Reducing Computational Resource Requirements for Making Payments

Summary of the Invention

[0001] The present invention relates to methods and systems for performing payments from senders to recipients, and in particular to methods and systems which permit more efficient usage of computational resources.

Background of the invention

[0002] Peer-to-peer payment (P2P payment, also called person-to-person payment) is an online technology that allows a "sender" to transfer funds from a payment account associated with the sender, to a payment account associated with a payment "recipient" via the Internet or a mobile phone. The sender may for example be a customer of a recipient who is a merchant. Typically, neither the sender nor the recipient has to trust the other with information about his or her payment account.

[0003] There are two conventional methods for initiating a P2P payment. In the first method, the sender and payee establish secure accounts with a trusted third-party vendor (typically, a financial institution), designating their respective bank account or credit card information to be used to transfer and accept funds. Using the third party's website or a mobile application, the sender and recipient can complete the process of sending or receiving funds. The sender can only send funds to someone who is a member of the network.

[0004] In the second method, the sender uses an online interface or mobile application (typically provided by his or her bank or another financial institution) to contact a third party site. To initiate the transfer, the sender indicates the amount of funds to be transferred, and provides contact data for the payee (an email address or phone number). Once the transfer has been initiated by the sender, the recipient receives a notification using the contact data. The recipient uses the online interface or mobile application to input his or her payment account information (e.g. the number of a payment account held by a bank, and details of the bank) to accept the transfer of funds. In this second method, recipients do not need to have a payment account with the same financial institution as the sender in order to receive a money transfer.

[0005] As P2P traffic grows, the computational infrastructure required to implement it is becoming increasingly expensive and complex.

Summary of the invention

[0006] The present invention aims to provide new and useful systems and methods for making a payment which address the problems of conventional systems and methods noted above.

[0007] In general terms, the invention proposes that a P2P system computationally analyses historical data relating to P2P traffic to predict periods of high usage, and, based on the predicted high usage periods, computationally interacts with payment senders, to schedule payments to avoid the predicted high usage periods.

[0008] The effect of this is to smooth the traffic over time, thereby reducing the number of transfers which are required at peak traffic periods. In this way, the computational expenses of operating the P2P process can be reduced, since the peak level of P2P traffic, which determines the computational resources which have to be devoted to the P2P traffic, can be reduced.

[0009] The operator of the P2P process provides incentives ("rewards") to payment senders who agree to their corresponding payments being performed at time other than the predicted high usage periods. Thus, these payment senders receive a benefit.

[0010] Furthermore, other payment senders who insist on their payment being performed at a high usage periods, may benefit, since if the volume of traffic at those periods is reduced, the quality of service of the payment process may be improved (e.g. the risk of the payment system becoming overloaded is reduced). Thus, payment senders who co-operate with the proposed payment scheduling, and payment senders who do not participate, both benefit. This in turn leads to greater levels of satisfaction by both types of payment sender, which would be expected to lead to improved business for the payment recipients.

[0011] The scheduling process comprises defining a plurality of time-slots (which may be thought of as "buckets") during a single day. For each time-slot, a corresponding estimate is made of the volume of payments during the time-slot (i.e. the estimated workload of the payment engine during that time-slot). The scheduling process then recommends to senders that the payment is scheduled during a time-slot other than the time-slot with the highest volume of payments.

[0012] For at least one of the time-slots, a corresponding incentive ("reward") is defined. If this is done for a plurality of the time-slots, those time-slots having lower workload estimates may have correspondingly higher rewards. During the interaction with the senders, the senders are offered the option for the money to be transferred to the recipient during one of the time-slots for which an reward has been defined, and may be informed about the amount

of the reward associated with each of those time-slots. The sender may choose one of these time slots. When a time-slot choice is received from the senders, the payment to the recipient's payment account is scheduled during the chosen time-slot.

[0013] The sender is given the reward corresponding to the chosen time-slot. The reward may, for example, be in the form of a discount to a standard service charge for making the payment (if so, the reward will typically be applied when the payment is debited to the sender's payment account, which may be earlier than the chosen time slot). Alternatively, the reward may be one or more reward points of a rewards program, or a cash-back payment (e.g. which may be used to cover some or all of the cost of making a future peer-to-peer payment using the system). In the latter cases, the reward may be made when or after the transfer is made to the recipient's account.

