Transaction Flow in Card Payment Systems Using Mobile Agents

O. R. Vincent, O. Folorunso, A. T. Akinwale, and A. D. Akinde
Computational Intelligence Group, Department of Computer Science, University of Agriculture, Abeokuta, Nigeria

vincent.rebecca@gmail.com; folorunsolusegun@yahoo.com; taakinwale@yahoo.com; adakinde@yahoo.com

Abstract

In this paper, a system for controlling transaction flow in card-based payment systems and its performance benefit using mobile agents is presented. The model consists of the negotiation semantics, the mobile agent module, and three interactive systems in e-payment: the bank, point-of-sale (POS), and the Interswitch. We present a mobile agent-based scenario for e-payment system and discuss the negotiation scheme and techniques that provide security for the model. Our scheme requires no interaction with the originator once the mobile agents are sent out. This is particularly of advantage in a situation when the user cannot stay online for a long period of time. Comparing our model with that of traditional Remote Procedure Call (RPC), the result shows that our scheme is more reliable and faster.

Keywords: Mobile Agents, E-commerce, Electronic Payment, Card Payment System and Transaction Delay

Introduction

In the past fifty to sixty years, the world has been experiencing a revolution in semi-conductor technology, digital techniques, and communication principles. Consequently, there exists a convergence between information technology and communications. This in turn has greatly impacted on virtually every aspect of human life, such as education, engineering, medicine, aviation, commerce, administration, domestics appliances, entertainment, and business. Furthermore, it has resulted in globalization, whereby distance has become irrelevant in human interactions and this has been credited mainly to computer networks.

To this end, Information and Communication Technology (ICT) has revolutionalized commercial transactions especially from the point of electronic connectivity, globalization, automation, remote control /monitoring /transactions and electronic payment. Over the years, electronic commerce has evolved into a popular and acknowledged way of conducting business (Folorunso, Sharma, Longe, & Lasaki, 2006). While researchers are still trying to understand it, e-commerce is growing incredibly, producing extraordinary results from both business and customer perspectives.

An electronic payment system involves the provision of payment services and transfers through such devices as tele-

Editor: Eli Cohen

phones, computers, the internet, ATMs, and smart cards (Wu, Osama, & Phu-Dung, 2006; Zoran, Ognjen, & Dragan, 2007). As a paperless system of making payment, it offers an alternative to the traditional systems, which involve the use of cash and check. Electronic payment systems have the advantage of enabling transactions to be processed quickly and more cheaply and also offer a much more convenient method of effecting settlement of transactions (Shintaro, 2003). The points of interaction include point-of sale (POS) terminals, automated teller machines (ATM), etc.

A point-of-sale (POS) terminal is a computerized replacement for a cash register. In a location where something is purchased, POS offers self-service for customer checkout. The POS system can include the ability to record and track customer orders, process credit and debit cards, connect to other systems in a network, and manage inventory. While an International Standard Organisation (ISO) deploys POS Terminals/ATM on behalf of the acquiring banks, a switching company enables the different banks to inter-operate and interconnect without having to incur the overhead cost of individually interconnecting. A financial transaction switching company used in this paper is the Interswitch Limited.

Though, e-commerce has boosted business transactions by making it easier and faster (Guan, Tan, & Hua, 2004), security is still the main issue in mobile agent based e-commerce applications. Security in mobile agent transaction is related to securing a runtime environment (like Java Virtual Machine) for a host, and host authentication before an agent moves to intended host. Another security problem is how to prove that the agent has not been tampered with over the channel which the agent can migrate.

There have been different approaches proposed to protect the platform, host, agent, and route (Westhoff, Schneider, Unger, & Kaderali, 1999). Recently, in order to secure the communication between agents on different hosts, Java Agent Development Framework (JADE) has enabled the Secure Socket Layer (SSL) protocol to provide confidentiality and integrity for all intra-platform connections (Bellifemine, Caire, Poggi, & Rimassa, 2003). Secure Electronic Transaction (SET) protocol was introduced to e-commerce because it satisfies the three criteria of security, scalability, and compatibility (Sheng, Tan, & Feng, 2004). The SET protocol supported by major corporations such as VISA Inc. MasterCard is an evolution of the existing credit-card based payment system which provides enhanced security for information transfer as well as authentication of transactions (Vincent & Akinde, 2007). Since a mobile agent may move to an insecure host or platform to communicate with other agents, the SSL channel may be unsecured for the mobile agent. It is very dangerous for mobile agents to carry sensitive (unencrypted) data (e.g., private key) through a sequence of machines, especially to an insecure host.

This paper, therefore, presents a transaction controlling model with a negotiation protocol for intelligent and mobile agents by providing secure communication channels in electronic payment systems with card transactions. The model consists of negotiation semantics, the mobile agent module, and three interactive systems in e-payment: the bank, POS and the Interswitch. We modified the negotiation model described in Muller (1996) and in Zeng and Sycara (1996) and adapted it as integration into operational transaction control architecture for mobile agents.

