

Mobile Payment Architectures for India

Deepti Kumar^{*}, Timothy A Gonsalves[†], Ashok Jhunjhunwala[‡] and Gaurav Raina[§]

^{*} [†] [‡] The Telecommunications and Computer Networking Group, IIT Madras

[§] Department of Computer Science and Engineering, IIT Madras

Email: ^{*} deepti@lantina.tenet.res.in, {[†] [tag](mailto:tag@tenet.res.in),[‡] [ashok](mailto:ashok@tenet.res.in)}@tenet.res.in, [§] gaurav@cse.iitm.ac.in

Abstract—Mobile payments are a new and alternative payment method. Instead of using traditional methods like cash, cheque, or credit cards, a customer can use a mobile phone to transfer money or to pay for goods and services.

Mobile payments have numerous advantages over traditional payment methods. Apart from their apparent flexibility, they enable consumers who do not have easy access to banking facilities to participate readily in financial transactions. Unfortunately, existing mobile payment solutions in India are not interoperable; i.e. they only offer services for merchants registered with them and do not allow the transfer of money to, or between, users of other payment providers. This limitation reduces the widespread adoption of mobile payments.

In this paper, we propose new mobile payment architectures that support interoperability. A key technical aspect of the mobile payment process is to lookup customer details, for which we propose the following three design options: (1) a central database, (2) a peer-to-peer query, or (3) a hierarchical lookup. These options are evaluated using relevant metrics such as the complexity of implementation and scalability with respect to system size. Based on our evaluation we recommend that initially the peer-to-peer design is chosen, and once the technology is more widespread, the central database option should be adopted.

I. INTRODUCTION

India has recently experienced phenomenal growth in the adoption of mobile phones. The subscriber base has reached 350 million with a growth of approximately 10 million subscribers a month [1]. Mobile phones compensate for inadequate infrastructure, such as slow postal services and the dearth of rural banks, and thus allow information to move more freely. Infact, according to the World Bank, mobile phones have a direct impact on economic growth: an extra 10 phones per 100 people in a typical developing country boosts GDP growth by 0.8 percentage points [2]. With mobile phones now so widespread, we are presented with an opportunity to usher in mobile payments.

Currently, mobile payment services are provided to subscribers who are customers of some predefined Telecommunication Service Providers (TSPs) and banks. For example, if a mobile payment provider has a relationship with a particular TSP and a bank then it can provide payment services to only those customers who have an account in that bank. The existing framework, thus, does not allow *interoperability* which is defined as establishing a common framework for processes and methods that enable movement of funds across accounts held by two customers participating in mobile payments [3].

The current mobile payment framework in India allows users to shop from only those merchants which are registered with their payment providers. Transactions from one

unregistered customer to another customer (C2C), or from a business to a customer (B2C), or between businesses (B2B) is not currently supported. This restricts *universality* of mobile payments.

For mobile payments to become widespread as a mode of payment, the conditions of simplicity and usability, universality, interoperability, security, privacy and trust, cost, speed and cross border payments have to be satisfied [4]. Issues related to interoperability and universality have not, up to now, been addressed in India due to the lack of an appropriate regulatory framework. To that end, the Reserve Bank of India (RBI) finalized a set of regulations which were ratified in September 2008. Regulations of the RBI state that only banks which are licensed and supervised in India and have a physical presence in India will be permitted to offer mobile banking services and these services shall be restricted to customers of banks and holders of debit/credit cards [3].

In this paper, we develop architectures for mobile payments which provide interoperability and universality while ensuring simplicity. For *simplicity* it is important to minimize the input from the user; for example, in order to make a payment the user should be required only to enter the recipient's mobile number and the amount to be transferred. This makes the mobile payment process simple to use on the limited display and keyboard of most mobile phones. This provision of making payments to un-registered users, which currently does not exist, is an important factor in ensuring universality, and is a key aspect of the new architectures. The proposed architectures are compatible with the regulatory regime in India.