[0014] The invention may be expressed as a method carried out by the server. Alternatively, it may be expressed as the server itself, or as a network of communicating devices (a "computerized network") comprising the server. It may further be expressed in terms of a software application downloadable into a computer device to facilitate the method. The software application is a computer program product, which may be stored in non-transitory form on a tangible data-storage device (such as a storage device of a server, or one within a communication device).

Brief description of the drawings

[0015] An embodiment of the invention will now be described for the sake of example only with reference to the following drawings, in which:

Fig. 1 shows schematically a computerized network of electronic devices for performing a method which is a first embodiment of the invention, the network including servers and communication devices;

Fig. 2 shows the logical structure of the software of a P2P server of the system of Fig. 1;

Fig. 3 shows a method of performing a payment transaction by the computerized network of Fig. 1;

Fig. 4 shows a method performed by a module of the software of Fig. 2;

Fig. 5 is a block diagram of the technical architecture of the servers of Fig. 1; and

Fig. 6 is a block diagram of the communication devices of Figs. 1.

Detailed description of the embodiments

[0016] Referring to Fig. 1, a computerized network of devices is shown which cooperate together to perform a method according to the invention. A first communication device 1 is associated with a person who wishes to make a payment (the payment "sender"). A second communication device 2 is associated with the person (the payment "recipient") to whom the sender wishes to make the payment. For simplicity the communication devices for just one sender and one recipient are shown, but it is to be understood that in reality there may be many senders who at any time desire to send payments to corresponding recipients.

[0017] The communication devices 1, 2 (which are described below in more detail with reference to Fig. 6) include a respective screen 1b, 2b and keys 1c, 2c. The screen 1b may be touch sensitive, in which case the screen 1b and keys 1c together provide a user interface of the communication device 1. Both the communication devices 1, 2 are able to communicate with a communication network 3 using respective communication interfaces (not shown) of the communication devices 1, 2.

[0018] In Fig. 1, the communication devices 1, 2 are depicted as smartphones, however they may be any electronic communication device, such as another mobile device (e.g. a tablet computer) or a personal computer (PC). The communication devices 1, 2 may communicate with the communication network 3 via a wireless connection (e.g. 3G, 4G, wifi or Bluetooth) or a wired connection.

[0019] The sender associated with the communication device 1 maintains a payment account (e.g. a bank account or a credit card account) at a financial institution (e.g. bank), referred to here as the issuer, and associated with an issuer server 4. The recipient associated with the communication device 2 maintains a payment account at a financial institution (e.g. bank), referred to here as the acquirer, and associated with an acquirer server 5. The issuer server 4 and acquirer server 6 are controlled by a P2P server 6 to make payments between the payment accounts they hold. The communication devices 1, 2 can communicate with the P2P server 6 using the communication network 3.

[0020] Fig. 2 shows the logical structure of the software running on the P2P server 6. It includes a component 61 for interfacing with the sender and recipient communication devices 1, 2: a payment transfer API 62 for performing a P2P payment; and a scheduler component 63. The scheduler component has access to a database 64 of data describing historical P2P payment transactions. The scheduler component 63 comprises two modules:

a reward calculation module 63a for calculating rewards associated with each of a number of time-slots, and a scheduled payment performance module 63b for performing payments according to a schedule determined by corresponding senders. The scheduled payment performance module 63b employs a database 65 of scheduled payments, as explained below.

[0021] The function of these three components will now be explained with reference to Fig. 3, which shows the overall method 100 carried out by the computerized network of Fig. 1 to perform a single payment operation, and Fig. 4 which shows the process 200 carried out by the reward calculation module 63a of the scheduler component 63.

[0022] Referring firstly to Fig. 3, steps 101 and 102 are the same as in some conventional P2P payment methods.

[0023] In step 101, the sender uses the communication device 1 to communicate with the P2P payment server 6. This may be done by the sender using a standard browser running on the communication device 1, or it may be done using a software application, such as one previously downloaded from the payment server 6 or the issuer 4.