Literature Review

Currently, a common e-payment method involves the client transmitting to the merchant details of a payment card such as a VISA credit card (VISA, 2006). The merchant receives the information and proceeds to carry out a payment request with the card issuer via traditional card payment procedures. This system is simple and does not require the development of a new commercial infrastructure. However, it is susceptible to fraud from either transacting parties. The card information transmitted over the Internet could also be stolen by malicious parties. Many electronic payment schemes have been proposed but not all of them offer solutions to these problems.

Research on an Agent-based Bill Payment Service (ABPS) (Wong & Lau, 2000) was also conducted at Queensland University and was hosted on digitally certified website. In the pay service model, consumers must first register with ABPS by providing their personal information. To acquire services, customers authorize an ABPS payment agent to pay the related parties. In their system, the payment agent is responsible for obtaining settlement instructions and settling bills via appropriate financial institutes or external payment services. This is somewhat cumbersome and may be stressful to consumers. The heavy burden of managing an-ever-increasing knowledge base and growing load for the single payment agent server is a problem. In the proposed model, different agents will be used and embedded with functional modules as well as decision-making logic according to negotiation agreement.

Galbiati and Soramäki (2007) presented a competitive multi-agent model of inter-bank payment systems where banks choose their level of available funds on the basis of private payoff maximization. The model consists of the repetition of a simultaneous move stage game with incomplete information, monitoring, and stochastic payoffs. Using Bayesian theory to update with payoffs, they carried out numerical simulations to solve the model and investigated two special scenarios.

Hwang and Sung (2006) presented a study of micro-payments based on one-way hash chain and reviewed some literature on supporting multiple payments. Their proposed micro-payment achieved the following three goals: micro-payment multiple transactions, service providers, and anonymity. Keegan, O'Hare, and Grady (2008) designed a framework called Easi-Shop as an ambient intelligence to assist everyday shopping. In their work, they verified the different between Ambient Intelligent (AmI) and e-commerce by augmenting m-commerce with intelligent and autonomous components. The significance of the added value was realized for average shoppers as they wander their local shopping high street.

A more recent study by Malamati, Ioanna, and Lambros (2008) considered an intelligent agent's negotiation strategy in the electronic market place. In this work, dynamic multi-lateral negotiation strategy based on a ranking mechanism that does not require a complicated rationale on behalf of the buyer agents was proposed. The aim was to extend the strategy to functionality of autonomous intelligent agents, so as to maximize their owner's utility. After considering both contact and decision issues, it was discovered that the strategy was based on market conditions which could be employed by others. Alfi, Critella, Pietronero, and Zaccaria (2009) introduced a minimum agents based model for financial markets to understand the nature and self organization of the stylized facts They focused on four essential ingredients: fundamentalist agent which tend to stabilize the market; chartist agents which induce destabilization; analysis of the price behaviour for the two strategies; herding behaviour which governs the possibility of changing strategy. However, this model does not consider what happens in the market after the agents are dispatched.

Notice that in all the above literature and several others not mentioned here, none has integrated the four key players in e-payment, namely: the Acquirer, the Merchant, the Issuer, and the Cardholder, in the use of mobile agents' technology. In addition, there is no literature to the best of our knowledge that studies what happens at every stage of transaction process. The proposed research work focuses on transactions involving negotiation, process and payment involving these four key players

Security Issues

The following threats are identified with possible solutions:

(i) *Denial of Transaction*. A malicious host could try to make a fake good offer so that its offer would become the best thereby making the payment agents for each card holder *(P_aC)* reject other offers. This malicious host could later deny making the offer. To address this, every host must sign its own offer to enforce non- repudiation.

(ii) *Modification of other offer*. A malicious host could try to modify or delete offers from other hosts in order to present its offer as the best. This is addressed by sending as many mobile agents as possible to different hosts to be visited. In this case, every agent will visit one host; hence the malicious host will have no advantage of tampering with the data because the mobile agent does not have information of other hosts.

(iii) *Threat of the Private Key*. A malicious host could try to steal the private key of the mobile agent to sign arbitrary documents. To prevent this attack, the mobile agent is restricted in such a way that it could only spend up to a certain amount of money.

In general, we address the issue of security by sending multiple agents instead of one agent, which has great impact on valid transactions. Using multiple agents to conduct a transaction has the big advantage that the transaction can be valid even if some mobile agents have been tampered with. It is believed that all cannot be tampered with at the same time. This would certainly not be the case if only one mobile agent is used. The security advantage of using multiple agents is robustness. The use of multiple agents has, however, the disadvantage of creating a significant overhead, for instance, if many agents have to conduct only one transaction (O'Hare, 2007). This is certainly not very efficient at first sight. Fortunately, this overhead is compensated in our proposed model by conducting multiple transactions in parallel. Different transactions are conducted in parallel; for example, one or more agents can pay for electronic appliances and also transfer money into different accounts This means that a group of many agents can conduct an arbitrary number of transactions such that, in contrast to threshold schemes in traditional scenarios, all the overhead caused by the use of multiple agents is compensated.