The scope of this paper is limited to markets where only banks are authorized to offer mobile payment services; for example, India. The rest of this paper is as follows. In Section II we outline some background information related to mobile payments. Section III details a real time scenario and the new architectures for mobile payments. An important process, during a mobile payment, is to lookup customer details for which three design options are outlined. Comparison and analysis of the proposed design options is given in Section IV. We conclude in Section V, and in Section VI we outline avenues for future work.

II. BACKGROUND

Mobile payment solutions may be classified according to the type of financial rules and regulations followed in a country. There are three types of mobile payment markets [4]: (1)

Highly regulated markets, where only the banks are entitled to offer mobile payment services. The bank account is linked to the mobile phone number of the customer, and when the customer makes a payment the bank account of the customer is debited; for example, in India. (2) *Moderately regulated markets*, which allow consolidated accounts where private companies can operate mobile payment services as licensed money transmitters; for example, in the USA. (3) *Minimally regulated markets*, which allow Telecommunication Service Providers (TSPs) to handle the subscribers' cash with a TSP account, and allow the TSP to accept and disburse cash from its outlets; for example, in Kenya.

A mobile phone can send and receive information over various channels. Typically three possible channels are used for sending or receiving information on a GSM mobile phone [5]: (1) Short Message Service (SMS), (2) Unstructured Supplementary Services Delivery (USSD), (3) WAP/GPRS. Security for SMS can be achieved by encrypting the messages before sending them. GPRS messages can be made secure by the use of SSL. It is difficult to achieve secure communication using USSD, therefore USSD should be used only for sending alerts and non-financial information.

The messages which carry inter-bank transaction and financial information are sent through a TCP channel using a standardized message format. In India two messaging standards, the ISO 8583 [6] and the Structured Financial Messaging System (SFMS) [7], have been proposed for use in the mobile payment process. SFMS which is based on SWIFT and ISO 7775 is used by Indian banks for settlements. ISO 8583 is used by Indian banks for payments and real-time transactions; for example, ATM transactions and credit card payments. Both formats are similar in terms of the fields and the implementation difficulty and can be used for transactions in mobile payments.

III. ARCHITECTURES FOR MOBILE PAYMENTS

In this section we propose mobile payment architectures for highly regulated markets where payments are carried out through bank accounts. The designs support interoperability, universality and simplicity which are essential for successful mobile payments.

Mobile payments involve the implementation of functions which are outside the traditional purview of the banks. For this purpose we define a logical entity called the Mobile Payment Provider (MPP) which facilitates financial transactions between mobile phone customers. A bank may decide to outsource the MPP function to a third party, or it can be its own MPP.

A. Outline of the Mobile Payment Process

We now provide an outline of the mobile payment process and of the proposed architecture.

A *Customer* is an account holder in a Bank B1 and is linked with a Telecom Service Provider T1 and an MPP M1. A *Beneficiary* is an account holder in a Bank B2, is linked with a Telecom Service Provider T2 and an MPP M2. See Figure

1 for the logical flow of a typical mobile payment transaction. The Customer may wish to pay the Beneficiary some amount of money. The payment can be initiated by the Customer or by the Beneficiary and is called the *Push process* or the *Pull process* of payment, respectively. The party who initiates the payment is called the initiating party and the other party is referred to as the 2nd party.

We now provide an outline of the Push process of payment. To begin with, the Customer opens the mobile payment application of M1 on his phone and enters the phone number of the Beneficiary and the amount to be paid. A message containing the phone number and the payment amount is sent to the Customer's MPP M1. M1 must then, using the Beneficiary's mobile phone number, find out the Beneficiary's MPP M2; this is done using either of the design options proposed in subsection III-B. The payment request is then sent to M2 where the Beneficiary's account and other details are stored. M2 obtains authorization from the Beneficiary and the transaction is routed to the Customer's MPP. M1 informs the Customer's Bank B1 about all the payment details and an inter-bank transfer of funds is initiated. Recall that in the Pull process, it is the Beneficiary that initiates the payment.

