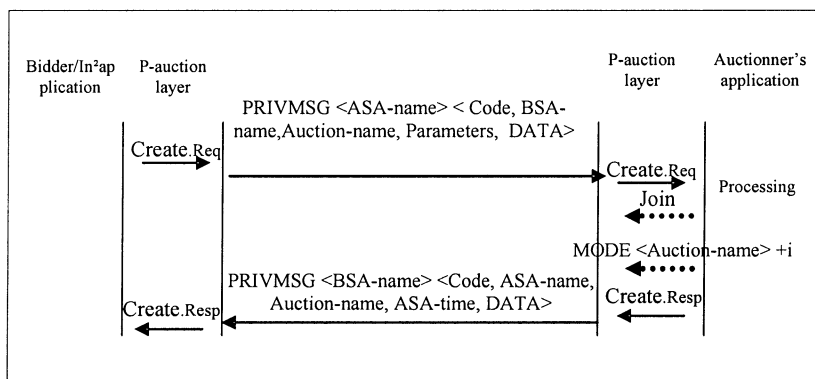


Figure 3: A successful setting up of an auction event

**Encapsulation during the Setting up of the auction event:** The Create.Reg message is first encapsulated within a PRIVMSG (carried in the <text-to-be-sent> parameter) that is sent to the Client ASA-name. As shown in Figure 3, if the decision of the auctioneer application is an acceptance of the creation, the Client ASA-name “sends” locally to the IRC server a JOIN message to ask for the channel creation. The Create.Resp message is encapsulated within a PRIVMSG message.

**Encapsulation during the Access:** This interaction requires the authentication of the BSA by the auctioneer application before this latter can be invited to join the channel assigned to the auction event. The Access.Reg message is encapsulated within a PRIVMSG in the <text-to-be-sent> parameter. After the auctioneer application has proceeded to the access control, the Access.Resp is carried by an INVITE IRC message (in case of acceptance) or a PRIVMSG message (in case of refusal). As a response to the INVITE, the P-auction layer of the BSA sends automatically a JOIN message.

**Encapsulation during the Exit:** The Exit-auction.Ind may be sent by a BSA or by an ASA. When sent by a BSA this message will be encapsulated within a PART IRC message. This IRC message is used by an IRC client to be removed from the list of active members of a channel. When sent by the ASA, the Exit-auction.Ind will be encapsulated in a KICK IRC message. This message is used by an IRC server to force the removal of an IRC client from a channel.

**Encapsulation during the bidding:** Bid.Reg, Bw.Reg and the Notif.Resp messages will be encapsulated within PRIVMSG messages.

**Encapsulation during the price quote announcement:** The PQ.Ind message will be encapsulated within a PRIVMSG message. To be broadcasted over the channel, the parameter <msgtarget> will contain the *Auction-name*.

**Encapsulation during the transaction notification announcement:** The Tr-notif.Ind will be encapsulated in a PRIVMSG message similarly to the PQ.Ind. The Tr-acc.Reg and the Tr-acc.Resp will be encapsulated within a PRIVMSG message. Several Tr-acc.Reg messages are sent if there are more than one winner.

**Encapsulation during the auction event closure:** Similarly to the Exit-auction.Ind when sent by an ASA, the Auction-closure.Ind will be encapsulated within a KICK message sent by the Client ASA-name to each active BSA. This results in the closure of the IRC channel and other resources allocated to the auction event.

### 5.3 Experimentation

The BSA-Protocol has been implemented in C language and uses an IRC software `irc2.10.3p3` over Linux. The IRC server has been configured (`ircd.conf` and `config.h` files) according to the installation instructions. The

IRC services are called by the use of two IRC primitives: `sendto-one()` for sending and `m-mypriv()` for receiving messages.