In addition, the different hosts are also equipped with a certification authority to certify incoming agents as being valid or malicious. The authentication of the agent is checked by this authority before the agent is allowed to make transactions.

Security Challenges of Mobile Agents

In general, one can distinguish four different sorts of security problems when using mobile agents. Some problems are more difficult to solve than others (Jansen, & Karygiannis, 1999). The first challenge is to protect a mobile agent in transit. This is easy to accomplish with standard security techniques. By using the transport layer security protocol SSL/TLS (Dierks, & Rescorla, 2003), nobody (except the host that sends the mobile agent and the host that receives it) can eavesdrop on the mobile agent or tamper with it without this being detected. In the rest of the paper we assume that SSL/TLS is used every time agents or other information are sent between different hosts.

The second security problem is to protect a host against visiting mobile agents. This problem is similar to protecting a PC against viruses or malicious mobile code and should, therefore, not be very difficult to solve. Mobile agents are run in sandboxed environments. In addition, mobile agents can be digitally signed. An agent platform's security policy can then determine the privileges the platform will grant to the agent. A mobile agent also has to be protected against other mobile agents. This can be accomplished by making the mobile agent only accessible via a limited set of (harmless) public methods.

The last, and by far most difficult security problem, is to protect the mobile agent against the host where it is being executed. The mobile agent is completely under control of this host and almost any imaginable attack is possible (Fritz, 1998). The host can eavesdrop on the mobile agent and can tamper with the agent or just refuse to execute it, etc. An overview of possible solutions to this problem has been given in Cleassen, Bart, and Joos (2003).

Model Scenario

The user has a mobile agent platform with which he/she can periodically connect to the network for a relatively short time and disconnect. He wants to make several transactions with a bank card which includes: (i) check account balance, (ii) transfer money into different account, (iii) pay bills (electricity bill, water bill, and phone bill), (iv) make some requests, such as stop payment, and (v) buy some products like clothes and shoes. In addition, he wants to disengage with his present health insurance scheme and thereafter engage with another one. For this reason, he wants to stop paying the present one.

Mobile agents are software programs that can travel autonomously from host to host to perform one or more tasks on behalf of a certain user. They can communicate and even negotiate with other agents. The user can go offline after the agent is sent out and come online again when the agent has performed its task. A mobile agent can also perform its task locally (at the host) which enhances the sending device when it has limited computing power. In addition, a mobile agent becomes very promising in our case where many and different transactions will be conducted as soon as agents are sent. The agent can check account balance, pay bills, transfer money into another account, make requests, and buy many products. In this case, the agents need to move about and search for these products. This is a complex case of transaction interaction since it will require an initial phase of information gathering. They will have to negotiate over the possible prices for the products; hence a large volume of data is also expected.

In the Remote Procedure Call (RPC), communication entities have fixed and well-defined roles (Aderounmu, 2004). The server offers services and a client makes use of these services, which implies a strict sense of dependency. Clients are reliant upon servers to provide the service that they require. The communication mechanism that takes place between client-server models is through a message passing protocol. However, message passing of this form has been criticized for being too low level, requiring programmes to determine network addresses and synchronization points themselves (Ogwu, Talib, & Aderounmu, 2006; Vincent, Folorunso & Akinde, in press). Using the mobile agent will increase the fault tolerance and robustness of the POS network by reducing the dependency of the operations on the Interswitch. Mobile agents are considered as a direct extension of client-server technology.

E-Transaction using Mobile Agents

Electronic payment is classified into either card-based or network/software-based and into closed or open systems (Jailani, Yatim, Yahya, Patel, & Othman 2008). For the purpose of this study, our model is classified as a card-based closed system. We identified four essential players in card payment system (Hayashi, 2006; Li, Tu, Yen, & Xai, 2007): the card holder, the Issuer, the Merchant and the Acquirer, and design agents representing these players aimed at monitoring and controlling events at every stage of payment. The Merchant is the agent who sells goods and services to whom payment will be made. The card -carrying agent called the Cardholder has its own account with the Issuer while the Merchant has its own account with the Acquirer. The Acquirer is the agent that acquires a transaction while the Issuer is the agent that issues the card. The model assumes that the same agent can play the role of Issuer and Acquirer, so also Merchant and Cardholder. At the point of sales, we have the switch, in this case, the Interswitch, since the debit card uses Interswitch.