In general, users may have multiple bank accounts in multiple banks and may also be linked with multiple MPPs. To begin with we analyze the scenario where a customer has one MPP and one bank account; see subsection III-B. The case of multiple bank accounts and multiple MPPs is described later in subsection III-C.

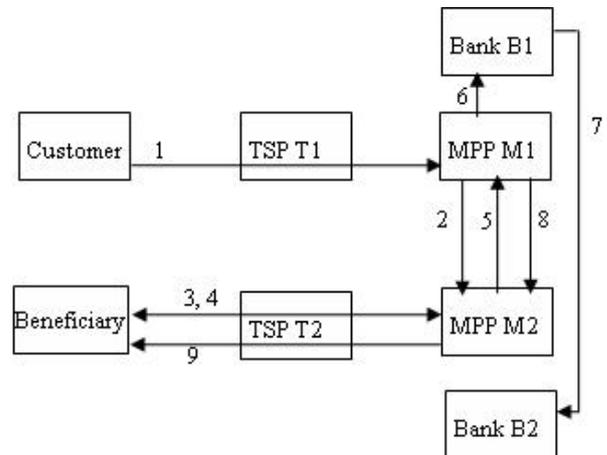


Figure 1. End-to-end logical flow of a mobile payment transaction

B. Proposed Design Options

A key issue in the above process is that, given a mobile phone number, M1 should be able to locate the MPP corresponding to it. In this subsection we propose three possible options to perform the lookup on the 2nd party's (M2's, in the above case) MPP.

We outline three design options which address the issues of interoperability, universality and simplicity of use. The construction of the design options facilitates interoperability

and universality. Simplicity is achieved as the initiating party has only to enter numbers, not text.

We propose, and subsequently analyze, three options: (1) a central database, (2) a peer-to-peer query, and (3) a hierarchical lookup. Our initial assumption is that each user registers only one account with one MPP.

1) *Central Database (CDB)*: Customer enters the Beneficiary's phone number and the amount to be paid on his mobile phone. The message is carried by T1 to M1. At M1, the Customer's account details are fetched. The query is then routed to a Central Database which keeps a list of all mobile phone numbers registered for mobile payments and the corresponding MPPs for each one. The central database is a separate database maintained by RBI or its authorized agency.

Each time an MPP gets a request to transfer payment to another phone number, this database is queried and the MPP information of the particular phone number is retrieved. In this case M2's information, such as its MPP ID, is obtained from the database and sent to M1. M1 then routes the request to M2 from where the Beneficiary is informed about the transaction and his account details are retrieved. The collected information is sent to M1 which informs its Bank B1 and an inter-bank funds transfer happens between B1 and B2 and both parties are notified.

2) *Peer-to-Peer (P2P)*: In this option, each MPP maintains a local database containing the <phone number, MPP ID> pair of all its customers. When the Customer enters the Beneficiary's phone number and the amount on his mobile phone, the message reaches M1 via T1. The initiating MPP, M1, sends a broadcast query to all other MPPs in the system to find the MPP corresponding to the Beneficiary's mobile phone number. The other MPPs check in their local databases if the mobile phone number belongs to them. The MPP which owns that particular mobile phone number responds positively. M1 now has information that M2 is the MPP of the Beneficiary. M1 sends a payment request to M2 from where the transaction proceeds as described in the Central Database option outlined above.

Once the Customer's MPP, i.e. M1, has got information about the MPP corresponding to the Beneficiary's phone number, it will remain in M1's cache. In case a payment request arrives for the same phone number, the result is fetched from the cache and the broadcast query is avoided. An entry will remain in the cache until it is refreshed. When the cache is full, an algorithm, like the Least Recently Used, will choose which items to discard to make space for the new ones.

To avoid a broadcast to all MPPs a staggered broadcast query can be used. The rules to govern the staggered broadcast query could be based on the population of each MPP, or the MPPs historical statistics of replying.

A modified solution to evade the problem of issuing a broadcast is to replicate the copy of a combined database at every MPP. Each time the database at any of the MPPs is changed the modification is broadcast to all other MPPs and synchrony is maintained. This solution discourages smaller players from entering the market, as even the smallest MPP has

to maintain resources for accommodating information about all users.