[0024] In either case, a graphical user interface (GUI) is presented to the sender, under the control of the component 61. If the sender is using a standard browser running on the communication device 1, the component 61 provides instructions (for example in HTML or XML) to control the GUI seen by the sender. If the sender is using a dedicated software application running on the communication device 1, the component 61 interfaces with and controls the software application. The component 61 may be implemented in the same way as a corresponding component of a conventional P2P server.

[0025] In step 102, the sender enters into the GUI a payment amount, recipient data, and payment account data relating to the sender's account at the issuer.

[0026] The recipient data may be recipient contact data, using which the P2P server 6 can contact the recipient (see steps 105, 110 below). Alternatively, the recipient data may be data which allows the P2P server 6 to access a database of previously stored recipient data. For example, the P2P server 6 may store, for each of a number of previous senders, a database of all the recipients to whom the sender has previously sent money, and the GUI may present those options to the sender. In this case, the recipient data may be just a selection of one of the options.

[0027] The payment account data is data sufficient to identify the issuer (e.g. an issuer name and/or an issuer identity number), and sufficient to identify to (at least to the issuer server 4) the identity of the sender. For example, the payment account data may be a PAN (payment account number) number relating to a payment card issued by the issuer (optionally together with card expiry data, card verification value (CVC) number, etc). As used in this document, the term "payment card" refers to any cashless payment device associated with a payment account, such as a credit card, a debit card, a prepaid card, a charge card, a membership card, a promotional card, a frequent flyer card, an identification card, a prepaid card, a gift card, and/or any other device that may hold payment account information, such as mobile phones, Smartphones, personal digital assistants (PDAs), key fobs, transponder devices, NFC-enabled devices, and/or computers.

[0028] In another alternative, the payment account data may be a name or identity number of the issuer, and the number of the sender's payment account.

[0029] In a further alternative, the payment account data may be a name or identity number of the issuer, plus a token issued by the issuer using which the issuer is able to access a database of client records to identify the sender's payment account.

[0030] In step 103, the interface component 61 uses the GUI to ask the sender if the transfer is an urgent one.

[0031] If, in step 104, the sender answers positively ("Yes"), then in step 105 the interface component 61 obtains recipient payment account data identifying the payment account of the recipient. Specifically, if in step 102, the sender entered contact data for the recipient, then the interface component 61 may perform step 105 by using the contact data to send a notification to the recipient that a payment is being sent to him or her (optionally telling him or her also the amount). The contact data may be associated with the communication device 2 (e.g. a phone number of the communication device 2, or an email address which the communication device is programmed to access). The recipient may then use the communication device 2 to contact the interface component 61 either via a browser or a dedicated software application running on the communication device 2. In either case, the recipient sees a GUI controlled by the interface component 61. Into it the recipient may enter recipient payment account data which is sufficient to identify the acquirer to the server 6, and sufficient to identify (at least to the acquirer server) 5 the payment account of the recipient. Again, this may be a PAN number; a bank identifier name and/or identity

number, and a bank account number; or a bank identifier name and/or identity number, and a token which will be recognised by the acquirer.

[0032] In a variation of the embodiment, if the sender has previously made a payment to the same recipient, the interface component 61 may have previously stored the recipient account data in a database assessable to the interface component 61. In this case, step 105 may be performed by obtaining the recipient account data from the database.

[0033] In step 106, the interface component 61 activates the payment transfer API 62 to send the payment, by communicating with the issuer server 4 and acquirer server 5. The exact flow of messages may vary from one implementation to another. In one case, the payment transfer API 62 may contact the acquirer server 5, which in turn contacts the issuer server 4 to request the payment. Alternatively, the payment transfer API 62 may contact the issuer server 4 to instruct it to make the payment to the acquirer.