In the mobile agents, the modules (illustrated later in Figure 2) include the bank, the POS, the Interswitch, the negotiation, and the mobile agents modules. Mobile agents will be dispatched from the agent modules to remote e-payment hosts using the agent transport protocol. In this context, the following entities are involved:

S_aM	-	Selling agent(s) for Merchant
P_aC	-	Payment agent(s) for each Cardholder
R_aP	-	Residing agent(s) for the POS
R_aI	-	Residing agent(s) for the Interswitch
R_aB	-	Residing agent(s) for each Bank

From the mobile agent module, mobile P_aC agents are dispatched to migrate to different hosts and execute locally functions for search of products, negotiation, request and payment. Mobile S_aM agents are also dispatch to sell their products. In this model different cases of possible transactions in Figure 2 are discussed below.

Case 1: P_aC and S_aM move to a platform (the negotiation platform), meet together, and negotiate for product price. In this case the transaction includes the buying of clothes and shoes. The agent could further move to the bank to make payment to another account. A multiple transaction is considered where agents could make different transactions once dispatched.

Case 2: Mobile agent(s) move to the bank platform for transaction. The transaction includes balance inquiry, account transfer, stop payment request, and change Personal Identification Number (PIN). We assume a case where the mobile agent carrying bank card issued by bank *A* moved to bank *B* host for its transaction.

Case 3: Mobile agent(s) move to Point of Sale (POS) platform for transaction, which includes payment for products like shoes, clothes, and drinks with its card.

In case 2, if the card is from a bank outside that of the R_aB or Automated Teller Machine (ATM) bank's network, the transaction is forwarded to InterSwitch, which determines where the transaction should be forwarded to a base Batch Identification Number (BIN) on the card. The Interswitch role as a company is to facilitate inter-bank transactions while the Point of Sale (POS), a terminal at which cards are used for payment, facilitates the initiation of financial transaction.

As shown in Figure 1, all POS terminals are connected to the InterSwitch. This means that all transactions from the POS terminals hit InterSwitch from where they are sent to the front-end of the Issuing Bank. InterSwitch uses the BIN on the card to determine where to forward the transaction. The BIN is the number that is used to identify the batch number of a particular card. All POS on the InterSwitch network are connected to the front-end of the Banks. This front-end is a Mini-Switch that drives the POS. These front-ends are then connected to InterSwitch. If a card within the bank is used on the POS, the front-end uses the BIN retrieved from the card to determine if the card was issued from within its network or not. If it was issued within its network, it automatically sends the transaction to the banking host, which authorizes or denies the transaction.

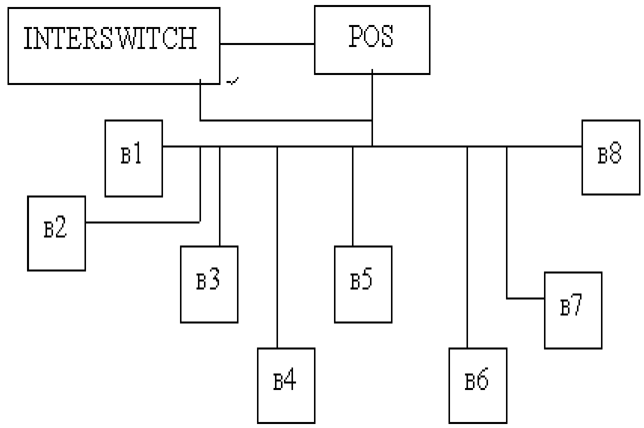

Figure 1: The Card-Payment Transaction Flow

Figure 2 illustrates how a mobile agent is used to process commercial transactions. The process consists of banking system *B*, point of sale (POS), and the Interswitch. The banking system *B* is a set of banks such that $B = \{b_1, ..., b_n\}$ where *n* is the number of banks connected to the Interswitch. As shown in Figure 2, $(B_{i's}(i = 1,...,8))$ is the banking system.