Table 1. BSA-Protocol messages and their encapsulation within IRC messages

P-auction message	IRC message	Encapsulation
Register.Req (Code, BSA-name, BSA-pwd, BSA-type, Notif.req, DATA)	PASS <password> NICK <nickname> USER <username> <mode> <hostname> <real name> PRIVMSG <msg-target><text-to-be-sent>	PASS <BSA-pwd> NICK <BSA-name> PRIVMSG <auctioneer> <Code, BSA-type, DATA>
Register.Resp (code, BSA-name, type, DATA)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <BSA-name> <Code, type, DATA>
Access.Req (Code, BSA-name, Auction-name, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <ASA-name><Code, BSA-name, Auction-name, DATA>
Access.Resp (code, BSA-name, ASA-name, Type, DATA)	INVITE <nickname><channel-name> JOIN <channel-name>{,<channel-name>}{<key>{,<key>}}.	INVITE <BSA-name> <Auction-name>  JOIN <Auction-name>
Exit-auction.Ind (Code, BSA-name, Auction-name, Data)	PART <channel>{,<channel>} KICK <channel><nickname><text-to-be-sent>	PART <Auction-name> or KICK <Auction-name><BSA-name><Data>
Bid.Req (Code, BSA-name, ASA-name, Auction-name, ASA-time, Expiry-time, Notif.req, Parameters, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <ASA-name> <Code, BSA-name, Auction-name, ASA-time, Expiry-time, Notif-req, Parameters, DATA>
Notif.Resp (code, BSA-name, ASA-name, Auction-name, ASA-time, Expiry-time, Type, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <BSA-name> <Code, ASA-name, Auction-name, ASA-time, Expiry-time, Type, DATA>
PQ.Ind (Code, ASA-name, Auction-name, ASA-time, Expiry-time, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <Auction-name> <Code, ASA-name, ASA-time, Expiry-time, DATA>
Tr-notif.Ind (Code, Auction-name, ASA-name, ASA-time, Expiry-time, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <Auction-name> <Code, ASA-name, ASA-time, Expiry-time, DATA>
Tr-acc.Req (Code, BSA-name, Auction-name, ASA-name, ASA-time, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <BSA-name> <Code, Auction-name, ASA-name, ASA-time, Data>
Tr-acc.Resp (Code, BSA-name, ASA-name, Auction-name, ASA-time, Data)	PRIVMSG <msg-target><text-to-be-sent>	PRIVMSG <ASA-name> <Code, BSA-name, ASA-name, ASA-time, Data>
Auction-closure.Ind (Code, BSA-name, Auction-name, ASA-name, ASA-time, Data)	KICK <channel><nickname><text-to-be-sent>	KICK <Auction-name> <BSA-name><Code, ASA-name, ASA-time, Data>

## 6. CONCLUSION

In this paper, we propose a distributed communication architecture for real-time auctions. We describe the specification of the BSA-Protocol involved between the auction site and the participants. We depict the encapsulation of this protocol in the IRC-server protocol. This specification provides the following communications services: synchronous group communications with push mode, time stamping, duration of message validity and notification. With regard to security, it provides access control, partial anonymity and non-repudiation of the auctioneer and of the winning bidders.

We are working on enhanced security for the BSA-protocol (such as signed notifications for non-repudiation, data integrity and confidentiality). The ASS architecture is under specification. Clock synchronization and fairness will be considered during this specification.

The advantages of this work lie in the fact that the AHS permit to free auction applications from communication issues and facilitates interoperability between these applications.

## REFERENCES

1. Amza, C., Cecchet, E., Chanda, A., Cox, A. Elnikety, S., Gil, R., Marguerite, J., Rajamani, K., and Zwaenepoel, W. (2002). "Specification and Implementation of Dynamic Web Site Benchmarks." Proceedings of the 5th IEEE Workshop on Workload Characterization (WWC-5).
2. Asokan, N. (1998). "Fairness in Electronic Commerce." PHD dissertation, University of Waterloo. <http://citeseer.ist.psu.edu/4056.html>.
3. Beam, C. and Segev, A. (1997). "Automated Negotiations: A survey of the State of the Art." Technical report 97-WO-1022, Haas School of Business, UC Berkeley.
4. Beam, C., Segev, A. and Shanthikumar, J.G. (1996). "Electronic Negotiation through Internet-based Auctions." CITM working paper 96-WP-1019.
5. Ben Ameer, H. (2001). "Enchères Multi-objets pour la Négociation Automatique et le Commerce Electronique." Master dissertation, Département d'Informatique, Faculté des Sciences et de Génie, Université Laval.
6. Ben Azzouz, L. (1999). "Emulation et Inter fonctionnement de messagerie EDI en environnement de messagerie interpersonnelle " Ph.D Dissertation, National School of Computer Sciences, University of Tunis II.
7. Ben Azzouz, L., Kaffel, H. and Kamoun, F. (2000). "An Edi User Agent over the Internet" Network and Information System Journal vol. 1(1), pp 483-504.
8. Ben Youssef, M., Alj, H., Vézeau, M. and Keller, R. (2001). "Combined Negotiations in E-commerce : Concepts and Architecture." Electronic Commerce Journal vol.1 No3, pp. 277-299.
9. Cardellini, V., Casalicchio, E. and Colajanni, M. (2001). "A Performance Study of Distributed Architectures for the Quality of Web Services." Proc. of The Hawaii's International Conference on System Sciences, Hawaii.