[0034] In either case, the issuer server performs various checks. These include an authorization operation (e.g. to check that sufficient funds are available in the sender's payment account), and optionally also a security operation (e.g. contacting the sender by SMS to the communication device 1, to obtain confirmation from the sender that the payment is intended). If either the security operation or the authorization operation are unsuccessful the method terminates, and the payment transfer API 62 controls the interface component 61 to send a notification of this to the sender and/or the recipient. Alternatively, if the authorization operation and security operation are successful, the issuer server 5 arranges a payment to the acquirer. This may just be to schedule a payment in the future, e.g. as part of a clearing operation performed periodically. Without waiting for the clearing operation, the issuer server 4 debits the payment account of the sender (for the payment amount, plus optionally a sender service charge), and the acquirer server 5 credits the payment account of the recipient (for the payment amount, minus optionally a recipient service charge) (step 107). The payment from the sender's payment account to the recipient's payment account has now been completed (even though the clearing operation is still in the future).

[0035] The communication between the issuer server 4 and the acquirer server 5 may be a payment network (such as the one operated by MasterCard Inc.). Optionally, the payment server 6 may be operated by the payment network, and in this case the P2P server 6 may have the dual role of communicating with the sender and recipient, and coordinating the communication between the issuer and acquirer.

[0036] Alternatively, if in step 104 the sender answers the payment is not urgent ("No"), then the interface component 61 activates a scheduler component 63 which is not present in a conventional P2P system.

[0037] In step 108 the scheduler component 63 controls the interface component 61 to present to the sender a number of time-slots in the future, and for each time-slot a corresponding reward for making the transfer into the recipient's payment account during the time-slot. A reward may, for example, be in the form of a reduced service charge for the transfer. Alternatively or additionally, it may be in the form of loyalty points or a cash-back. The cash-back may be credited to an account held by the P2P server 6 and associated with the sender, rather than being credited to the sender's payment account with the issue. The sender may be able to use the cash-back to cover some or all of the cost of instructing future payments using the P2P server 6, thus incentivising the sender to use the P2P server 6 again.

[0038] The calculation of these rewards is performed by the reward calculation module 63a, and is explained below with reference to Fig. 4. For each time-slot presented to the sender, the reward may be high if the reward calculation module 63a estimates that the proportion of the current day's payments which will be in that time-slot is low.

[0039] Note that if at least part of the reward associated with a given time-slot is a reduction of the service charge (compared to the service charge which would be made if the payment is made urgently in step 106), then the display of the reward could be simply by displaying the resultant reduced service charge for each corresponding time-slot.

[0040] Note that the components 61 and 63 may not supply the sender with information about all time-slots with the immediate future. For example, there would be little value in presenting information about future time slots for which the expected level of payment traffic is higher than the current level (since it would make no sense to incentivize the sender to use those future time slots for the transfer). Optionally, the sender may only be given information about one or more of the future time slots, such as the ones having the lowest level of expected traffic. These future time slots are ones associated with rewards set by the reward calculation module 63a.

[0041] In step 109, the sender chooses one of the time-slots. Thus, if the sender choses a time-slot for which it is estimated that a low proportion of the day's traffic will be in that time-slot, the sender will (in a later step of the method) be given a correspondingly high reward.

Note that if only one time–slot is presented to the sender, the sender's confirmation that the presented time-slot is acceptable is equivalent to a choice of that time-slot. The sender may be given an option of rejecting all the presented time-slot(s), in which case the method jumps to step 106, and the payment is performed at once.

[0042] In step 110, the scheduled payment performance module 63b of the scheduler component 63 registers the selection, and schedules the payment into the recipient's payment account to be made during the chosen time-slot. It does this by entering a scheduled payment into the database 65. At this stage the scheduler component 63 may control to the interface component 61 to perform an operation identical to that of step 106, of obtaining the recipient payment account data (e.g. by contacting the recipient using the contact data supplied in step 102).

[0043] In step 111, the scheduled payment performance module 63b accesses the database 65, and checks whether the chosen time-slot has arrived. If not, the scheduled payment performance module 63b waits until the next time-slot, and performs the same check again. When the result of the check is positive, the method proceeds to step 112.

[0044] In step 112, the scheduled payment performance module 63b instructs the payment transfer API 62 to perform the transfer in a way identical to step 106. In step 113, the payment account of the sender is debited, and the payment account of the recipient is credited, in the same way as in step 107. The sender is rewarded or charged according to the reward corresponding to the choice the sender made in step 109 (i.e., the reward which was displayed in step 108 in association with the chosen time-slot). The method then ends.