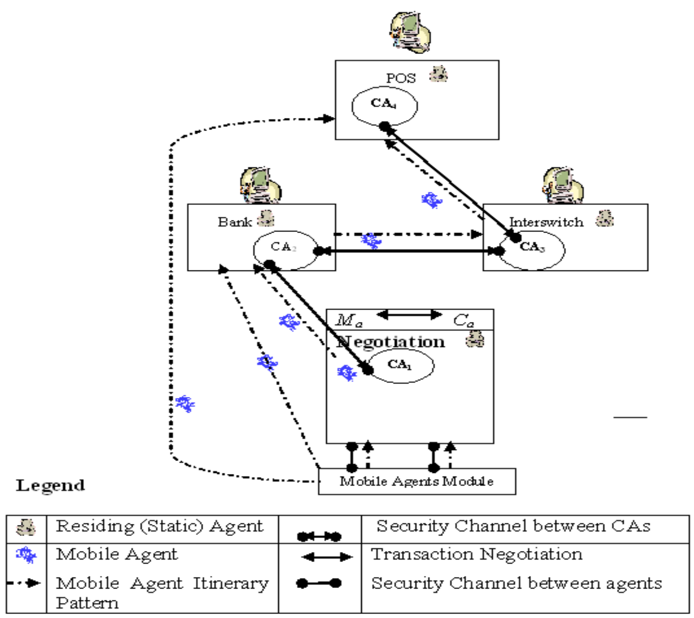

Figure 2: Multi-agents Interaction Model for Card-based Transaction

Basically, a switching company enables the different banks to interoperate and interconnect without having to incur the overhead cost of individually interconnecting, while an International Standard Organisation (ISO) deploys POS terminals/ATM on behalf of the acquiring banks. Examples of such financial transaction switching companies are Interswitch Limited, Cards Technology Limited (CTL)-in conjunction with MasterCard International e-Tranzaction and Valuecard. For this study, the Interswitch is used. The Interswitch is an electronic transaction switching and payment processing company. When a card holder goes to a merchant with a POS to transact business, money moves across the Interswitch for any inter-bank transaction. The model in Figure 2 is designed to control the transaction process among banks, the Interswitch, and the POS. It provides a negotiation channel where S_aM meets with the P_aC and negotiates for price of products. If the P_aC finds some negotiation that satisfies its attribute, it makes the decision to get the products. Each mobile agent platform has a verifier, called the Certification Authority (CA), to provide secure communication between the agents. Even the agent must register to get its certificate after setting up and must get their public key from the *CA* who signs all certifications for the whole platform in the system. The agents then store their private key in their home platform or in a security platform they trust. When an agent wants to move to another home, it clones itself, makes a representative mobile agent, signs, and sends. All agents communicate with each other through the Agent Communication Language (ACL) message while in RPC communication is done using communication messaging standard ISO 8583 message format.

The agent parameters and collected data are securely transferred between platforms, but the actual program code of the agent is retrieved each time from its original location. We assume a generic code that can be used for many users. At the end, the agent will return to the semi-trusted platform, a platform that verifies the authenticity of the transaction conducted.

The Negotiation

The negotiation model focuses on the Merchant and the Cardholder with the bank made up of the Issuer and the Acquirer. Negotiation in this case is a mechanism allowing autonomous intelligent mobile agents to move around the market place and request for any desired product where a mutual agreement on the product to buy and at what price is determined. We assumed a multi-products and multi-transactions in multi-agents environment for our scheme.

The negotiation in the context of our scheme is defined as follows:

Definition 1: Let D be a negotiation domain with 6-tuple (A, J, μ, M, U, P) where

- $A = \{a_1, ..., a_k), k \geq 4$ is a set of intelligent players of every a_i carrying electronic cash or product(s).
- $J = \{j_1, ..., j_n\}$, $n \leq k$ is the number of issues covered in a negotiation. These include product price, quality and merchant fee.
- μ is a function which assigns roles to buying and selling agents. $\mu(a) = j$, $\forall a \in A$
- for all players $a_i \in A$ is a set of possible agreement M_i such that $\forall a_i \in A$, $M_i \subset M$
- $U = (u_1, ..., u_k)$ where $u_i : A \rightarrow \Re$ is the utility function for agent a_i
- $P = (k, \pi, J \times K \rightarrow 2^k)$ is a negotiation protocol. K is a finite set of communication

primitives. The two primitives for this case are {start and done} $\subseteq K$. These primitives facilitate internal control system and are not communicated. π maps communication primitives into a subset of admissible reactions with respect to specific roles within the protocol. Each member of *P* is to make at least two transactions. Each component of transaction type is contained by the mobile agent platform. The transaction $(2^k)_{k=1,...,K}$ is terminated if there is no 2_{k+1} such that $(a^k)_{k=1,...K+1} \in P$. The scheduling attributes of a product include the time it would take to complete the transaction.

The minimum number of agents is four because all the four key players in e-payment are represented in the model: the card holder, the issuer, the merchant, and the acquirer. Once the set of issues has been determined over which agents negotiate, then the negotiation process of an alternate succession of contract proposals on behalf of the S_aM is made. In this case, the negotiation process is initiated by the P_aC (Buying Agent) who sends to S_aM an initial request for proposal. The issues to be negotiated in our case are: brand, quality, price and delivering methods.

Mathematical Description

We can describe our model mathematically as follows. Let the transaction request from the card-holder be represented by x and the reply from the Bank or POS by y. We assume that there are n requests per transaction with agent code m in byte. For any single transaction requested, the total amount of bandwidth usage in byte is given by

$$z = (x + y) \tag{1}$$

for client-server computing. If for each transaction, there are n number requests, the total bandwidth usage in bytes would be:

$$z = n(x + y) \tag{2}$$

Assume that $x = y$, then

$$z = 2xn \tag{3}$$

In the simplest case, where it is assumed that an agent can make a complete transaction by moving to and from between two nodes, that is, the mobile agent moves to one host to make a request, get its desired product(s), and goes back to the mobile agent platform. A simple case is a situation when a complete transaction is made when agent visits one host and gets back to the home host. For example, *PaC* moves to Bank *A* host with a card issued by Bank *A*, changes its Personal Identication Number (PIN), checks account balance, pays money into its account, and transfers money into another account. This is an example of our scenario in case 3, the size of the agent code would be $2m$, such that