10. CCITT (1991a). Message Handling Systems: Electronic Data Interchange Messaging System, Recommendation X435.
11. CCITT (1991b). Message Handling: Electronic Data Interchange Messaging Service, recommendation F435.
12. EPRI (2001). Auction Agents for the Electric Power Industry, Research Conducted for the Electric Power Research Industry, Palo Alto, CA.
13. Harkavy, M., Tygar, J.D., Kikuchi, H. (1998). "Electronic Auctions with private bids", Proc. Third USENIX Workshop on Electronic Commerce. Boston, Massachusetts, August 31-September 3, 1998.
14. Kaabi, S., Kaffel, H. and Kamoun, F. (2001). "Evaluation of HTTP, E-mail and NNTP with regard to Negotiation Requirements in the Context of Electronic Commerce." Proc. of ICECR-4, Dallas Texas, pp. 439-447.
15. Kaabi, S., Kaffel, H. and Kamoun, F. (2003). "Specification of a Communication Protocol for Real-time Auctions." Proc. ICECR6, Dallas, Texas, pp 129-138.
16. Klein, S. (1997). "Introduction to Electronic Auctions." EM-Electronic Auctions. EM-Electronic Markets, vol..7(4), pp. 3-6.
17. Kumar, M. and Feldman, S.I. (1998a). "Business Negotiations on the Internet." Technical Report IBM Institute for Advanced Commerce, Yorktown Heights.
18. Kumar, M. and Feldman, S.I. (1998b). "Internet auctions." Proceedings of the third USENIX Workshop on Electronic Commerce, Boston, Aug 31-Sept 3, pp. 49-59.
19. Maxemchuk N.F. and Shur, D.H. (2001). "An Internet multicast system for the stock market." ACM Transactions on Computer Systems, vol.19(3), pp.384-412.
20. Omote, K. (2002). "A study on electronic auctions." submitted for Japan Advanced Institute of Science and technology in partial fulfillment of the requirements for the degree of doctor of philosophy. March 2002.
21. Panzini, F., and Shrivastava, S.K. (1999). "On the provision of replicated Internet auction Services." Proceedings of the 18<sup>th</sup> symposium on Reliable Distributed Systems, Lausanne Oct. 19, pp. 390-395.
22. Peng, C.S., Pulido, J.M., Lin, K.J. and Blough, D.M. (1998). "The design of an Internet-based Real-Time Auction System.." Proceedings of the first IEEE workshop on dependable and real-time e-commerce systems (DARE-98), Denver Jun 98, pp.70-78.
23. Shih, Jansson and Drummon. (1997). *MIME-based Secure EDI*, draft-EDIINTint-as1-04.txt.
24. Ströbel, (2000). "On auctions as the negotiation paradigm of electronic markets." EM-Electronic Markets, vol. 10, No.1,
25. Turban, E. (1997). "Auctions and Bidding on the Internet: An Assessment" EM - Electronic Auctions. EM - Electronic Markets, Vol. 7, No. 4,
26. Wurman, W.Y., Wellman, M.P., and Walsh, W.E. (1998). "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents." Proceedings of the Second Intl. Conference on Autonomous Agents 98, Minneapolis MN USA, pp. 301-308.
27. IETF (2000a). IRC-ARCH, [www.ietf.org/rfc/rfc2810](http://www.ietf.org/rfc/rfc2810)
28. IETF (2000b). IRC-CHAN, [www.ietf.org/rfc/rfc2811](http://www.ietf.org/rfc/rfc2811)
29. IETF (2000c). IRC-Client, [www.ietf.org/rfc/rfc2812](http://www.ietf.org/rfc/rfc2812)
30. IETF (2000d). IRC-Server, [www.ietf.org/rfc/rfc2813](http://www.ietf.org/rfc/rfc2813)