[0045] As noted above, method 100 shows what happens for each payment handled by the P2P server 6. It is to be understood that the P2P server 6 will typically be simultaneously performing method 100 for multiple payments, each having a corresponding server and recipient. In each time-slot, the scheduled payment performance module 63b will typically have to perform a large number of the scheduled payments, as well as a number of urgent ones.

[0046] Turning to Fig. 4, the process 200 carried out by the reward calculation module 63a is explained. Conveniently, the process 200 may be initiated at intervals, for example once each 24 hour day, but the intervals may be longer or shorter. While the method 200 is performed once, the method 100 will typically be performed many times: once in respect of

each payment which is received during the day. In other words, the process of method 200 is running in parallel with multiple instances of the method 100.

[0047] In step 201, the reward calculation module 63a extracts historical data from the database 64. In step 202, the reward calculation module 63a applies historical trend analysis to predict the number of transactions to be performed during a certain future time period. This will be the time period during which the scheduler component 63 schedules payment transfers during this performance of the method 200.

[0048] Conveniently, the future time period may also be a 24 hour day, but it may be longer or shorter than this. The trend analysis may for example be based on the current day of the week. The predicted number of transactions may, for example, be the average of the number of transactions on the corresponding day of the previous one or more weeks. Optionally, the predicted number of transactions may be corrected to take into account any trends in the data, such as a seasonal trend (for example, in the weeks before Christmas), or an annual trend (e.g. the rising number of people who participate in P2P transactions each year). Thus, the trend analysis may take into account any one or more of yearly data (e.g. number of transactions per year), monthly data (e.g. number of transactions in corresponding month of previous years) and daily data (e.g. number of transactions in previous days).

[0049] The future time period is divided into a plurality of time-slots (which may also be thought of as "buckets"). For the sake of simplicity, these time-slots are assumed below to be of the same length, but in some embodiments of the invention, this is not the case. In step 203, the scheduler forms an initial estimate of the number of transactions to be performed in each of the time-slots. In one example, these estimates may just be equal to the estimated total number of transactions during the future time period, divided by the number of time-slots in the future time period. However, depending what historical data is available in the database 64, more sophisticated estimates are possible. For example, if the historical data indicates that payments are more likely at certain times of day, that fact can be taken into account in the initial estimate of the number of transactions to be performed in each of the time-slots.

[0050] If the future time period is longer than the periodicity with which the method 200 is carried out, then when the method 200 is initiated, the database 65 will already contain payments scheduled to occur during the future time period. In step 204, for each time-slot,

the estimated number of payments in the respective time-slot is updated by adding to it number of scheduled payments in that time-slot.

[0051] In step 205, using the updated estimated number of payments in each corresponding time-slot, the reward calculation module 63a estimates the respective proportion of the current day's payments which will be in that time-slot.

[0052] In step 206, using the respective estimated proportions of the current day's payments in each of the time-slots, the reward calculation module 63a assigns a respective reward to at least one of the time-slots.

[0053] For each such time-slot, the reward may be higher according to how low is the corresponding estimated proportion of the current day's payments within that time-slot. There are many ways in which this could be done. For example, there could be a number of predefined rewards, and the predefined rewards might have a predetermined ranking. Thus, the reward calculation module 63a could assign the predefined rewards to the time-slots (starting with the time-slot having the lowest predicted estimated proportion of the current day's payments) such that the order of the ranking of the predefined rewards was equal to the order of the estimated proportion of the number of payments in each of the time-slots.

[0054] Alternatively, the reward calculation module 63a could identify any of the future time slots for which the expected number of requests is lower than the current number of requests by more than a certain threshold, an assign an equal reward to each of those identified future time slots.

[0055] Note that if for all the future time slots the expected level of requests is higher than the current level of requests, the method may pass at once to step 105, so as to perform the transfer immediately.