3) *Hierarchical*: In this case, the Telecom Service Provider of the mobile payment user maintains a database containing the MPP for each mobile payment user. The Customer's message containing Beneficiary's mobile phone number and amount reaches its MPP M1. The MPP locates the Beneficiary's MPP by sending the mobile phone number to Customer's TSP T1, where the telephone number mapping is done and the TSP of the Beneficiary is resolved by hierarchically looking at the phone number [8]. The TSP information is sent to M1 from where a request is sent to Beneficiary's TSP T2. At T2, the Beneficiary's MPP is located from the database and the payment request is routed to it, after which it takes the same route as described in the Central Database option outlined previously. This option requires that the TSPs modify their existing databases to accommodate the MPP information.

C. Multiple MPPs and Multiple Accounts

In this subsection we generalize our options to allow one user to have multiple account numbers and MPPs. If a user of mobile payments has multiple accounts then he will have to designate one account number as default. The bank to which the account number belongs is linked to one or more MPPs. Out of these MPPs the user marks any one of them as a default MPP. This default MPP will be the one that will be looked up when the initiating party's MPP searches for the MPP of the 2nd party (as described in the subsection above). Registering this default account will give the user the convenience of making a mobile payment by using only the mobile number. It also reduces the number of user interventions required and makes payments simple.

If the initiating party does not wish to use the default account number then he will have the choice of selecting the account number from the list made available on his mobile phone by his MPP. This method provides the flexibility of using a different account number each time at the cost of ease of use. If the 2nd party does not wish to use a default account an extra element of user intervention is required to determine the 2nd party's account number.

IV. COMPARISON OF THE PROPOSED DESIGN OPTIONS

We now examine, in some detail, the three proposed designs. Analysis is done on the basis of scalability with respect to the subscriber and MPP populations, unfairness, complexity, server capability and storage space required. Calculations are made for a wide range of values for the number of subscribers, MPPs and the number of transactions per customer. The assumptions stated below consider an initial rollout to widespread and global use of mobile payments.

- 1) We analyze the problem at four different levels of subscriber population: 10,000, 1 million, 100 million and 6 billion.
- 2) The number of MPPs have been varied as 5, 10, 50, 100 with the respective increase in population of subscribers as mentioned in step (1) above.

- 3) The data that is sent as a part of the lookup message consists of a 10 digit phone number and a 6 digit MPP ID which is encapsulated in a TCP/IP packet resulting in a total length of approximately 60 bytes.
- 4) The maximum number of transactions per customer happens in the peak 10 hours and are increased in the following order for subscriber populations in (1) above: 1, 5, 10, 20.
- 5) In case of P2P, the number of MPPs is assumed to be 5, 10, 50, 100 for the subscriber populations in step (1) above. All MPPs have an equal share of subscriber base. The case where MPPs have unequal numbers of subscribers is also analyzed.

We employ a simple analytical framework. Let the total subscriber population in the system be p and the number of MPPs be m . The average message size is d Bytes and the number of transactions in a period of h hours is n .

The rate of requests/second at the lookup server is

$$r = \frac{p}{m} \times n \times \frac{m-1}{3600 \times h}$$

The bandwidth required is

$$r \times d$$

The lookups per second to be performed are

$$r \times \log_2 \left(\frac{p}{m} \right)$$

Using these expressions, we compare the three designs along several dimensions.

1) *Scalability with Respect to Subscriber Population:* As the number of subscribers increases the load at each MPP increases. We therefore evaluate the number of requests per second that have to be served by the lookup server in each case. The bandwidth requirements are also measured with the increase in load.

The number of requests per second that the server has to serve is approximately the same in both the CDB and the P2P solutions as shown in Table I. This is because in P2P each MPP gets a broadcast from other MPPs for most of the requests, compared to all the requests in the case of CDB.