$$n = \frac{1}{x}(2m + x) \tag{4}$$

From our model in Figure 2, if we consider a complex case 1, when agent has to move to more than one host before getting its desired output, the size of agent code is *3m*, and for case 2, it is *4m*. The agent in this case would transfer the entire request in the transaction as a batch to the R_aB or R_aP, execute the request and retrieve only the final result of the transaction back to the user. Using the above specified assumption, the total bytes transmitted over the network in executing the n-transaction request is given by

$$z = \sum_{i=1}^{N} m_i + xn + y \tag{5}$$

where the sum over m_i represents the total size of the agent code transferred from one node to another to make complete transaction. Equating (3) and (5), we have the number of transaction requests per service that will make the bandwidth usage in a client-server based system equal to that of agent based system:

$$n = \frac{1}{x}\left(\sum_{i=1}^{N} m_i + x\right) \tag{6}$$

The bandwidth usage in case 1 and 2 is not necessary as high in proportion to the host visited before making transaction as one could have expected. This is because all banks and POS are connected to the Interswitch, which has a specified bandwidth for all transactions for its network. Since all transactions are made in the same platform, the complexity of the transaction may not affect the bandwidth. Consider equation (3), where the bandwidth for simplest case is *2xn*, that is the agent visits one host, makes the transaction, and returns back to the user. For case 1 and 2 as seen in equation (5), m does not increase as in the simplest case.

Furthermore, we can calculate the transaction delay by representing the response of an agent by a state vector differential equation. This is due to the fact that the state vector describes the future response of a system, given the present state and the changes along the line. The state vector elements usually consist of the position of the separate bodies. Thus, it is an equation that expresses the relationship between the body and the state. In this case, we use a discrete-time approximation based on the division of the time axis into sufficiently small time increments. The successive time intervals $t = 0, T, 2T, 3T, \ldots,$ where T is the increment time it takes an agent to move from one node to another ($T = \Delta t$). That is, when a POS of a Bank *A* could not recognize a card at a glance, it sends it to the Interswitch, then there is always a time increment for getting a response. If the time increment T is sufficiently small compared with the time constant of the POS to process a transaction, the response evaluated by discrete-time methods will be accurate. The linear state vector differential equation of the transaction could then be written as

$$\chi = Ax + Bu \tag{7}$$

$$y = Cx + Du \tag{8}$$

The vector x is the state of the selling agent and u is the code that an agent carries. A is a constant $n \times n$ system matrix, B is a constant $n \times p$ output matrix. The vector y is the state of the buying agent, C is the constant $p \times n$ input matrix, and D is a constant $p \times m$ matrix. Since we are considering only a single insertion of cards into POS as single-input, the problem reduces to that of single-input single-output (SISO). Using basic definition of a derivative

$$\chi(t) = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - \Delta(t)}{\Delta t} \tag{9}$$

we can determine the value of $x(t)$ when t is divided into interval from POS to Interswitch and back to POS, $\Delta t = T$, thus approximating the derivative as

$$\chi = \frac{x(t + T) - x(t)}{T} \times 100 \tag{10}$$

Performance Evaluation

In this section, we carry out a performance evaluation of our model using the parameters presented in mathematical description. First, we examined the dependence of the bandwidth on the number of requests per transaction in client-server and mobile agent-based systems. We illustrate in Figure 3 the dependence of bandwidth z on the number of transaction requests n for a limiting case where we consider transaction carried out between two nodes. Figure 4 shows that z depends linearly on n, with an intercept on the positive z axis, which according to our model equation (5) is equal to x + 2m. This implies, if we have a constant transaction request x, the value of n increases with the size of the agent code m. Based on this, it is deduced that the smaller the agent code size, the faster the performance of the agent based system, which is an improvement to the RPC systems. The agent systems always perform better than RPC system since the least amount of transferable information over the network is a byte. In the bandwidth usage, our analysis shows that mobile agent approach is 72% faster than RPC.