[0056] Typically, during any given time-slot the P2P server will receive multiple payment requests, each having a corresponding sender and receiver. For each payment request, a respective instance of the method 100 would be performed. In each instance of step 108, the components 61 and 63 would present the corresponding sender with at least one of the possible time-slots, and, for each presented time-slot, display the corresponding reward which the reward calculation module 63a last assigned to that time-slot.

[0057] Once a certain number of payments have been scheduled in different instances of step 110, the risk reward calculation module 63a could be triggered (step 208 of process 200) to

update the estimated number of transactions in each of the timeslots. Then the method could return to step 205, and perform steps 205 and 206 again.

[0058] Fig. 5 is a block diagram showing a technical architecture of the P2P server 6. The issuer server 4 or acquirer server 5 may also have this technical architecture.

[0059] The technical architecture includes a processor 222 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 224 (such as disk drives), read only memory (ROM) 226, random access memory (RAM) 228. The processor 222 may be implemented as one or more CPU chips. The technical architecture may further comprise input/output (I/O) devices 230, and network connectivity devices 232.

[0060] The secondary storage 224 is typically comprised of one or more disk drives or tape drives and is used for non-volatile storage of data and as an over-flow data storage device if RAM 228 is not large enough to hold all working data. Secondary storage 224 may be used to store programs which are loaded into RAM 228 when such programs are selected for execution.

[0061] In this embodiment, the secondary storage 224 has an order processing component 224a comprising non-transitory instructions operative by the processor 222 to perform various operations of the method of the present disclosure. The ROM 226 is used to store instructions and perhaps data which are read during program execution. The secondary storage 224, the RAM 228, and/or the ROM 226 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media.

[0062] I/O devices 230 may include printers, video monitors, liquid crystal displays (LCDs), plasma displays, touch screen displays, keyboards, keypads, switches, dials, mice, track balls, voice recognizers, card readers, paper tape readers, or other well-known input devices.

[0063] The network connectivity devices 232 may take the form of modems, modem banks, Ethernet cards, universal serial bus (USB) interface cards, serial interfaces, token ring cards, fiber distributed data interface (FDDI) cards, wireless local area network (WLAN) cards, radio transceiver cards that promote radio communications using protocols such as code division multiple access (CDMA), global system for mobile communications (GSM), long-term evolution (LTE), worldwide interoperability for microwave access (WiMAX), near field communications (NFC), radio frequency identity (RFID), and/or other air interface protocol radio transceiver cards, and other well-known network devices. These network

connectivity devices 232 may enable the processor 222 to communicate with the Internet or one or more intranets. With such a network connection, it is contemplated that the processor 222 might receive information from the network, or might output information to the network in the course of performing the above-described method operations. Such information, which is often represented as a sequence of instructions to be executed using processor 222, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave.

[0064] The processor 222 executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk based systems may all be considered secondary storage 224), flash drive, ROM 226, RAM 228, or the network connectivity devices 232. While only one processor 222 is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors.

[0065] Although the technical architecture is described with reference to a computer, it should be appreciated that the technical architecture may be formed by two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the technical architecture 220 to provide the functionality of a number of servers that is not directly bound to the number of computers in the technical architecture 220. In an embodiment, the functionality disclosed above may be provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third party provider.

[0066] It is understood that by programming and/or loading executable instructions onto the technical architecture, at least one of the CPU 222, the RAM 228, and the ROM 226 are changed, transforming the technical architecture in part into a specific purpose machine or apparatus having the novel functionality taught by the present disclosure. It is fundamental

to the electrical engineering and software engineering arts that functionality that can be implemented by loading executable software into a computer can be converted to a hardware implementation by well-known design rules.

[0067] Fig. 6 is a block diagram showing a technical architecture of any one the communication devices 1, 2.

[0068] The technical architecture includes a processor 322 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 324 (such as disk drives or memory cards), read only memory (ROM) 326, random access memory (RAM) 328. The processor 322 may be implemented as one or more CPU chips. The technical architecture further comprises input/output (I/O) devices 330, and network connectivity devices 332.

[0069] The I/O devices comprise a user interface (UI) 330a, a camera 330b and a geolocation module 330c. The UI 330a may comprise a touch screen, keyboard, keypad or other known input device. The camera 330b allows a user to capture images and save the captured images in electronic form. The geolocation module 330c is operable to determine the geolocation of the communication device using signals from, for example global positioning system (GPS) satellites.