TABLE I
REQUESTS PER SECOND ARRIVING AT AN MPP

Subscriber Population	10 k	1 m	100 m	6 b
Number of MPPs	5	10	50	100
CDB (req/sec)	0.278	138.8	27,777	3,333,333
P2P (req/sec)	0.22	125	27,222	3,300,000
Hierarchical (req/sec)	0.05	13.9	555.5	33,333

In both cases as the number of subscribers increases from 1 million to 100 million the load on the lookup server increase from 130 and 27,000 requests per second. This indicates that the bandwidth requirements are 27Kbps and 5.5Mbps, respectively. In the Hierarchical solution the requirement is as low as 100Kbps for 100 million customers. We conclude that bandwidth is not an issue in the implementation of any of the options.

2) *Scalability with Respect to MPP Population:* Scalability is calculated in terms of the extra messages that the lookup server will have to process when the number of MPPs in the system increase.

In P2P, the traffic of the query messages increases considerably as the number of MPPs grows. When all MPPs have the same population, the probability of broadcasting a request rises with the addition in the number of MPPs. This thereby increases the probability that an MPP receives a *broadcast miss*, i.e. a request which is not meant for the MPP. An increase in the probability of broadcast miss causes a reduction in the percentage of *broadcast hits*, i.e. a request not meant from the MPP. This implies that as the subscriber population and the number of MPPs in the system increases, there will be a larger number of redundant messages. Hence, the P2P solution is not scalable when the number of MPPs in the system increases thereby increasing the probability of broadcast miss.

The reader is referred to Table II for the probability of broadcast miss at an MPP.

TABLE II
BROADCAST MISS AT AN MPP IN PEER-TO-PEER

Number of MPPs	5	10	50	100
Probability of a broadcast miss at an MPP	0.48	0.72	0.95	0.97

The CDB and Hierarchical solutions remain unaffected with the addition in the number of MPPs.

3) *Unfairness:* Unfairness is defined in terms of the *overhead* which is the ratio of broadcast miss to broadcast hit. It is the deviation of the overhead from the ideal condition where all MPPs are of equal sizes.

In the CDB and the Hierarchical solutions each MPP receives only requests intended for its account holders. There is no unfairness. With the P2P design, unfairness occurs in the system when MPPs are unequally divided. An MPP suffers from unfairness if it receives more than its proportionate share of messages. Our analysis shows that if all MPPs are of similar size, or there is only one dominant MPP, unfairness is low. However, if there are two or more MPPs with a large customer base and all others are small, unfairness increases significantly.

For example, with 5 MPPs, if MPP 1 and MPP 2 have a customer share of 35% and all the others have 10% each, then all the smaller MPPs have to serve broadcast misses sent by the larger MPPs. Thus, even a small MPP has to be equipped with a powerful server in order to process the large number of broadcast misses.

4) *Complexity:* The architectures are evaluated in terms of the complexity of implementation and operation.

The CDB solution requires an organization to set up and maintain the database. Given the large geographical spread and large subscriber base in India, the CDB server becomes a single point of failure and a bottleneck. To avoid these problems, the database can be replicated in different regions of the country. This increases the complexity of registration and updating of the account and MPP information.

This cost, however, is insignificant in the P2P design where each MPP sets up its own local database and there is no synchronization issue or the requirement of setting up a central body.

The hierarchical solution requires every TSP to maintain a database with <phone number, MPP ID> values. These are unrelated to the TSP's primary business and it might not want to make changes with its database without significant benefits. Moreover, owing to differing processes and regulations, the information provided by the TSPs may not be trusted by the banks. Thus the hierarchical option is ruled out in India's highly regulated market.

5) *Server Processing Capacity*: As lookup is the most frequent operation, server capacity is defined by the number of lookups the server is able to perform per second.

For 1 million subscribers, the CDB server will have to perform approximately 2768 lookups per second. This is only 1.3 times faster than that required for a P2P server. Given the modest size of the database on the order of 100 MB, it is feasible to store the entire database in RAM. Using binary search in RAM, each lookup will take less than 1 ms on a 3 GHz machine. Thus an inexpensive 3 GHz quad-core processor will suffice for the load.