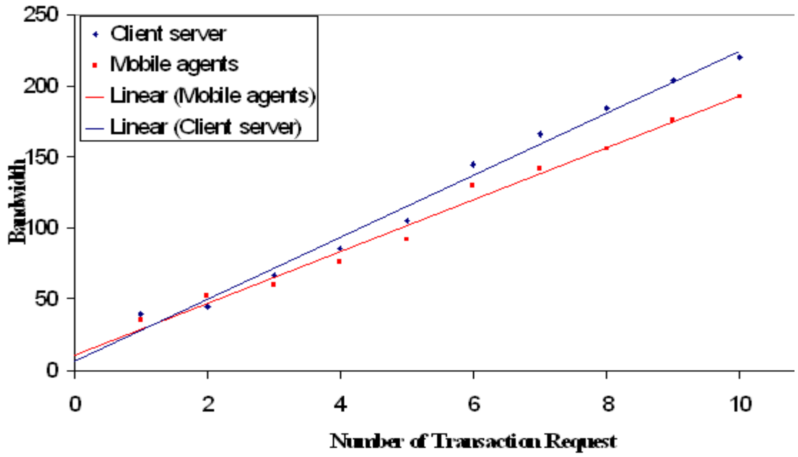

Figure 3: Bandwidth against Number of Transaction

In Figure 4, we show how the percentage of transaction delay varies with the number of transaction request. This measures the adaptive strength of the two schemes varying the rate of transaction delay. The delay rate state is simulated with a stochastic model, and the level of delay rate for each simulation was varied while measuring the percentage delay of transaction for the two different schemes. The delay rate of the RPC is 64.34 higher than the mobile agents. The result shown in Figure 4 depicts the advantage of mobile agent paradigm over the client server.

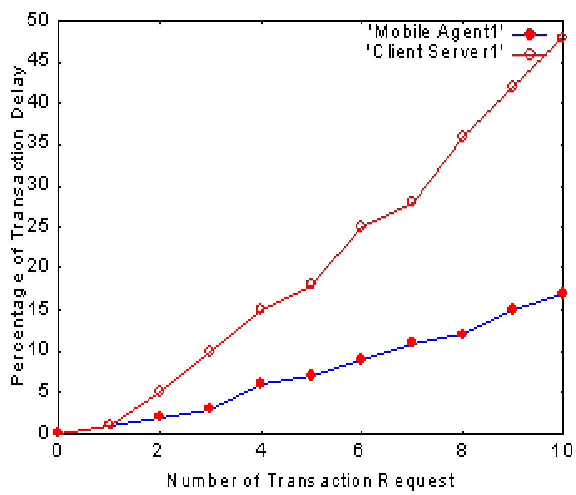

Figure 4: Percentage Transaction delay against Number of Transaction

Conclusions

Mobile agent-based card transaction is a promising scheme that brings transactional benefits to both the cardholders and merchants who accept cards. One of the benefits is reducing transactional costs; the network imposes no-surcharge rule, which prohibits merchants from charging a fee to cardholders unlike the case of RPC where merchants are assumed to set the same price for cash users and cardholders. In this paper, we have presented a transaction control model with a negotiation protocol for mobile and intelligent agents in electronic payment systems with card transactions and analyzed the performance of transaction denial and acceptance. The model consists of interactive systems such as bank, POS, Interswitch, the negotiation semantics, and the mobile agent module called the controller. Possible means of making negotiations locally from agents-centered perspective are also described. The implication of Interswitch connection in this regard is that the POS does not need to send any request of inquiry to the Interswitch; the Interswitch alerts the POS once there is any transaction to be conducted.

Furthermore, we describe some mathematical explanation for agent acceptance and denial and compare our model with that of RPC. The simulation shows proofs of the superior schemes provided by mobile agents over the traditional client-server computing in electronic transactions.

Acknowledgement

The authors would like to acknowledge the useful comments of Prof. Dr. Jüergen Dix of Technical University of Clausthal, Germany and discussions with Dr. G. A. Aderounmu of the Obafemi Awolowo University, Nigeria.

References

Aderounmu, G. A. (2004). Performance comparison of remote procedure calling and mobile agent approach to control and data transfer in distributed computing environment. *Journal of Network and Computer Applications, 27*, 113-129.

Alfi, V., Critella, M., Pietronero, L. & Zaccaria, A. (2009). Minimum agent based model for financial markets I: Origin and self-organization of stylized facts. *European Physics Journal, 67*, 385-397.

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (1999). JADE : A white paper. *EXP, 3*(3), 6-19. Retrieved from http://jade.tilab.com/papers/WhitePaperJADEEXP.pdf

Cleassen, J., Bart, P., & Joos, V. (2003). How can mobile agents do secure electronic transactions on untrusted hosts? – A survey of the security issues and the current solutions. *ACM Transactions on Internet Technology, 3*(1), 28–48.

Dierks, T., & Rescorla, E. (2003). *The TLS Protocol Version 1.1.* IETF Internet Draft, March 2003.

Folorunso, O., Sharma, S. K., Longe, H. O. D., & Lasaki, K. (2006). An agent-based model for agriculture e-commerce system. *Information Technology Journal, 5*(2), 230-234.

Fritz, H. (1998). A model of attacks of malicious hosts against mobile agents. *Proceedings of the 4th ECOOP Workshop on Mobile Object Systems: Secure Internet Mobile Computation, July 1998*.

Galbiati, M., & Soramäki, K. (2007). A competitive multi-agents method of interbank payment systems. Proceedings of the *Joint Bank of England/European Central Bank Conference on 'Payments and monetary and financial stability'.*

Guan, S., Tan, S. L., & Hua, F. (2004). A modularized electronic payment system for agent-based e- commerce. *Journal of Research and Practice in Information Technology, 36*(2) 67-87.