[0070] The secondary storage 324 is typically comprised of a memory card or other storage device and is used for non-volatile storage of data and as an over-flow data storage device if RAM 328 is not large enough to hold all working data. Secondary storage 324 may be used to store programs which are loaded into RAM 328 when such programs are selected for execution.

[0071] In this embodiment, the secondary storage 324 has an order generation component 324a, comprising non-transitory instructions operative by the processor 322 to perform various operations of the method of the present disclosure. The ROM 326 is used to store instructions and perhaps data which are read during program execution. The secondary storage 324, the RAM 328, and/or the ROM 326 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media.

[0072] The network connectivity devices 332 may take the form of modems, modem banks, Ethernet cards, universal serial bus (USB) interface cards, serial interfaces, token ring cards, fiber distributed data interface (FDDI) cards, wireless local area network (WLAN) cards, radio transceiver cards that promote radio communications using protocols such as code

division multiple access (CDMA), global system for mobile communications (GSM), long-term evolution (LTE), worldwide interoperability for microwave access (WiMAX), near field communications (NFC), radio frequency identity (RFID), and/or other air interface protocol radio transceiver cards, and other well-known network devices. These network connectivity devices 332 may enable the processor 322 to communicate with the Internet or one or more intranets. With such a network connection, it is contemplated that the processor 322 might receive information from the network, or might output information to the network in the course of performing the above-described method operations. Such information, which is often represented as a sequence of instructions to be executed using processor 322, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave.

[0073] The processor 322 executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk based systems may all be considered secondary storage 324), flash drive, ROM 326, RAM 328, or the network connectivity devices 332. While only one processor 322 is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors.

[0074] Whilst the foregoing description has described exemplary embodiments, it will be understood by those skilled in the art that many variations of the embodiment can be made within the scope and spirit of the present invention.

[0075] For example, although as described above, the payment transfer API 62 is only activated in step 106 or 112, and implements the entire payment process at that time, it is alternatively possible in the case that answer in step 104 was "no" for the API to be activated earlier (for example, following step 110) to instruct the issuer server 4 to debit the payment sum (plus any service charge) to the payment account of the sender. That is, during the time-slot chosen by the sender, the API 63 would have only to arrange for the payment to be transmitted to the recipient's payment account (and optionally for any additional reward to be sent to the sender).

Claims

1. A computer-implemented method performed by a server, for performing payments between payment accounts, the method comprising:

(i) analysing a database of data describing previous payments, to obtain, for each of a plurality of future time-slots a corresponding estimate of the number of payments the server will have to perform in each time-slot,

(ii) based on said estimates of the number of payments the server will have to perform in each time-slot, assigning to at least one of the plurality of time-slots a corresponding incentive;

(iii) receiving from a payment sender a request to make a payment from a first payment account maintained by an issuer and associated with the payment sender, to a payment recipient, said payment recipient being associated with a second payment account maintained by an acquirer;

(iv) presenting to the payment sender the at least one time-slot to which an incentive was assigned;

(v) receiving from the payment sender a selection of one of the presented time-slots; and

(vi) communicating with at least one of the issuer and acquirer, to cause the payment to the second payment account within the selected time-slot.

2. A computer implemented method according to claim 1 further comprising, in said step (iv), for each presented time-slot presenting the corresponding incentive.

3. A computer implemented method according to claim 1 or claim 2, further including:

upon said step (iii) of receiving said request, transmitting to the payment sender a query of whether the payment is urgent, and receiving from the payment sender a response to the query; and

if the response to the query is positive, omitting steps (iv)-(vi), and communicating with at least one of the issuer and acquirer, to cause the payment to the second payment account substantially immediately.

4. A method according to claim 1, claim 2 or claim 3 further including, upon said step (v) of receiving from the payment sender a selection of one of the time-slots, updating a scheduled payment database of scheduled future payments,

said step (vi) being performed by, in each of successive time-slots, determining using said scheduled payment database whether the present time-slot is the time-slot in which the payment is scheduled, and if the determination is positive, to communicate with at least one of the issuer and acquirer, to instruct the payment.