In the hierarchical option the server performs lookups at 230 records per second for 1 million subscribers, which is easily achievable on the type of servers that TSPs deploy for call routing.

6) *Storage*: The central database requires 2GB of storage space in case of 100 million customers. Since the number is modest the entire database can fit into the DBMS or OS cache, and slow disk activity can be minimized.

In case of P2P the storage space needed is 40 MB which is the same as in the hierarchical option. Although the storage space required in the CDB option is 50 times more than in any other solution, storage space is not an issue of concern in any of the architectures.

A. Summary of our Evaluation

CDB Option: From the above analysis we conclude that the CDB is complex in terms of initial setup, maintenance and requires replicated high-end servers with fast lookup speeds. At 1 million subscriber base, 130 requests per second hitting the lookup server is a fairly high number, and requires extremely high-end computational equipment. Therefore, the CDB option should be adopted when the subscriber population reaches 750,000 with approximately 6 MPPs in the system.

P2P Option: This solution has almost no initial complexity and has low maintenance for a low customer base of 10,000 - 100,000. For a customer base of approximately 1 million, it requires that even small MPPs have high speed servers to do lookups. It is to be noted that for an unequally populated system, broadcast misses at a smaller MPP increase where there are two or more large MPPs, thereby causing unfairness. This is a good option to start with; essentially, till the system size reaches a certain size.

Hierarchical Option: This solution is conceptually simple but it does not seem to be a practical solution in the current regulatory environment.

For the current market needs in India, the P2P solution is best suited for fast initial deployment of mobile payments. As the use of mobile payments picks up, it is recommended that a central organization be established and the system migrate to a replicated CDB.

V. CONCLUSION

We propose several architectures for mobile payments that support interoperability, universality and simplicity in the context of highly regulated markets where the customer's mobile phone number is linked to a bank account number.

For the mechanism to lookup customer details, during the mobile payment process, three options were proposed: (1) a central database, (2) a peer-to-peer query, (3) a hierarchical lookup. Our quantitative and qualitative evaluation shows that initially, among the options considered, the most suitable design for the Indian markets is the peer-to-peer design. This option involves the least complexity and setup time in the current situation where only a small number of subscribers use mobile payments. As the system size increases, the central database option should become the preferred solution.

The Interoperability Standard for Mobile Payments in India has been accepted by the Mobile Payment Forum of India (MPFI) [9] and development is under way for trial deployment by several banks.

VI. FUTURE WORK

The next natural step towards enabling widespread mobile payments in India is to implement the proposed standards and carry out a thorough performance evaluation. To that effect, a testbed for conformance testing of the mobile payment servers is currently being established at IIT Madras.

VII. ACKNOWLEDGEMENTS

We are grateful to members of the MPFI Technology Committee for their efforts in formulating the Interoperability Standard for Mobile Payments.

REFERENCES

- [1] *Subscriber Figures for October 2009*, Cellular Operators Association of India. Available: <http://www.coai.com/statistics.php>.
- [2] *The power of mobile money*, issue September, 2009. The Economist.
- [3] *Mobile Banking transactions in India - Operative Guidelines for Banks*, issued Oct. 2008. Reserve Bank of India. Available: <http://www.rbi.org.in/Scripts/>
- [4] Carr, M., *Framework for Mobile Payment Systems in India*, Advanced E-Business Methods. Edited By: Milena M. Head Eldon Y. Li, Information Science Reference, 2009.
- [5] Mckitterick, D., and Dowling, J., *State of the art review of mobile payment technology*. Technical report, Jun. 13 2003.
- [6] *International Organization for Standardization, —Financial transaction card originated messages — Interchange message specifications —. Part 1: Messages, data elements and code values*. Reference Number: ISO 8583-1:2003(E). Edition 1. Switzerland, Oct. 21, 2008.
- [7] Institute for Development and Research in Banking Technology. Available: <http://www.idrft.ac.in/>.
- [8] *National Numbering Plan*, April 2003. Department of Telecommunications, India.
- [9] Mobile Payment Forum of India. Available: <http://www.mpf.org.in/>.