Hayashi, F. (2006). A puzzle of card payment pricing: Why are merchants still accepting card payments, *Review of Network Economics, 5*(1), 144-174.

Hwang, M. S. & Sung, P. C. (2006). A study of micro-payment based on one-way hash chain. *International Journal of Network Security, 2*(2), 81-90.

Jailani, N., Yatim, N. F. M., Yahya, Y., Patel, A. & Othman, M. (2008). Secure and auditable agent-based e-marketplace framework for mobile users. *Computer Standard and Interfaces, 30*, 237-252.

Jansen, W., & Karygiannis, T. (1999). Mobile agent security. NIST Special Publication 800-19, October.

Keegan S., O'Hare, G. M. P., & O'Grady, M. J. (2008). Easishop: Ambient intelligence assists everyday shopping. *Information Sciences, 178*, 588-611.

Li, P., Tu, M., Yen, I-L., Xia, Z. (2007). Preference update for e-commerce applications: Model, language, and processing. *Electron Commerce Research, 7*(1), 17-44.

Malamati L., Ioanna R., & Lambros P. (2008). An intelligent agent negotiation strategy in the electronic marketplace environment. *European Journal of Operational Research, 187*, 1327-1345.

Muller, J. P. (1996). A cooperation model for autonomous agents. *Proceedings of The Third Workshop on Agent Theories, Architectures, and Languages (ATAL' 96),* Budapest, Hungary, 245-260.

O'Hare, G. (2007). *Mobile agents: Agile agents for ambient intelligence*. Lecture notes presented in School of Computer Science and Information, University College Dublin.

Ogwu, F. J., Talib M., & Aderounmu, G. A. (2006). An analytical survey of mobile agent for resource management in a network. *Journal of Statistics and Management Systems, 9*(2), 427-439.

Sheng, V., Tan, S. L., & Feng, H. (2004). A modularised electtronic payment system for agent-based e-commerce. *Journal of Research and Practice in Information Technology, 36*(2) 67-87.

Shintaro, K. (2003). New perspectives on m-commerce research. *Journal of Electronic Commerce Research, 6*(3) 160-164.

Vincent, O. R., & Akinde, A. D. (2007). Building mechanisms of trust in securing electronic payment system. *Proceedings of the National Conference on Advanced Data Computing Communications and Security, Kadi, India (2007)*, 233-237.

Vincent, O. R., Folorunso, O. & Akinde, A. D. (in press). Improving e-payment security using elliptic curve cryptosystem. *Journal of Electronic Commerce Research*.

VISA. (2006). *3-D secure protocol specification – Extensions for mobile internet devices, international.*

Westhoff, D., Schneider, M., Unger, C., & Kaderali, F. (1999). Methods for protecting a mobile agent's route. *Proceedings of International Information Security Workshop '99 (ISW'99),* LNCS 1729, 57-71.

Wong, O., & Lau, R. (2006). Possibilitic reasoning for intelligent payment agents. *Proceedings of the 2nd Workshop on AI in Electronic Commerce AIEC*, 170-180.

Wu, X., Osama, D., & Phu-Dung, L. (2006). The design and implementation of a smartphone payment system based on limited-used key generation scheme. *Proceedings of the 3rd IEEE International Conference on Information Technology*, New York, 62-28.

Zeng, D. & Sycara, K. (1996). How can an agent learn to negotiate? *Proceedings of The Third Workshop on Agent Theories, Architectures, and Languages (ATAL' 96),* Budapest, Hungary, 233-244.

Zoran, D., Ognjen, M. & Dragan, G. (2007). Internet payment system: A new payment system for internet transactions. *Journal of Universal Computer Science, 13*(4), 479-503.

Biographies

Rebecca Vincent is a Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta. She obtained a B.Sc degree in Mathematical Sciences (Computer Science Option) and M.Sc in Computer Science from the University of Agriculture, Abeokuta in 2000 and 2005 respectively. She is currently rounding up her Ph.D research on Mobile Agents for E-Commerce. Her research interest include: Images and Vision, Knowledge Management, Computational Complexity, E-commerce, Agents and Mobile Agents. She is a member of Nigeria Computer Society and has published in notable International and local Journals.

Dr. Olusegun Folorunso is a Senior Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta. He obtained a B.Sc degree in Mathematical Sciences from the University of Agriculture, Abeokuta in 1992, M.Sc in Computer Science from University of Lagos in 1997 and a Ph.D in Computer Science in 2003 from the university of Agriculture, Abeokuta. His research interest includes Adoption of Information Systems strategies, Human Computer Interactions (HCI) , Knowledge Management, Image Processing and Computational Intelligence . He is a member of Nigeria Computer Society and Computer Professional Registration Council of Nigeria. He has published in reputable international and local Journals.

(Biographies not available for A. T. Akinwale and A. D. Akinde)