5. A method according to any preceding claim in which, for at least one time-slot, said corresponding incentive is a reward, the reward being higher if the estimated number of payments the server will have to perform in the corresponding time-slot is lower.

6. A method according to any preceding claim in which said step (i) comprises:

using said data describing previous payments to estimate, for a time period consisting of said time-slots, a corresponding estimate of the number of payments the server will have to perform during the time period, and

deriving said estimate of the number of payments the server will have to perform during each time-slot by partitioning among the time-slots the estimated number of payments the computer will have to perform during the time period.

7. A method according to claim 6 in which the data describing previous payments comprises at least one of:

(a) data indicative of a daily trend in the number of payments the server is requested to perform;

(b) data indicative of a monthly trend in the number of payments the server is requested to perform; and

(c) data indicative of an annual trend in the number of payments the server is requested to perform.

8. A method according to any preceding claim further including, after step (v):

increasing the estimated number of payments the server will have to perform in the selected time-slot; and

for at least one of the time-slots, updating the corresponding incentive.

9. A method according to any preceding claim in which the request contains:

payment account data comprising data sufficient to identify the issuer, and data sufficient to identify, at least to the issuer, the payment account of the payment sender; and

contact data of the payment recipient;

the method further comprising contacting the payment recipient using the contact data, and receiving, from the payment recipient, payment account data comprising data sufficient to identify the acquirer, and data sufficient to identify, at least to the acquirer, the payment account of the payment recipient.

10. A server for transmitting to financial institutions messages instructing payments between payment accounts, the server comprising a processor and a data storage device storing program instructions, the program instructions being operative upon being run by the processor, to cause the processor:

(i) to analyse a database of data describing previous payments, to obtain, for each of a plurality of future time-slots a corresponding estimate of the number of payments the server will have to perform in each time-slot,

(ii) based on said estimates of the number of payments the server will have to perform in each time-slot, to assign to at least one of the time-slots a corresponding incentive;

(iii) to receive from a communication device associated with a payment sender a request to make a payment from a first payment account maintained by an issuer and associated with the payment sender, to a payment recipient, said payment recipient being associated with a second payment account maintained by an acquirer;

(iv) to present to the payment sender the at least one time-slot to which the incentive was assigned;

(v) to receive from the payment sender a selection of one of the presented time-slots; and

(vi) to communicate with at least one of the issuer and acquirer, to cause the payment to the second payment account within the selected time-slot.

11. A server according to claim 10 in which the program instructions are operative to cause the server to present to the sender, for each of the presented time-slots, the corresponding incentive.

12. A server according to claim 10 or 11 further comprising a scheduled payment database, the program instructions further being operative to cause the processor:

upon receiving from the payment sender a selection of one of the time-slots, to update a scheduled payment database of scheduled future payments,

to determine, in each of successive time-slots, whether the present time-slot is the time-slot in which the payment is scheduled, and if the determination is positive, to communicate with at least one of the issuer and acquirer, to instruct the payment.

13. A computerized network for performing payment transactions, the network comprising:

a server according to claim 10, 11 or 12;

a first communication device associated with the sender, the communication device being operative to transmit to the server the request, and the selection of one of the time-slots.

14. A computerized network according to claim 13 further comprising a second communication device associated with the recipient, the server being operative to receive from the first communication device contact details associated with the second communication device, and to transmit a notification to the second communication device according to the contact details.

15. A computer program product, comprising program instructions for implementation by the processor of a communication device, the program instructions being operative to cause the processor:

(i) to receive data input from a payment sender instructing a payment from a first payment account maintained by an issuer and associated with the payment sender, to a payment recipient;

(ii) to transmit to a server a payment request instructing the payment;

(iii) to receive from the server, and present to the user, one or more future time-slot options;

(iv) to receive data input from the user selecting one of the time-slot options; and

(v) to transmit to the server data indicative of the selected time-slot option.

16. A computer program product according to claim 15 in which the program instructions are further operative to receive from the server, and present in association with each presented time-slot option, a corresponding incentive.